

ALMA MATER STUDIORUM – UNIVERSITÀ DI BOLOGNA  
CAMPUS DI CESENA

---

Dipartimento di Informatica - Scienza e Ingegneria  
Corso di Laurea Magistrale in Ingegneria e Scienze Informatiche

**DAILY MEDICAL TEAM BRIEFINGS IN AMBIENTE  
COLLABORATIVO CON SCHERMI MULTI-TOUCH**

*Elaborato in*  
Pervasive Computing

*Relatore*  
Prof. Mirko Viroli

*Presentata da*  
Luca Bazzocchi

*Co-relatore*  
Dott. Gianluca Aguzzi

---

Quarta Sessione di Laurea  
Anno Accademico 2021 – 2022



## PAROLE CHIAVE

Interaction

Collaboration

Multi-Touch

Medical Briefings

Multi-User Application



*A tutte le persone che per me  
sono state fonte di ispirazione e crescita personale.*



# Abstract

Quotidianamente, le equipe mediche operanti nei reparti ad alto impatto sulla salute si riuniscono per discutere, caso per caso, lo stato e le terapie dei pazienti ricoverati nel proprio reparto. Queste riunioni, che richiedono la presenza di un nutrito gruppo di professionisti, sono in genere abbastanza concitate e richiedono di accedere con rapidità a dati e materiali di vario tipo. Attraverso un continuo confronto tra loro, che porta anche alla stesura di un verbale d'equipe, i partecipanti alla riunione decidono come procedere con la cura del paziente. Spesso però, in questi processi lavorativi, vi è una forte mancanza o inadeguatezza hardware e software di strumenti collaborativi, con la sola presenza di computer e software antiquati accompagnati da proiettori. Partendo da queste problematiche, si è voluto realizzare un prototipo in grado di supportare la collaborazione e la manipolazione del materiale dei pazienti per migliorare i meeting di equipe. Il prototipo consiste in una applicazione che fa uso di due schermi multi-touch: un coffee table e uno schermo a parete. Durante ogni sessione, l'applicazione è suddivisibile fino a quattro postazioni di lavoro per schermo, qui gli specialisti sono grado di visualizzare, manipolare e condividere il materiale di un paziente tra di loro, ma anche effettuare ricerche su internet, prendere appunti e redigere il verbale d'equipe. Tutto ciò avviene in modo sia individuale, perché ogni specialista lavora nella propria postazione, che cooperativo, in quanto i dati di una sessione sono condivisi tra i due schermi e facilmente spostabili da una postazione all'altra. Questo approccio permette di promuovere la collaborazione ed è in grado di aumentare l'efficacia dei briefing medici.

I processi riguardanti la raccolta dei requisiti e le riunioni volte alla valutazione di ogni release, sono state svolte con la partecipazione del personale dell'AUSL della Romagna.

# Indice

<b>1</b>	<b>Introduzione</b>	<b>1</b>
1.1	Ambienti Collaborativi . . . . .	1
1.2	Problematiche in Ambienti Collaborativi . . . . .	2
1.3	Lavoro Collaborativo in Ambito Sanitario . . . . .	3
1.4	Tecnologia Touch-Screen . . . . .	5
1.5	Obiettivi della Tesi . . . . .	7
<b>2</b>	<b>Caso di studio</b>	<b>9</b>
2.1	Briefing Giornalieri di Equipe Medica . . . . .	9
2.2	Schermi Multi-Touch . . . . .	10
<b>3</b>	<b>Specifiche</b>	<b>11</b>
3.1	Proof of Concept . . . . .	11
3.2	Department Page . . . . .	12
3.3	Sessione . . . . .	12
3.4	Postazione di lavoro . . . . .	13
<b>4</b>	<b>Design Architettuale</b>	<b>17</b>
4.1	Modularità della soluzione . . . . .	17
4.2	Module . . . . .	18
4.3	Module Strategy . . . . .	20
4.4	Cambio di stato . . . . .	23
4.5	Workstation . . . . .	24
4.6	Condivisione dei dati tra schermi . . . . .	25
<b>5</b>	<b>Interazione</b>	<b>27</b>
5.1	Inizializzazione di una sessione . . . . .	27
5.2	Sharing tra due postazioni . . . . .	28
<b>6</b>	<b>Dettagli Implementativi</b>	<b>31</b>
6.1	Module State Manager . . . . .	31
6.2	Implementazione delle Funzionalità . . . . .	34
6.3	Moduli Implementati . . . . .	37



<i>INDICE</i>	ix
6.4 Publishing and Device Discovering . . . . .	38
6.5 Session Management . . . . .	39
6.6 Serializzazione e Deserializzazione . . . . .	40
6.7 Interfaccia Utente . . . . .	41
<b>7 Utilizzo dell'Applicativo</b>	<b>43</b>
<b>Conclusioni e sviluppi futuri</b>	<b>53</b>
<b>Ringraziamenti</b>	<b>57</b>
<b>Bibliografia</b>	<b>59</b>

# Elenco delle figure

1.1	Collaborazione interprofessionale in ambito medico. . . . .	4
1.2	Esempi di gesture effettuati su un touch screen. . . . .	6
3.1	Use case del membro del team nella propria postazione. . . . .	13
4.1	Diagramma della strutturazione dei moduli. . . . .	18
4.2	Estensioni della classe Module. . . . .	19
4.3	Architettura di una funzionalità all'interno di un modulo. . . . .	21
4.4	Architettura di una strategia reattiva all'interno di un modulo. . . . .	22
4.5	Architettura di un modulo con focus sul cambio di stato dato dalla <code>ChangeStateStrategy</code> . . . . .	23
4.6	Composizione di una postazione basata sui diversi impieghi di essa. . . . .	24
4.7	Gestione della condivisione dei dati tra i due schermi. . . . .	26
5.1	Flusso d'interazione che porta all'inizializzazione di una sessione nei due schermi. . . . .	28
5.2	Condivisione dei contenuti tra due postazioni. . . . .	29
6.1	Esempio di diagramma degli stati di un modulo. . . . .	34
6.2	Esempio di verbale prodotto alla fine della sessione con un paziente. . . . .	40
6.3	Grafica di esempio di una singola postazione. . . . .	41
7.1	Pagina riservata a uno specifico dipartimento per la scelta del paziente su cui iniziare una sessione. . . . .	43
7.2	Applicazione avviata in entrambi gli schermi mentre si sta scegliendo il paziente. . . . .	44
7.3	Pagina relativa alle informazioni e documenti del paziente. . . . .	45
7.4	Piccolo browser incorporato all'interno del prototipo. . . . .	46
7.5	Pagina in cui sono indicati i nominativi dei membri del team. . . . .	47
7.6	Pagina in cui vengono descritte tutte le scelte e decisioni sulla cura del paziente. . . . .	47
7.7	Pagina in cui si raggruppano e manipolano le immagini e i documenti necessari a prendere decisioni sulla cura del paziente. . . . .	48

7.8	Esempio di manipolazione tramite il touch di un Image Module	50
7.9	Module Page che sta effettuando una condivisione dei propri contenuti con l'altro schermo e presenta la tastiera virtuale aperta.	50
7.10	Esempio di utilizzo contemporaneo di due postazioni di lavoro nello stesso schermo. . . . .	51
7.11	Esempio di utilizzo contemporaneo di due postazioni di lavoro in entrambi gli schermi. . . . .	51
7.12	Esempio di verbale prodotto alla fine della sessione con un paziente.	52



# Capitolo 1

## Introduzione

In ambiente sanitario i professionisti sono sempre spronati a lavorare dando grande importanza alla collaborazione e allo scambio di informazioni e conoscenze tra di loro, in modo tale da essere in grado di prescrivere le migliori terapie per la cura dei propri pazienti. La cura di un paziente è un lavoro di gruppo in cui diversi specialisti si incontrano, dialogano e collaborano tra loro per prendere decisioni. I vari processi devono essere altamente rigorosi e devono rispettare determinate politiche, prefissate a priori in quanto la medicina è un ambito molto delicato in cui sono impegnate figure altamente specializzate. Questo lavoro potrebbe essere accompagnato da una tecnologia che valorizzi la collaborazione e migliori le interazioni tra i vari individui, incrementando così l'efficacia dei meeting medici; ma spesso, in realtà, vi è una grande mancanza o inadeguatezza hardware e software di strumenti collaborativi, con l'unica dotazione di computer e software antiquati accompagnati da proiettori.

### 1.1 Ambienti Collaborativi

Con il termine **Collaborazione** si intende una relazione reciprocamente vantaggiosa e ben definita intrapresa da due o più persone o organizzazioni per raggiungere obiettivi comuni; mentre con **Ambiente** si indica il luogo e le condizioni dove queste relazioni operano e avvengono. Un ambiente collaborativo include un impegno volto alla definizione di relazioni e obiettivi reciproci, una responsabilità condivisa e volta al successo, una autorità reciproca e la condivisione di risorse e ricompense.

Il successo di una collaborazione non è il risultato di una formula o il prodotto di una struttura, la collaborazione è un processo che richiede attenzione, impegno e lavoro. Il *comportamento* e la *specializzazione* delle persone che lavorano all'interno di un ambiente collaborativo sono fattori essenziali; è infatti necessaria la comprensione, il rispetto e la fiducia reciproca tra i membri parte-

cipanti, come anche una appropriata selezione dei membri per specializzazione e background nell'ambito di interesse in modo tale che ogni persona abbia un coinvolgimento continuo nell'iniziativa collaborativa. Altri importanti fattori riguardano la *flessibilità*, ovvero come il gruppo collaborativo rimane aperto a vari modi di organizzarsi e portare a termine il proprio lavoro, l'*adattabilità*, come il personale ha la capacità di sostenersi a seguito di grandi cambiamenti, come la modifica degli obiettivi, la scelta di *ruoli specifici*, per far capire i diritti e le responsabilità del singolo individuo, una *comunicazione aperta e frequente*, i membri del team interagiscono, si aggiornano e discutono dei problemi apertamente cercando di comunicare tutte le informazioni necessarie alle altre persone, e una *visione condivisa*, con la missione, gli obiettivi e la strategia chiaramente concordati [1].

## 1.2 Problematiche in Ambienti Collaborativi

Lo sviluppo di applicazioni collaborative prende in considerazione come l'approccio e l'interazione tra più utenti in contemporanea possa rendere più efficace ed efficiente un determinato lavoro, rispetto all'impiego di una sola persona, ma quando questo non è gestito correttamente, l'ambiente può diventare caotico. Avere più persone, ognuna con i propri obiettivi e diversi livelli di esperienza, che utilizzano insieme la stessa tecnologia, introduce possibili situazioni imprevedibili e inaspettate, inoltre, più è complesso il sistema collaborativo, maggiore è la probabilità che si verifichino situazioni di disagio. L'utente che deve gestire ambiguità di questo genere, dovrà controllare continuamente che tutto proceda senza problemi, o sarà sopraffatto dall'ambiente e non sarà in grado di portare a termine il suo normale lavoro.

L'obiettivo è rimuovere, o almeno ridurre, l'imprevedibilità dagli strumenti a supporto degli utenti, in modo tale che essi si concentrino solo ed esclusivamente sulla collaborazione e non sugli strumenti. Questo è possibile definendo delle *policy*, ovvero un sistema deliberato di linee guida per indirizzare le decisioni e ottenere risultati razionali e per definire come gli utenti e il sistema interagiscono tra loro.

Una policy deve:

1. ridurre l'imprevedibilità di un sistema, in modo che l'applicativo risponda alle azioni degli utenti in modo previsto e secondo il principio "*least surprise*", ovvero, l'idea che un utente non dovrebbe essere sorpreso dal modo in cui si svolge un'interazione o il funzionamento di un oggetto dall'interfaccia o dal suo design;

2. far richiedere meno sforzo possibile agli utenti, portando la maggior parte del carico di lavoro sul sistema e non sugli utenti.

Al contempo la policy non deve essere troppo rigida, in quanto bisogna tenere conto che la dinamicità dei componenti di un sistema, pur venendo limitata, deve mantenere degli aspetti fondamentali che promuovano la collaborazione e interazione tra utenti [2].

Come l'architettura definisce l'interazione tra i suoi componenti e la consapevolezza in questi delle funzionalità riguardanti la collaborazione, influenza il riuso, le performance, la facilità di modifica, la tolleranza ai guasti e il tempo richiesto a sviluppare l'applicazione [3]. Questo comporta inoltre una difficoltà nel decomporre il problema in esame, in parti più piccole e le conseguenze date dalla scelta di diverse decomposizioni non sono sempre evidenti.

### 1.3 Lavoro Collaborativo in Ambito Sanitario

L'ambito sanitario è un esempio primario di lavoro collaborativo. Negli ospedali, il personale si prende cura dei pazienti mettendo a frutto la propria esperienza e i propri sforzi; i medici sono persone altamente specializzate che giornalmente si confrontano e lavorano tra di loro per avere un impatto positivo nella cura del paziente. In questo caso si parla di collaborazione interprofessionale (IPC) e di team multidisciplinari (MDTs), il cui impiego ha portato migliori risultati clinici e di processo per i malati di cancro, evidenziando una migliore sopravvivenza dei pazienti [4]. Lavorare in modo collaborativo implica delle relazioni di lavoro regolari a fronte di compiti altamente connessi e interdipendenti, mentre lavorare in modo interprofessionale implica una prospettiva integrata tra i diversi specialisti coinvolti sulla cura del paziente [5]. La ricerca ha ripetutamente documentato l'impatto dei problemi legati alla collaborazione sui processi lavorativi e sulla sicurezza dei pazienti; pertanto, i professionisti devono assicurarsi di collaborare in modo efficace per fornire un'assistenza ai pazienti sicura e di alta qualità [6]. Inoltre, i sistemi informatici degli ospedali raramente sono integrati o standardizzati, le immagini relative ai pazienti sono spesso di bassa qualità o non soddisfano requisiti specifici, tutto ciò porta importanti ritardi nel processo decisionale [7].

In ambito sanitario emergenze, eccezioni e altre contingenze sono abbastanza frequenti e tali da ostacolare la standardizzazione dei processi lavorativi, inoltre è un lavoro non di routine e questo rende difficile programmare in anticipo eventi e attività, a differenza di altri ambienti industriali. Queste caratteristiche portano a dipendere da intense attività di comunicazione delle informazioni e di elaborazione per raggiungere una adeguata coordinazione, tutto questo raggiungibile grazie anche ad un ampio impiego di tecnologia dell'informazione

e della comunicazione (ICT), per coordinare il lavoro. La costruzione di ottimi rapporti tra i professionisti può portare ad un aumento della produttività nell'uso dell'ICT [8].



Figura 1.1: Collaborazione interprofessionale in ambito medico.

Fonte: <https://www.rn.com/nursing-news/importance-of-interprofessional-collaboration/>

Non sempre però un ampio uso della tecnologia ha portato dei successi, ci sono state anche resistenze e fallimenti [9]. Un'implicazione chiave è che le soluzioni tecniche, da sole, non sono la risposta per promuovere il lavoro interprofessionale, che talvolta risulta scadente, con problemi legati alla mancanza di responsabilità condivise, poca chiarezza sui ruoli e una limitata istruzione e formazione interprofessionale [10]. Questo porta alla conclusione che le applicazioni ICT possano fornire strumenti efficaci per promuovere la comunicazione e la raccolta di informazioni, ma che l'efficacia di una collaborazione interprofessionale dipenda maggiormente dalle relazioni sociali e dal contesto dell'organizzazione in cui si inseriscono [11]. Infatti, sono presenti studi in cui viene mostrato un utilizzo non ottimale dell'ICT nei team, pur dimostrando che la raccolta e visualizzazione dei dati in tempo reale possano migliorare il coordinamento dell'assistenza centrata sul paziente [12]. Sono però stati pubblicati articoli che dimostrano l'importanza dell'ICT nel supporto di team di professionisti [13], ad esempio nelle cure dei pazienti in fin di vita, la comunicazione di dati e informazioni tra i professionisti e il resto del personale



medico deve essere altamente efficace. Ma anche studi che rivelano come l'ICT nelle cliniche rurali e nelle case di cura si stato impiegato per la riduzione del trasporto di emergenza di pazienti in queste case. Pertanto, si tiene conto che, l'uso continuo ed efficiente dell'informatica a supporto degli ambienti medici, può ridurre l'onere non solo per i professionisti, ma anche per i pazienti e le loro famiglie [14].

La natura dell'ambiente sanitario crea una sfida nel colmare il divario tra il tangibile, data dall'importanza degli oggetti fisici di uso comune in medicina, e il virtuale [15].

## 1.4 Tecnologia Touch-Screen

Il touch screen è un dispositivo elettronico che accetta input dell'utente direttamente dal monitor, lo schermo è in grado di percepire la posizione e il movimento delle dita e altri oggetti passivi come le penne stilografiche.

Uno schermo touch presenta tre componenti principali:

1. **sensor**: rileva i punti di contatto, ha generalmente una corrente elettrica o un segnale che gli passa attraverso e il tocco che causa un cambio di voltaggio o di segnale. Questo cambiamento si usa per rilevare la posizione del tocco;
2. **controller**: è un intermediario tra il sensore e il computer. Prende l'informazione del sensore e la traduce in una informazione recepibile dal computer;
3. **driver software**: indica al computer come interpretare l'informazione dell'evento touch inviatogli dal controller.

Tramite questi componenti si realizza l'architettura e il funzionamento base di questa tecnologia [16].

I touch screen, per come sono strutturati, permettono di interagire con i computer a un livello differente rispetto alla tecnologia a cui si è abituati; strumenti come il mouse e la tastiera stanno lentamente scomparendo per essere rimpiazzati da approcci più comodi e naturali, tra cui il toccare direttamente lo schermo con le proprie dita per manipolare gli elementi presenti.

L'hardware e il software alla base di questi schermi touch rilevano l'input dell'utente in diversi modi, un input che non è uno scroll o click del mouse, ma delle gesture effettuate direttamente sullo schermo. Quello che si ha normalmente è una libreria software che lavora ad un livello tra lo schermo hardware e l'applicazione, che restituisce l'oggetto toccato e il tipo di gesture effettuata su di esso. Le gesture sono date dal movimento di una o più mani

per esprimere un'idea o un significato e queste, sulla base del movimento e il numero di dita presenti, possono essere più o meno complicate: il movimento del singolo componente tramite un dito è meno complicato di una selezione effettuata usando due dita contemporaneamente.

In fig. 1.2 sono visualizzati alcuni esempi di gesture e tra questi è presente anche l'utilizzo del multi-touch.

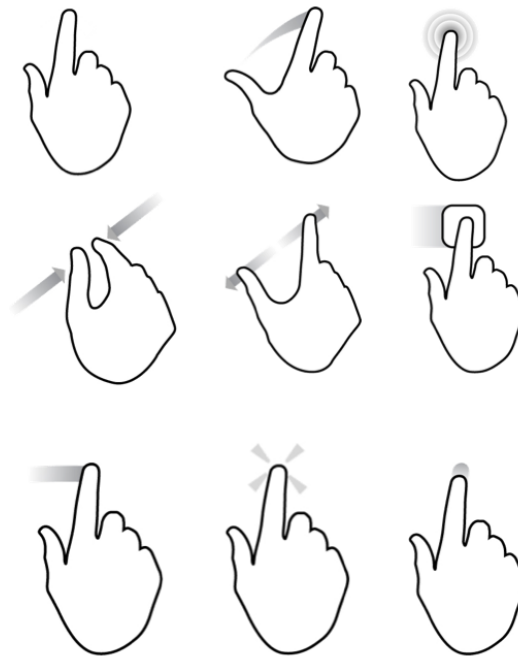


Figura 1.2: Esempi di gesture effettuati su un touch screen.

Fonte: <https://www.graffletopia.com/stencils/432>

Normalmente il mouse e la tastiera sono gli strumenti utilizzati per interagire con il computer, questi strumenti, tramite la premuta di pulsanti, forniscono dell'input al computer che viene gestito singolarmente sulla base del tipo. Tuttavia, tramite l'utilizzo del multi-touch, è possibile eseguire più azioni contemporaneamente evolvendo dalla normale gestione singola dell'input. Il limite è dato solamente dall'uso che ne viene fatto e dalle effettive manipolazioni possibili tramite gesture. Questo amplia l'utilizzo tradizionale del dispositivo a disposizione del singolo utente, con applicazioni multi-user con utenti che svolgono molteplici azioni contemporaneamente e magari anche collaborando tra di loro [17].

Tutto ciò porta a una nuova interpretazione sul design e sull'architettura delle applicazioni per tener conto delle attività svolte in modalità concorrente, questo porta ad una struttura ad eventi con un flusso di controllo asincrono

rispetto al tradizionale approccio sincrono, dove si attende il completamento della singola computazione prima di procedere a quella successiva.

Negli ultimi decenni, l'uso del touch screen sta venendo sempre più impiegato in ambito medico, questa tecnologia ha il potenziale di semplificare la rieducazione di pazienti con lesioni cerebrali [18], ma anche di migliorare la formazione di team interprofessionali attraverso un teleconsulto in aule virtuali con professionisti del settore, mentre viene mostrato in tempo reale il paziente sotto cura [19]. Gli schermi tattili sono stati impiegati per facilitare l'incontro e la comunicazione tra i medici e i pazienti più anziani [20] e, in modo simile, è stato valutato in positivo l'uso di chioschi touch screen all'interno degli ospedali, per soddisfare i bisogni informativi dei pazienti con domande in ambito medico [21].

Sulla base dei successi nell'uso delle tecnologie dell'informazione e della comunicazione ma anche nell'impiego specifico di touch screen in campo medico, si procede a presentare l'idea del sistema che è stato conseguito con questa tesi.

## 1.5 Obiettivi della Tesi

Dopo una ricerca approfondita sulle problematiche presenti nelle riunioni in cui i medici, discutono, caso per caso, lo stato e le terapie dei pazienti ricoverati nel proprio reparto, si è voluto realizzare un prototipo in grado di supportare la collaborazione e la manipolazione del materiale dei pazienti per migliorare questi meeting.

Nell'applicativo risultante si vuole avere la possibilità di interagire direttamente con i dati e il materiale del paziente tramite il touch. Questa interazione non dovrà però essere singola per dispositivo, come nell'utilizzo standard di un computer, ma deve promuovere la collaborazione tra più individui nello stesso schermo. Ciò può essere reso possibile tramite l'utilizzo del multi-touch, quindi la rilevazione contemporanea di più tocchi, e la suddivisione dello stesso schermo in più aree, ognuna con le stesse funzionalità, che chiameremo *postazioni*. In aggiunta il prototipo non verrebbe avviato su un solo schermo, ma su due schermi contemporaneamente: un coffee table e uno schermo a parete. Questo può dare la possibilità a tutti gli specialisti, anche in caso di team numerosi, di ricoprire un ruolo attivo durante l'intera durata della riunione.

Per rendere più efficienti i meeting, ogni postazione deve dare la possibilità di reperire con facilità le informazioni e documenti del paziente in esame, in modo tale da non presentare rallentamenti dati dall'aver solamente una persona operativa per dispositivo. Visto che in ogni riunione saranno presenti medici con diversi livelli di esperienza e specializzazione, le postazioni devono presentare diverse funzionalità abilitate tramite il touch, tra cui:

1. poter manipolare il materiale dei pazienti;
2. prendere appunti;
3. effettuare ricerche su internet;
4. condividere i dati con un'altra postazione;
5. redigere il verbale d'equipe.

Uno dei punti focali riguarda la possibilità di condividere i dati tra due postazioni contemporaneamente; questa condivisione, a differenza di un normale invio di dati in maniera unilaterale, sarà bidirezionale e permetterà perciò una doppia interazione con un conseguente aggiornamento in entrambi gli schermi.

Nelle riunioni mediche odierne vi è il problema che a fine di una sessione è frequente perdere il riferimento del materiale utilizzato e gli appunti presi, avendo traccia solo di ciò che viene riportato nel verbale di equipe. Inoltre, una terminazione improvvisa di una riunione dovuta da fattori esterni, rallenta la ripresa della sessione stessa in un secondo momento. Il prototipo, perciò, introdurrà il concetto di sessione e lo espanderà dando la possibilità di salvare l'intero stato dei due schermi: viene memorizzato tutto il materiale manipolato e il relativo orario di apertura, gli appunti presi ma anche lo stato corrente del verbale d'equipe, insieme ai nominativi dei membri del team e la durata del meeting. In questo modo sarà possibile fermare e riprendere una riunione con facilità o, nel caso si presentassero pazienti con problematiche simili, sarebbe possibile riguardare in un secondo momento una sessione archiviata per esaminare il materiale a cui si è fatto riferimento.

Questi sono gli obiettivi designati per promuovere la collaborazione e aumentare l'efficacia dei briefing medici e saranno analizzati nei capitoli successivi.

# Capitolo 2

## Caso di studio

Grazie al continuo progresso scientifico e tecnologico, un numero sempre maggiore di persone ha introdotto, sia nella propria abitazione che in ambito lavorativo, delle innovazioni tecnologiche in grado di supportare e semplificare determinate attività quotidiane.

L'obiettivo del progetto è la realizzazione di un prototipo che promuova l'interazione tra più persone e che sia in grado di migliorare la cooperazione, l'efficienza e l'efficacia di una attività normalmente svolta in un ambiente collaborativo che non impiega al massimo la tecnologia a disposizione.

### 2.1 Briefing Giornalieri di Equipe Medica

Quotidianamente, le equipe mediche operanti nei reparti ad alto impatto sulla salute (come Terapia Intensiva, Neurochirurgia, Oncologia, Trauma Team, Stroke Unit, e similari) si riuniscono per discutere, caso per caso, lo stato e le terapie dei pazienti ricoverati nel proprio reparto.

Ogni riunione prevede la presenza di un nutrito gruppo di professionisti, alcuni in presenza, altri in remoto (per il cosiddetto teleconsulto specialistico), che analizzano dati eterogenei relativi a ciascun paziente: cartella clinica, piano terapeutico e aderenza della cura, immagini radiologiche, referti di esami (tutti con dati digitali, strutturati), etc. Sulla base di questi dati e di un confronto tra loro che porta anche alla stesura di un verbale di equipe, i partecipanti alla riunione decidono come procedere con la cura del paziente.

Le riunioni sono in genere abbastanza concitate e richiedono di accedere con rapidità a materiali di vario genere, da parte di specialisti eterogenei in termini di professionalità. Inoltre, le emergenze, le eccezioni e altri tipi di contingenze sono abbastanza frequenti e ostacolano la collaborazione interprofessionale. Le imprevedibilità presenti negli strumenti a supporto del personale medico devono

essere rimosse o almeno ridotte, affinché gli specialisti si concentrino solo ed esclusivamente sulla collaborazione.

## 2.2 Schermi Multi-Touch

Ai Daily Medical Team Briefings, si è scelto d'impiegare due schermi multi-touch: un coffee table e uno schermo a parete, come tecnologia di base su cui implementare il prototipo. Attraverso l'utilizzo del touch, i medici potranno eseguire delle gesture e altri tipi di interazione per manipolare e gestire il materiale relativo ai pazienti ricoverati. Il focus è sulla collaborazione tra più individui e il rendere i membri del team sempre operativi, senza che nessuno debba necessariamente aspettare gli altri. Questo è reso possibile grazie all'utilizzo di più schermi e contemporaneamente, più postazioni all'interno di essi. Il non dover usare mouse e tastiera, strumenti utilizzati normalmente per interagire con il computer, espande il singolo e tradizionale utilizzo di un dispositivo. Ciò rende l'applicazione multi-user e allo stesso tempo porta a un apprendimento più rapido e naturale della tecnologia grazie all'uso diretto delle proprie mani.

Si presuppone che tali schermi siano posizionati all'interno della stessa stanza, che siano in grado di accedere alla stessa rete e che utilizzino Windows come sistema operativo. Sulla base di queste specifiche, si è scelto d'implementare la soluzione attraverso WPF (Windows Presentation Foundation) [22] che presenta supporti per la rilevazione e analisi del touch sulle interfacce.

Una soluzione che faccia uso di entrambi gli schermi in modo collaborativo, ovvero che supporti la cooperazione per raggiungere un fine comune e agevoli la condivisione di materiale, applicata a questo contesto, sarebbe un'ottima dimostrazione di come potrebbero evolvere i meeting di equipe, a oggi gestiti in modo non propriamente smart.

# Capitolo 3

## Specifiche

L'obiettivo è la realizzazione di un prototipo che permetta agli specialisti di collaborare interagendo con i dati e documenti relativi al paziente. Il focus viene posto sull'interazione e non sulla raccolta e organizzazione dei dati, questo è un problema a sé stante che non viene approfondito all'interno di questa trattazione. Si assume perciò che il prototipo sia supportato da un software in grado di mantenere organizzati e di reperire i dati dei singoli pazienti nei vari reparti.

Inoltre, non verranno approfonditi tutti gli aspetti riguardanti l'autorizzazione e l'accesso ai dati sensibili, la cui osservazione e manipolazione deve essere riservata al solo personale autorizzato. In questo caso, alcune specifiche dovrebbero essere analizzate nuovamente e reingegnerizzate rispetto agli standard nazionali ed europei, ma non essendo questo il nostro campo di competenza, si rimanda tale aspetto a sviluppi futuri.

La raccolta dei requisiti e delle specifiche è stata fatta in collaborazione con il personale dell'AUSL Romagna, frutto di un'ampia interlocuzione con il personale e i clinici. In questo capitolo vengono analizzate queste specifiche e le caratteristiche necessarie per implementare il tipo di applicativo prefissato.

### 3.1 Proof of Concept

Il progetto consiste nella creazione di una applicazione software avviabile in due schermi multi-touch che permetta, a un team di medici, di svolgere una o più sessioni per i pazienti del proprio reparto.

L'evoluzione rispetto ai normali meeting di equipe è triplice in quanto si aggiunge la possibilità di:

- manipolare in modo semplice e naturale le immagini e i documenti riguardanti un paziente attraverso l'utilizzo di gesture;

- far partecipare attivamente tutti i membri del team, assegnando a ciascuno di essi dei task specifici e indipendenti in sottosezioni dello schermo;
- salvare lo stato corrente di una sessione per riprenderla in futuro.

Al fine di rendere la soluzione modulare e riusabile all'interno di una sessione, si introduce il concetto di **postazione**, cioè la sottoparte dell'intero schermo che comprende di tutte le funzionalità dell'applicazione ed è adibita al lavoro di almeno un individuo.

Ogni schermo può contenere una o più postazioni, fino a un massimo di quattro, in cui i vari professionisti potranno lavorare su molteplici task e in modo indipendente rispetto alla postazione ove decideranno di posizionarsi, in quanto tutte avranno le stesse funzionalità e potenzialità.

Lo stesso software, presente sia nel coffee table che nello schermo a parete, è in grado di far comunicare tra di loro le diverse postazioni al fine di condividere il materiale manipolato o l'uso dello schermo.

## 3.2 Department Page

La pagina iniziale dell'applicazione è la pagina relativa al dipartimento. Qui i medici visualizzano, sulla base del dipartimento di appartenenza, i pazienti ricoverati e hanno la possibilità di:

1. cercare o visualizzare un paziente tramite:
  - 1.1 nome;
  - 1.2 stato (in attesa, sotto osservazione, sotto trattamento);
  - 1.3 codice (verde, giallo, viola), sulla base dello stato;
  - 1.4 posizione all'interno del reparto;
2. iniziare una nuova sessione con il paziente cercato;
3. riprendere una sessione salvata in precedenza.

## 3.3 Sessione

La sessione rappresenta tutto quell'iter che parte dall'analisi di un determinato paziente e, attraverso la discussione e la raccolta dei dati tra i vari specialisti, porta alla stesura di un verbale d'equipe che rappresenta il processo tramite il quale verrà curato il paziente.

I requisiti che la sessione deve soddisfare sono i seguenti:



1. è focalizzata sul singolo paziente;
2. presenta una durata;
3. presenta il nome degli specialisti;
4. si può concludere o mettere in pausa in qualunque momento;
  - 4.1 ciò porta al salvataggio della sessione e alla produzione di un report;
5. è possibile continuare dall'ultimo snapshot salvato;
6. tiene conto delle immagini e documenti aperti durante la sessione.

### 3.4 Postazione di lavoro

Iniziata una sessione con un paziente, è possibile suddividere lo schermo multi-touch in una o più postazioni, fino a un massimo di quattro, tenendo presente che:

1. durante la sessione, tale scelta può essere modificata in qualunque momento;
2. ogni postazione non ha un ruolo specifico;
3. il ruolo che si impiega in una postazione dipende dalle azioni che si effettuano e dalla pagina che si sta visualizzando.

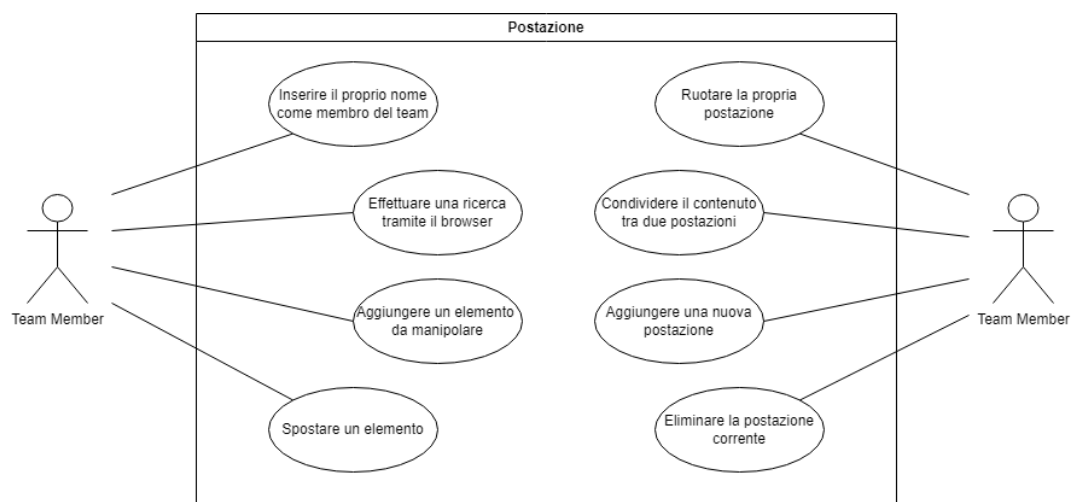


Figura 3.1: Use case del membro del team nella propria postazione.

Ogni postazione presenta, a lato, un menu, tramite il quale è possibile effettuare una serie di azioni quali:

1. cambiare la pagina al momento visualizzata;
2. ruotare la visualizzazione di 180 gradi;
3. aggiungere una nuova postazione;
4. eliminare la postazione corrente.

Di seguito vengono specificati i requisiti che le varie pagine devono soddisfare.

**Patient Page** La Patient Page è la pagina adibita alla visualizzazione di tutte le informazioni anagrafiche riguardanti il paziente oggetto della sessione. Tutti i file contenenti documenti, immagini, terapie passate e in corso, referti, allergie e cartelle cliniche sono consultabili all'interno della relativa sezione nella pagina. Ogni file può essere aperto:

- tramite applicazione esterna;
- all'interno della Module Page per essere manipolato.

**Module Page** La pagina dei moduli è una delle parti principali e focus dell'applicazione. All'interno di questa pagina, inizialmente vuota, uno specialista è in grado di manipolare e aggiungere al suo interno degli elementi, detti moduli, tramite il touch. Un **modulo** è un componente di base, indipendente, che può essere manipolato; ogni file aperto all'interno della Module Page, che sia un'immagine o un documento, diventa un modulo. Inoltre, attraverso il touch, ogni modulo può essere spostato tra le varie Module Page presenti nella postazione dello stesso schermo o condiviso nell'altro.

I due schermi, tramite un apposito pannello di connessione, possono connettere una o più coppie di Module Page tra di loro, in questo modo, finché connesse, le coppie di pagine condividono tutte le variazioni e l'inserimento di elementi in real time.

Nello specifico, in questa pagina, ogni specialista può:

1. scrivere o disegnare tramite touch;
2. prendere appunti tramite tastiera virtuale;
3. inserire documenti;
4. aprire immagini;

5. tramite touch su un modulo:
  - 5.1 fare zoom-in e zoom-out;
  - 5.2 effettuare una rotazione;
  - 5.3 spostarlo all'interno della stessa postazione o in un'altra;
6. effettuare lo sharing della propria postazione sulla postazione dell'altro schermo:
  - 6.1 entrambe le pagine presenteranno lo stesso contenuto;
  - 6.2 ogni azione su un modulo su uno schermo, avviene a sua volta anche nell'altro schermo e viceversa.

**Team Members Page** La Team Members Page contiene le informazioni riguardanti i membri del team che stanno effettuando la sessione sul paziente. Tali informazioni sono condivise tra i due schermi e tra tutte le postazioni. In qualunque momento, uno specialista, può aggiungere il suo nome e cognome all'elenco dei partecipanti.

**Report Page** Come da nome, all'interno di questa pagina è possibile redigere il report finale dell'intera sessione, ovvero il verbale d'equipe. Questo testo è condiviso tra tutte le postazioni e tra i due touch screen e aggiornato periodicamente, in modo tale che qualunque specialista possa proseguire il report se colui che se ne stava occupando fino a quel momento, risulta impegnato o ha altre urgenze.

Questa pagina è inoltre provvista, come la Module Page, di una tastiera virtuale per la scrittura.

Al concludersi della sessione, tutto il testo viene inserito all'interno del verbale.

**Browser Page** Questa pagina incorpora al suo interno un browser che permette ai medici di effettuare ricerche su internet nel caso in cui siano necessarie informazioni o immagini specifiche non presenti all'interno dell'applicazione.



# Capitolo 4

## Design Architettuale

In questo capitolo viene descritto il modello di massima prodotto dall'analisi del dominio applicativo in tutte le sue macro-parti e ne vengono, nel dettaglio, esplorate le componenti più rilevanti, analizzandone le scelte architetture.

### 4.1 Modularità della soluzione

Le sezioni principali del sistema si possono vedere come il raggruppamento di due componenti fondamentali:

- **modulo:** componente di base indipendente che può essere manipolato;
- **postazione:** una parte dell'applicazione completa su cui è possibile effettuare delle azioni.

Un modulo è un contenitore che può avere al suo interno un'immagine, un documento, un assistente vocale, un testo; ovvero, ogni elemento che può essere manipolato dal software. Interattivamente tramite il touch, ogni modulo può:

1. essere spostato all'interno della stessa postazione;
2. inviato verso un'altra postazione attraverso una gesture specifica;
3. essere ingrandito o rimpicciolito attraverso zoom-in o zoom-out;
4. essere condiviso tra i due schermi.

Un modulo deve essere facilmente manipolabile e componibile tramite funzionalità che rappresentano ciò che il modulo stesso può o non può fare. Inoltre, un modulo deve essere facilmente estendibile in moduli più complessi con all'interno oggetti manipolabili e di varia natura, come immagini e documenti.

La postazione, che rappresenta una partizione del software completo, è composta da un insieme di moduli e componenti software e utilizza i moduli per presentare diverse funzionalità all'utente. In aggiunta, ogni postazione è suddivisa a sua volta in pagine, ognuna con uno scopo ben delineato.

Tramite questi due componenti è possibile modellare gli aspetti d'interazione e collaborazione tra più individui. Lo stesso dominio applicativo potrebbe cambiare facilmente sulla base dei dati e materiale che viene visualizzato e manipolato.

## 4.2 Module

Un *Module*, da qui denominato modulo, è il componente principale d'interazione e manipolazione, motivo per il quale si è deciso di porre attenzione alla massima possibilità di personalizzare i moduli.

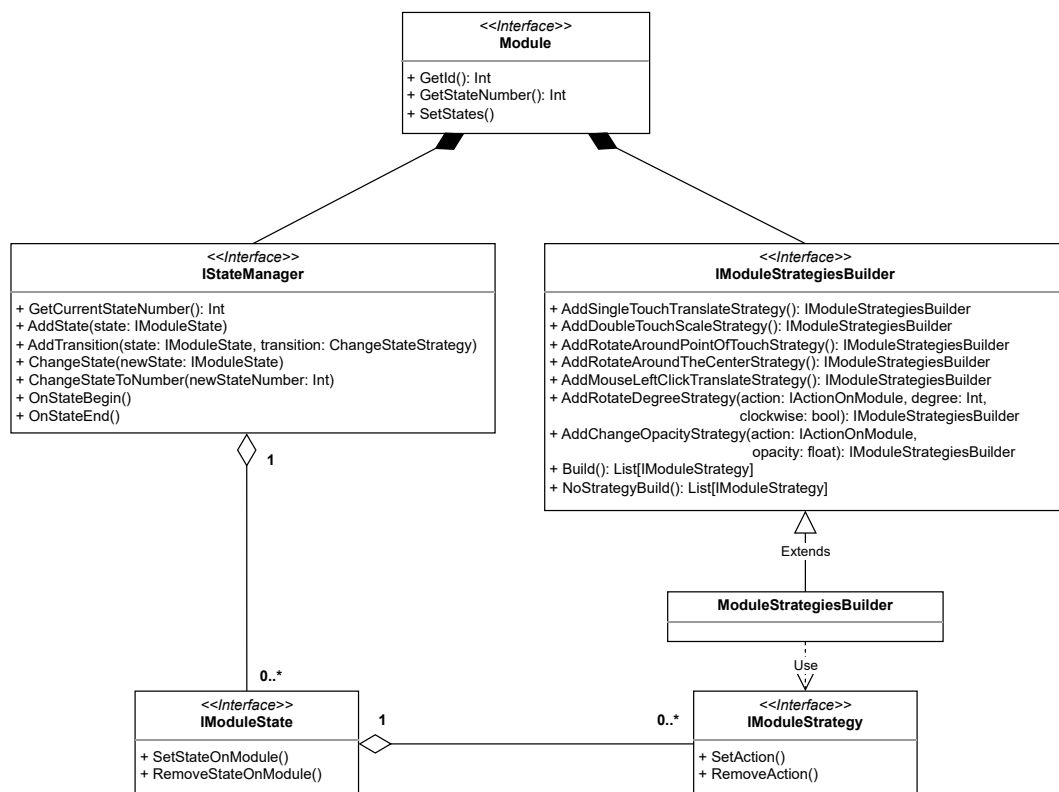


Figura 4.1: Diagramma della strutturazione dei moduli.

In fig. 4.2 è rappresentata una versione semplificata della struttura che compone un modulo. Ogni modulo presenta uno o più stati, dove lo stato

indica le funzionalità abilitate sul modulo stesso; nella transizione da uno stato e a un altro, il comportamento dell'intero modulo e, quindi l'interazione con esso, cambia.

Più in dettaglio abbiamo:

- **IModuleStrategy**: rappresenta la singola funzionalità abilitata su un modulo e richiede una interazione specifica (ad esempio una gesture) per essere attivata e produrre il cambiamento implementato in questa strategia;
- **IModuleStrategyBuilder**: utilizzo del pattern builder per agevolare la creazione delle funzionalità;
- **IModuleState**: costituisce uno dei possibili stati in cui il modulo può trovarsi. Questo presenta un insieme di funzionalità abilitate mentre lo stato è attivo;
- **IStateManager**: è il gestore degli stati di un modulo che utilizza lo state pattern. Esso definisce le transizioni: una tripla contenente lo stato di partenza, lo stato di arrivo e l'interazione necessaria per abilitare la transizione stessa.

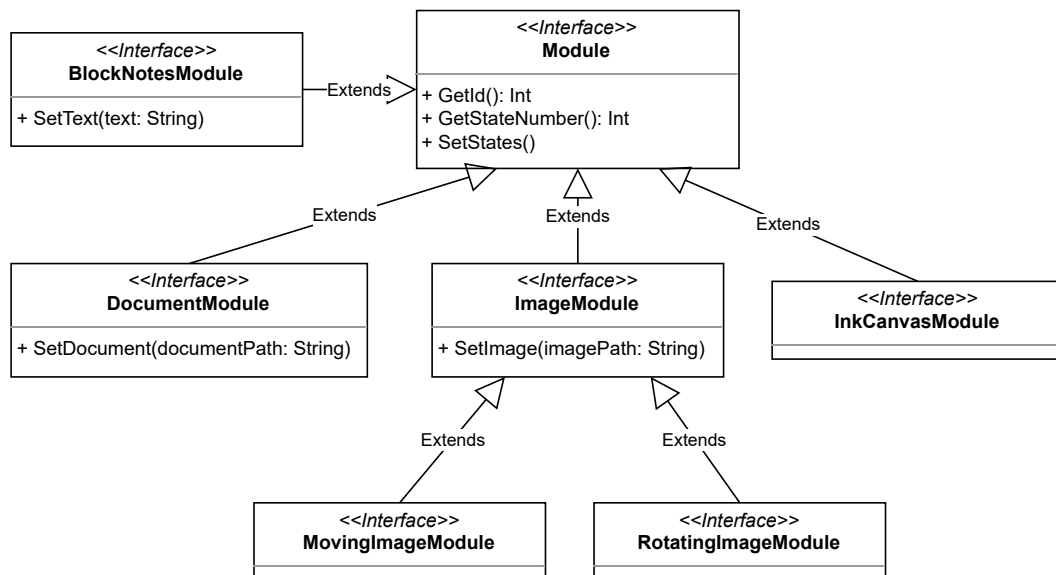


Figura 4.2: Estensioni della classe Module.

Il Module è effettivamente un contenitore di componenti, e questi sono interagibili e manipolabili dall'utente. Questi componenti si distinguono gli

uni dagli altri sulla base di ciò che rappresentano e dalle funzionalità che sono abilitate nei loro possibili stati. Dal Module possono essere prodotte varie estensioni quali:

- **BlockNotesModule**: blocco note in cui è possibile scrivere al suo interno attraverso l'uso della tastiera virtuale;
- **InkCanvasModule**: sezione bianca sulla quale disegnare, scrivere, cancellare o selezionare tratti tramite il touch;
- **DocumentModule**: modulo adibito alla visualizzazione di un documento pdf;
- **ImageModule**: permette di manipolare una immagine. L'interazione dipende dalle sue estensioni.

Attraverso la struttura appena definita, il comportamento di un modulo creato in precedenza risulta molto semplice da definire e implementare.

### 4.3 Module Strategy

Analizzata la struttura generale di un modulo, si procede a esaminare più a fondo come le varie funzionalità, ovvero i **Module Strategy**, siano composte e suddivise.

Tra la classe *ModuleStrategiesBuilder* e *IModuleStrategy* è presente il pattern di progettazione Strategy. Tale strategia presenta due metodi: *SetAction()* che inserisce il metodo all'interno del modulo e ne permette l'esecuzione e *RemoveAction()* per rimuovere tale strategia.

Per comprendere meglio la figura 4.3, è necessario specificare le categorie principali nelle quali le strategie sono state suddivise:

1. **Strategie Reattive** che determinano un comportamento la cui attivazione avviene solo se una singola e determinata azione (*IActionOnModule*) avviene sul modulo, come il tap di un dito. In fig. 4.3 sono quelle che estendono *IUpdateStrategy*;
2. **Strategie Attive** che definiscono il comportamento del modulo quando avvengono determinate azioni prolungate nel tempo, come il movimento di un dito sullo schermo. Esse estendono direttamente la classe *AbstractModuleStrategy* o implementano *IManipulationDeltaStrategy*.



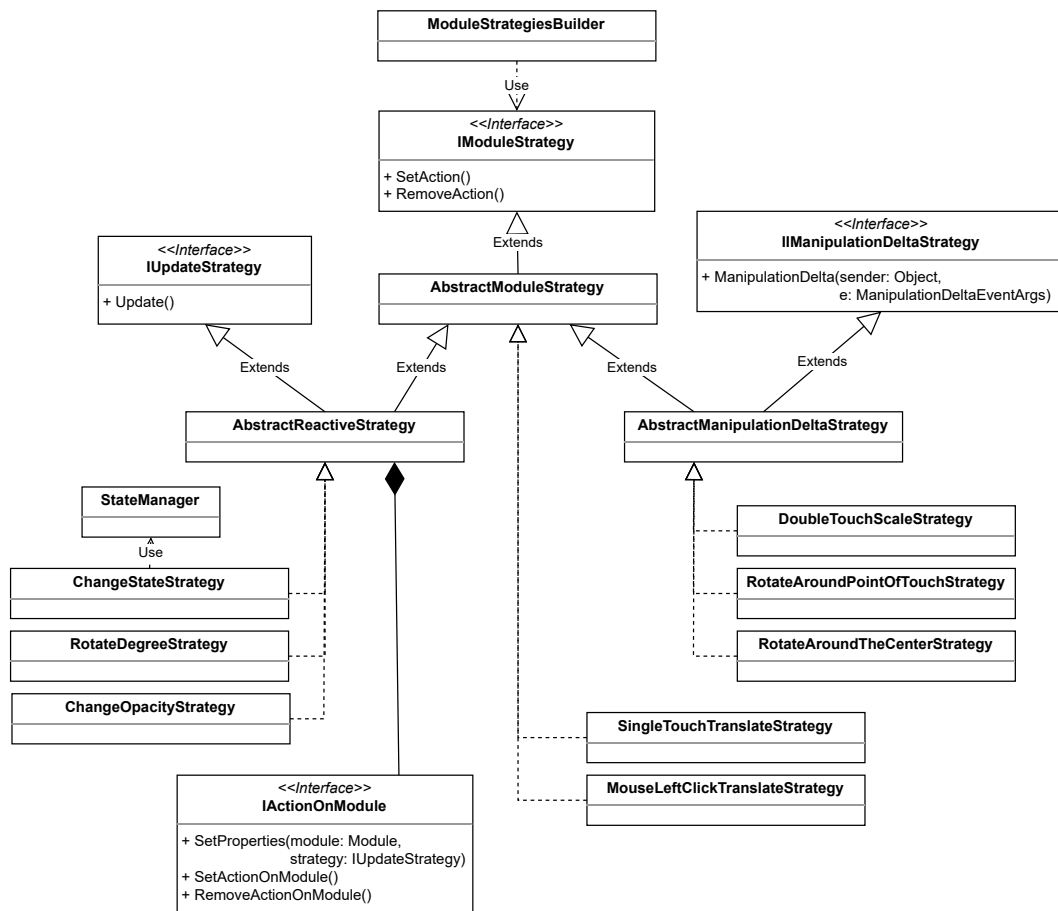


Figura 4.3: Architettura di una funzionalità all'interno di un modulo.

Le strategie attive, che richiedono i valori dati da una manipolazione continua sul modulo tramite il touch, utilizzano l'interfaccia *IManipulationDeltaStrategy*; quelle presentate in fig. 4.3 sono le seguenti:

- **DoubleTouchScaleStrategy**: inserisce all'interno del modulo una funzionalità per la ricezione del doppio tocco e successivo scaling dell'oggetto;
- **RotateAroundPointOfTouchStrategy**: permette la rotazione attorno al punto di tocco che viene mantenuto immobile;
- **RotateAroundTheCenterStrategy**: permette la rotazione attorno al centro del modulo;
- **SingleTouchTranslateStrategy**: tramite il touch è possibile traslare il modulo a proprio piacimento;

- **MouseLeftClickTranslateStrategy**: muovendo il mouse mentre si tiene premuto il pulsante sinistro, è possibile traslare il modulo a proprio piacimento.

Scendiamo ora più nel dettaglio delle strategie reattive e la loro effettiva differenza da quelle attive.

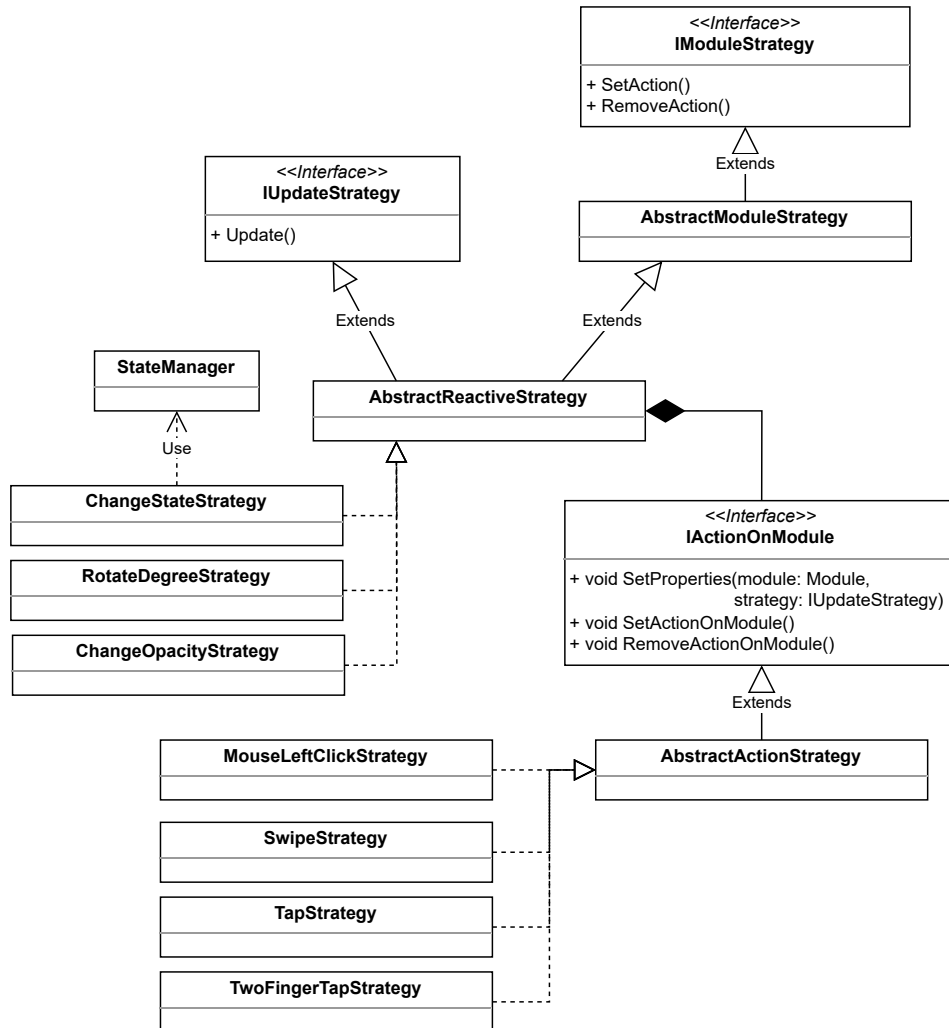


Figura 4.4: Architettura di una strategia reattiva all'interno di un modulo.

Come illustrato in fig. 4.4, le strategie reattive, che per la loro esecuzione attendono un evento specifico, si compongono di un *IActionOnModule*, che rappresenta l'azione dell'utente da verificare. Una volta che l'evento si è verificato, verrà chiamato il metodo *Update()* dell'interfaccia *IUpdateStrategy* che eseguirà un comportamento specifico all'interno del modulo, come ad

esempio la modifica di opacità o la rotazione a un certo angolo attraverso la *ChangeOpacityStrategy* o la *RotateDegreeStrategy*.

In questo modo, ogni strategia reattiva può essere associata a una qualunque azione, che sia il click tramite il mouse o uno swipe verso una certa direzione. Questa decisione è a discrezione del programmatore e di come vuole impostare il modulo.

## 4.4 Cambio di stato

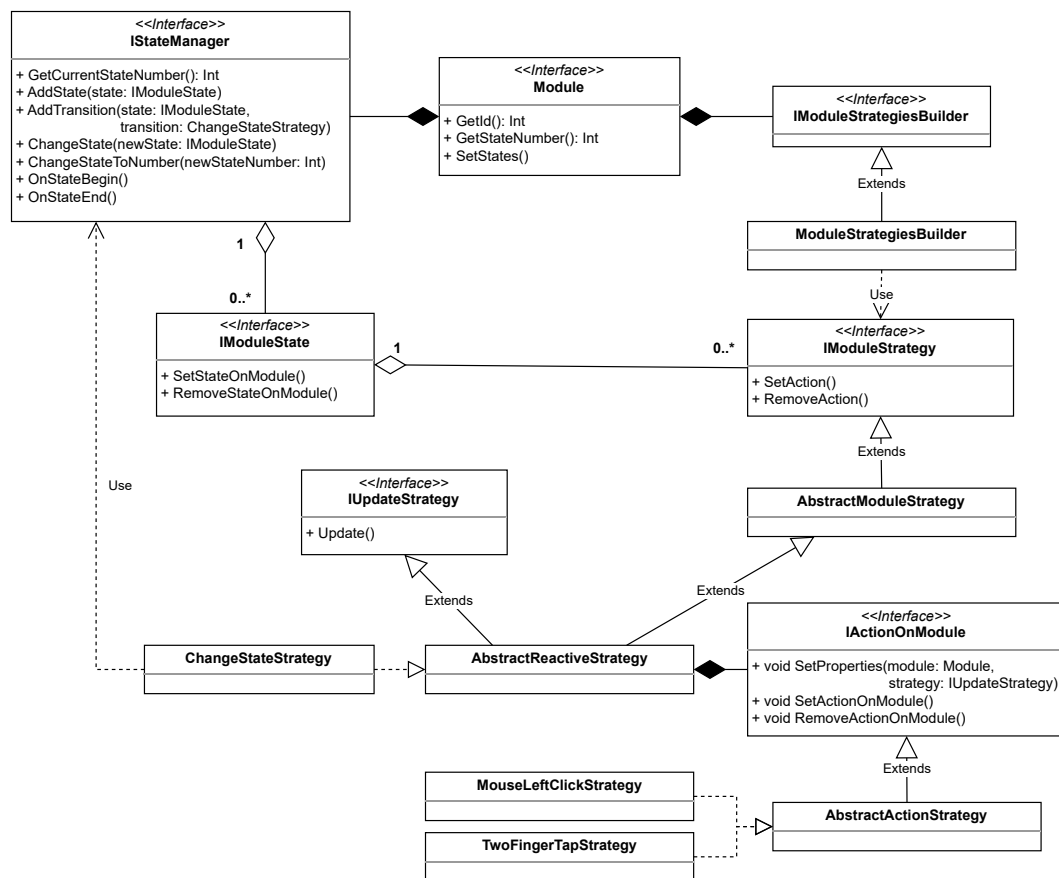


Figura 4.5: Architettura di un modulo con focus sul cambio di stato dato dalla *ChangeStateStrategy*.

La parte fondamentale dell'intera gestione degli stati all'interno di un modulo è rappresentata dalla strategia reattiva *ChangeStateStrategy*. Tale strategia è accoppiata a un'azione, come il click sinistro del mouse o un tap con due dita; ciò abilita le transizioni dello *State Manager* e quindi il passaggio di stato del modulo.

Questo funzionamento è rappresentato nella fig. 4.5 e permette, senza aggiungere ulteriori astrazioni logiche, di riutilizzare il design delle strategie reattive per applicarlo al cambio di stato.

In un modulo è quindi possibile definire, tramite il builder, le varie strategie che ne determinano le funzionalità. Tali strategie sono presenti all'interno di ogni singolo stato e, l'abilitazione di una transizione da uno stato all'altro è determinata dall'azione specificata all'interno del *ChangeStateStrategy*.

## 4.5 Workstation

Come da specifiche iniziali, ogni schermo può utilizzare fino a un massimo di quattro postazioni e ognuna di esse è il punto in cui si svolge il lavoro di uno specialista durante una sessione. Sulla base delle specifiche e dei requisiti necessari per ottenere un prodotto che migliori i briefings medici, si è deciso di comporre la postazione di molteplici pagine, dall'utilizzo differente sulla base delle funzionalità definite durante le specifiche.

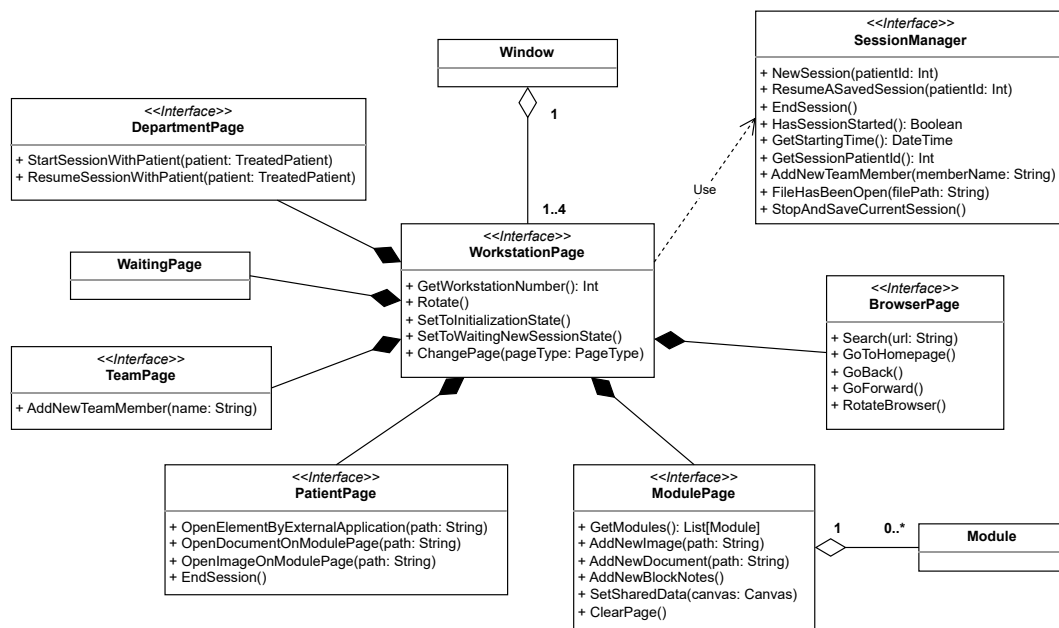


Figura 4.6: Composizione di una postazione basata sui diversi impieghi di essa.

L'utilizzo e la visualizzazione di alcune pagine sono determinati dallo stato della sessione e dall'impiego o meno dell'altro schermo.

La Department Page, in cui si sceglie il paziente su cui avviare una nuova sessione, è visualizzabile solo in uno dei due schermi, mentre l'altro rimane in uno stato di attesa. Di conseguenza, all'apertura dell'applicazione, viene

effettuato, in maniera automatica, un controllo sulla presenza o meno dell'altro schermo (analizzeremo questo aspetto in maniera dettagliata nella prossima sezione); se questa verifica dà esito positivo, lo schermo si mette in stato di attesa, definita *WaitingPage*, nella quale si aspetta l'inizio di una sessione, altrimenti viene mostrata la *DepartmentPage*. Per tutta la durata della sessione, lo specialista può navigare in qualunque pagina desideri o della quale abbia necessità, fatta eccezione per le due pagine precedentemente citate, che tornano attive, nei rispettivi schermi, solo quando viene richiesta la fine della sessione all'interno della *PatientPage*.

In fig. 4.6 si può inoltre osservare la presenza del *SessionManager* che gestisce e manipola tutti i dati di una sessione. Questo gestore è utilizzato attivamente da tutte le postazioni per produrre il verbale d'equipe finale e ogni modifica porta alla condivisione di tali cambiamenti con l'altro schermo.

## 4.6 Condivisione dei dati tra schermi

L'utilizzo contemporaneo dei due schermi presuppone uno scambio di dati tra essi. Al fine di promuovere l'usabilità lato utente, si è cercato di agevolare l'inizializzazione della connessione; infatti, non è richiesto alcun input da parte dello specialista in quanto, tramite un sistema di publish e discover dei device, ogni schermo raccoglie le informazioni necessarie a stabilire la connessione. Tale procedimento destinato al **DeviceDiscover**, è descritto in maniera più dettagliata nella sezione 6.4.

La connessione tra gli schermi è instaurata dal **ConnectionManager** una volta noti l'indirizzo IP e la porta dell'altra macchina. Al fine di definire quale schermo può scegliere il paziente su cui lavorare e quale dovrà rimanere in attesa dell'avvio della sessione, il primo schermo a connettersi alla rete svolgerà la funzione di server mentre il secondo quella di client.

Stabilita la connessione, la classe adibita alla creazione e analisi dei messaggi ricevuti è il **MessageManager**. Ogni azione all'interno di una singola postazione, da ambo gli schermi, che altera lo stato della sessione, porta alla creazione di un messaggio specifico; la fig. 4.7 presenta una versione semplificata di tale gestore con i messaggi ritenuti più importanti.

Il passaggio in rete di ogni messaggio richiede una serializzazione e, successivamente, una deserializzazione di esso. Per semplificare il procedimento di analisi di un messaggio arrivato, le informazioni che si vogliono inviare vengono racchiuse all'interno di un oggetto di tipo **Message**. Questa classe presenta due metodi:

- *GetData()*: torna, sotto forma di stringa, le informazioni che sono state serializzate e inserite alla creazione del messaggio;

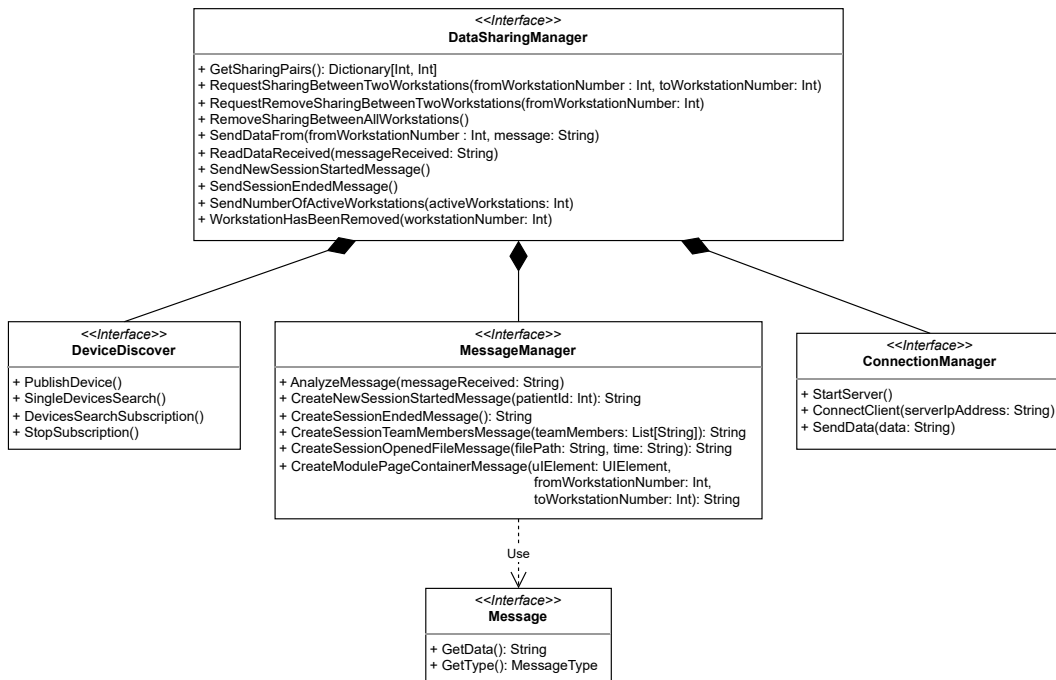


Figura 4.7: Gestione della condivisione dei dati tra i due schermi.

- *GetType()*: rappresenta il tipo dei dati racchiusi all'interno del messaggio per poterli deserializzare nel modo appropriato.

Il gestore dei messaggi, perciò, serializza le informazioni per creare il messaggio da inviare; successivamente, alla ricezione di un nuovo messaggio, ne analizza il tipo per poi deserializzarlo correttamente e applicare quanto richiesto.

Alla base di questa architettura per la condivisione dei dati è presente il **DataSharingManager**, che funge da intermediario tra le varie componenti dell'applicazione, come il *SessionManager* e la *ModulePage*.

# Capitolo 5

## Interazione

In questo capitolo illustriamo alcuni flussi d'interazione tra i due schermi multi-touch e i dati che essi si scambiano.

### 5.1 Inizializzazione di una sessione

All'avvio del software in uno dei due schermi, l'applicazione si trova nella fase d'inizializzazione, in cui viene effettuata una device discovery per controllare se l'altro schermo è attivo o meno. Se la ricerca non ha successo, significa che non è ancora stato pubblicato nessun device, in tal caso se ne crea uno nuovo e, dopo la sua pubblicazione in rete, la macchina si inizializza come server, viceversa se non viene trovato il device dell'altro schermo, significa che esso è attivo e già operativo.

Al verificarsi di quest'ultimo caso e utilizzando le informazioni ottenute, il primo schermo si connette all'indirizzo del server come client e si mette in attesa dell'inizio di una nuova sessione. In questo stato, dove entrambe le applicazioni sono in grado di comunicare, vi sono due possibili stati del server:

1. Sessione già iniziata: gli specialisti hanno già iniziato a lavorare su uno dei due schermi e perciò i dati correnti della sessione vengono immediatamente inviati non appena si stabilisce la connessione;
2. Sessione non ancora iniziata: il secondo schermo è ancora nella pagina del dipartimento per la scelta del paziente. I dettagli della sessione verranno inviati non appena questa avrà inizio.

Ottenuti nel primo schermo tutti i dati relativi allo stato corrente della sessione, questa viene inizializzata rendendo operativa l'applicazione e passando dalla *WaitingPage* alla *PatientPage*.

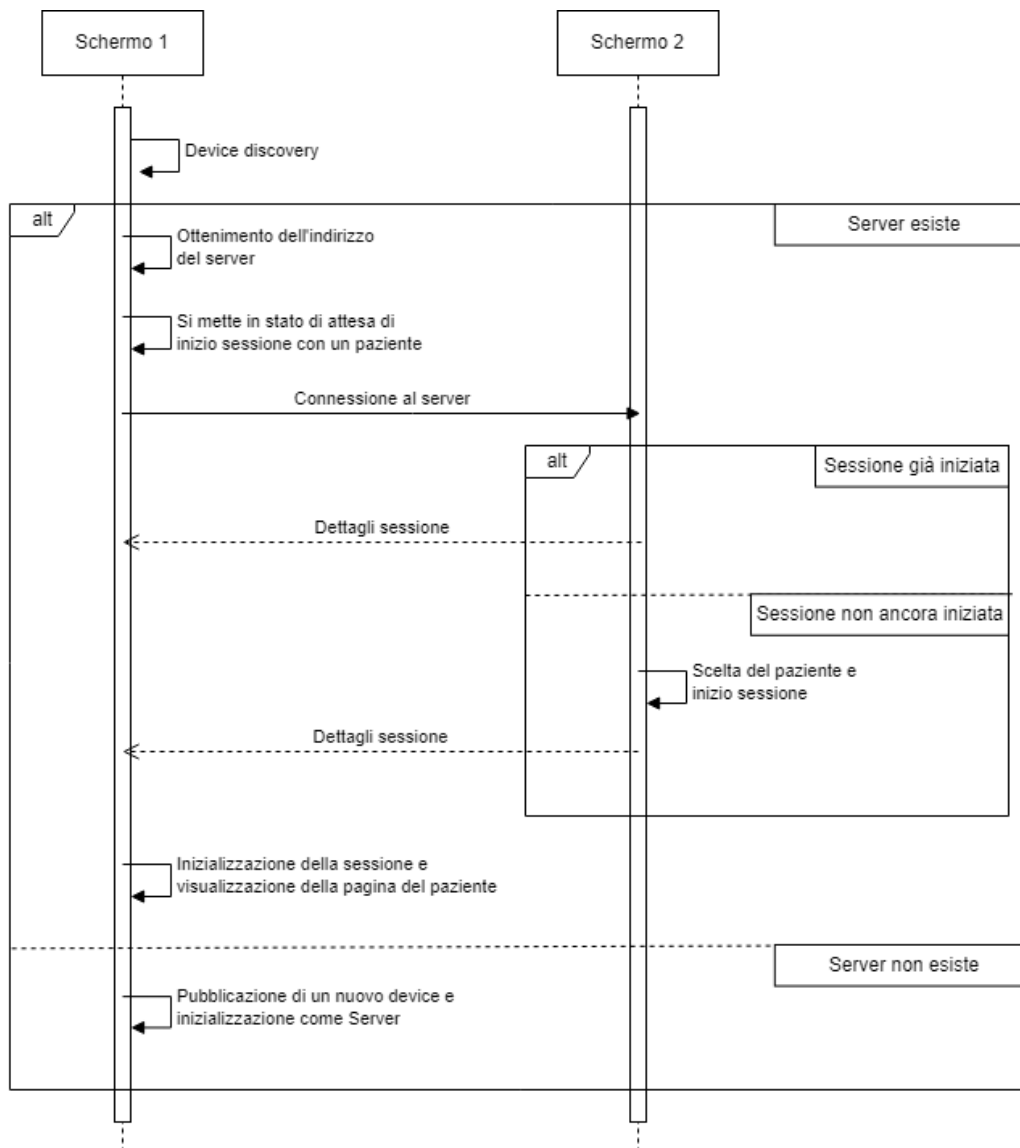


Figura 5.1: Flusso d'interazione che porta all'inizializzazione di una sessione nei due schermi.

## 5.2 Sharing tra due postazioni

Una parte fondamentale che accresce la collaborazione e la cooperazione tra gli specialisti, è la possibilità di condividere e manipolare contemporaneamente il contenuto delle *ModulePage* di due postazioni in schermi differenti.

Se i due schermi sono connessi tra di loro, tramite il pannello di condivisione presente in ogni pagina dei moduli, è possibile vedere in tempo reale quali altre



postazioni sono attive nell'altro schermo per poi decidere con quale di esse effettuare la condivisione. Ogni postazione di uno schermo può essere in stato di sharing con un'altra in modo mutualmente esclusivo; non è perciò possibile per due postazioni essere in stato di condivisione con la stessa postazione dell'altro schermo e quindi il numero massimo di condivisioni attive è pari a quattro, che si verifica quando tutte le postazioni sono in stato di sharing.

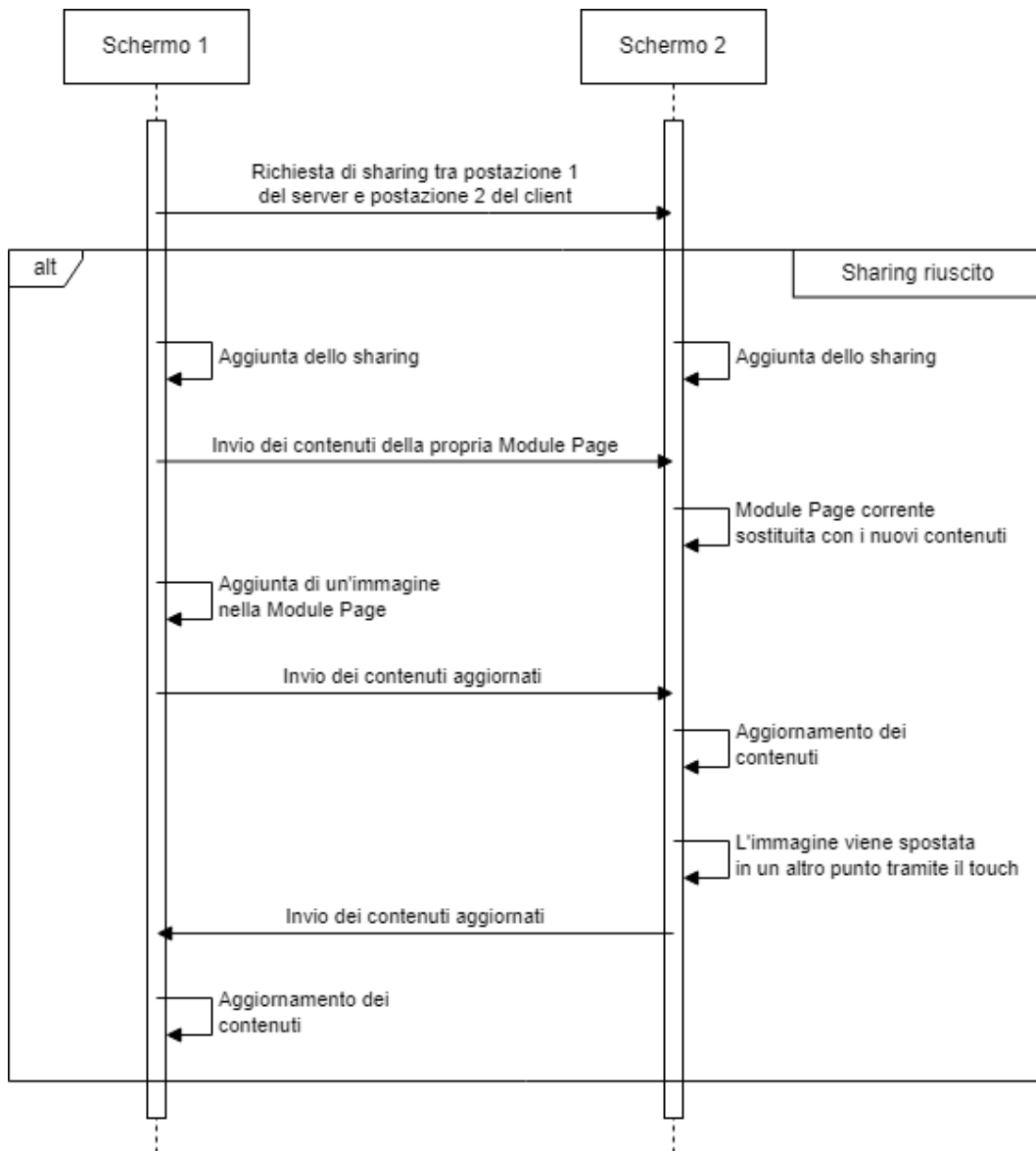


Figura 5.2: Condivisione dei contenuti tra due postazioni.

Come rappresentato nel diagramma di fig. 5.2, lo sharing parte dalla richiesta

di uno dei due schermi. Se non sono sorti problemi, entrambe le applicazioni mantengono memorizzate le postazioni che sono in sharing tra di loro, per rimuoverle da ulteriori richieste. Successivamente, lo schermo che ha inoltrato la richiesta è quello che invia i contenuti della propria Module Page, sovrascrivendo i dati dell'altra.

Da qui in poi, qualunque modifica all'interno della Module Page, che avvenga in una postazione o nell'altra, verrà intercettata e condivisa con l'altro schermo per mantenere i contenuti sempre aggiornati.

# Capitolo 6

## Dettagli Implementativi

### 6.1 Module State Manager

La complessa architettura di un Module permette la progettazione e la creazione delle sue funzionalità con estrema facilità e, allo stesso tempo, consente di rendere agevole tutto il processo di creazione del suo comportamento. In questo esempio si vuole creare un modulo composto da tre stati:

1. *No Strategy State*: appena creato, il modulo deve trovarsi in uno stato dove nessuna manipolazione è abilitata, ed è possibile la sola transizione per lo stato successivo;
2. *Movement State*: in questo stato si vuole poter interagire con il modulo, ovvero avere la possibilità di ruotarlo attorno al punto di tocco, fare zoom-in e zoom-out con due dita, traslarlo tramite il movimento di un dito o il click sinistro del mouse su di esso. Qui vogliamo far uso delle sole strategie attive;
3. *Clockwise Degree Rotation State*: in quest ultimo stato si dà la possibilità di ruotare in senso orario e per un grado specifico il modulo. Si utilizzeranno solo delle strategie reattive che presentano la singola azione specifica che abilita tale rotazione.

L'utente avrà sempre la possibilità cambiare lo stato del proprio modulo attraverso un tap con due dita o attraverso il clic destro del mouse. Questo cambiamento sarà ciclico passando per ogni stato, per poi ripartire dal primo.

Di seguito viene mostrato il codice che è necessario per implementare le specifiche appena illustrate, che analizzeremo suddividendolo in tre parti:

---

```
//Definition of 3 states
IModuleState noStrategyState = new ModuleState(moduleStrategiesBuilder
    .NoStrategyBuild());

IModuleState movementState = new ModuleState(moduleStrategiesBuilder
    .AddRotateAroundPointOfTouchStrategy()
    .AddDoubleTouchScaleStrategy()
    .AddSingleTouchTranslateStrategy()
    .AddMouseLeftClickTranslateStrategy()
    .Build());

IModuleState clockwiseDegreeRotationState = new
    ModuleState(moduleStrategiesBuilder
    .AddRotateDegreeReactiveStrategy(new TapStrategy(), 90, true)
    .AddRotateDegreeReactiveStrategy(new MouseLeftClickStrategy(),
        -90, true)
    .Build());

//Adding states to the manager
this.stateManager.AddState(noStrategyState);
this.stateManager.AddState(movementState);
this.stateManager.AddState(clockwiseDegreeRotationState);
```

---

Listato 6.1: Creazione degli stati di un modulo

Con questa prima parte di codice creiamo i tre stati sopracitati e le strategie associate a ognuno di essi. Successivamente si definiscono le azioni necessarie per effettuare un cambiamento di stato:

---

```
//Creation of some ChangeStateStrategy, the strategy that contain the
    action that enables the transition from one state to another
ChangeStateStrategy changeToSecondStrategy =
    new ChangeStateStrategy(new TwoFingerTapStrategy(), this,
        movementState, this.stateManager);
ChangeStateStrategy changeToThirdStrategy =
    new ChangeStateStrategy(new TwoFingerTapStrategy(), this,
        clockwiseDegreeRotationState, this.stateManager);
ChangeStateStrategy changeToPrimaryStrategy =
    new ChangeStateStrategy(new TwoFingerTapStrategy(), this,
        noStrategyState, this.stateManager);
```

```
ChangeStateStrategy changeToSecondStrategyTwo =
    new ChangeStateStrategy(new MouseRightClickStrategy(), this,
        movementState, this.stateManager);
ChangeStateStrategy changeToThirdStrategyTwo =
    new ChangeStateStrategy(new MouseRightClickStrategy(), this,
        clockwiseDegreeRotationState, this.stateManager);
ChangeStateStrategy changeToPrimaryStrategyTwo =
    new ChangeStateStrategy(new MouseRightClickStrategy(), this,
        noStrategyState, this.stateManager);
```

---

Listato 6.2: Definizione della strategia per il cambio di stato

Infine si aggiungono le transizioni indicando per ognuna di esse lo stato di partenza e il `ChangeStateStrategy`.

---

```
//Adding the transitions by setting the arrival state and the
    strategy to enable the transition
this.stateManager.AddTransition(noStrategyState,
    changeToSecondStrategy);
this.stateManager.AddTransition(movementState, changeToThirdStrategy);
this.stateManager.AddTransition(clockwiseDegreeRotationState,
    changeToPrimaryStrategy);

this.stateManager.AddTransition(noStrategyState,
    changeToSecondStrategyTwo);
this.stateManager.AddTransition(movementState,
    changeToThirdStrategyTwo);
this.stateManager.AddTransition(clockwiseDegreeRotationState,
    changeToPrimaryStrategyTwo);
```

---

Listato 6.3: Aggiunta delle transizioni che comportano il cambio di stato

Partendo dall'esempio e attraverso poche linee di codice, è subito ravvisabile l'immediata identificazione del comportamento di questo oggetto. Questo è tutto quello che è necessario definire all'interno del modulo per ottenere un comportamento complesso e il programmatore è in grado di modificarlo, aggiungendo o rimuovendo funzionalità, in un lasso di tempo praticamente irrisorio. La potenza di tale architettura risiede nell'elevato numero di funzionalità che il modulo può utilizzare, inoltre, tale struttura, concede la possibilità di avere la stessa tipologia di strategia reattiva con effetti diversi sul modulo a seconda dell'azione che la genera.

Prendendo come esempio la *RotateDegreeReactiveStrategy*, effettuando un tap tramite il touch si avrà una rotazione di 90 gradi in senso orario, mentre se

si effettua un click sinistro con il mouse, avremo una rotazione in senso opposto.

In fig. 6.1 è mostrato il diagramma degli stati che si ottiene tramite il codice sopracitato.

Questo è solo un esempio di quello che l'architettura appena descritta è in grado di supportare; tramite l'aggiunta di nuove strategie e nuove azioni abilitative per queste componenti, ogni modulo potrà avere maggiori gradi di libertà.

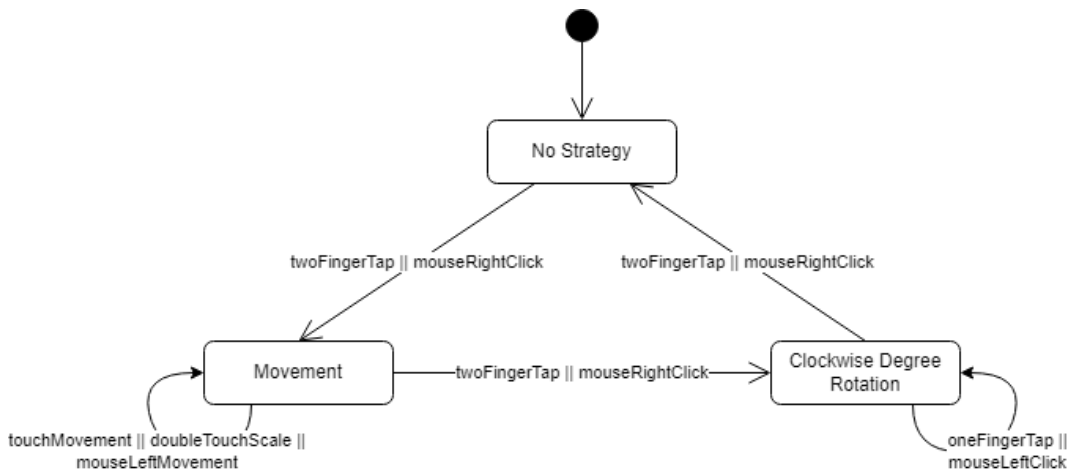


Figura 6.1: Esempio di diagramma degli stati di un modulo.

## 6.2 Implementazione delle Funzionalità

Durante lo sviluppo si è cercato utilizzare al massimo il supporto che WPF permette per gli elementi grafici. Per i moduli si è deciso di estendere la classe nativa *Control*, tramite la quale è stato possibile definire vari stili grafici per ogni tipologia di modulo e implementare singolarmente i vari *Event Handler*.

Come descritto in precedenza, per l'architettura di una funzionalità all'interno di un modulo (fig. 4.3), ogni **ModuleStrategy** presenta due metodi: `SetAction()` e `RemoveAction()`. Questi metodi aggiungono e rimuovono degli specifici Event Handler, associati alle funzioni che determinano la strategia / funzionalità al verificarsi dell'evento.

Per una maggiore chiarezza si prenda come esempio la *SingleTouchTranslateStrategy* il cui scopo è traslare il modulo tramite un singolo touch. Il modulo, estendendo la classe *Control*, presenta la possibilità di definire un handler per gestire l'evento *TouchMove*; in questo modo non è necessario definire il come rilevare un evento, ma solo cosa sia richiesto per eseguire correttamente tale funzionalità. Per tale motivo è stato definito, all'interno della strategy un metodo privato, che stabilisce come manipolare la posizione del modulo sulla

base della posizione del dito e della direzione del movimento su di esso; il `SetAction()` e `RemoveAction()` semplicemente aggiungono e rimuovono tale metodo dal `TouchMove` definito per estensione della classe `Control`.

I metodi per l'aggiunta e la rimozione di tali strategie vengono chiamati alla `OnStateBegin()` e alla `OnStateEnd()` durante i cambi di stato nello `State Manager`.

---

```
public class SingleTouchTranslateStragety : AbstractModuleStrategy
{
    //Constructor
    SingleTouchTranslateStragety(CustomControls.Module module) :
        base(module) {}

    public override void SetAction()
    {
        this.module.TouchMove += TouchMove;
    }

    public override void RemoveAction()
    {
        this.module.TouchMove -= TouchMove;
    }

    private void TouchMove(object sender, TouchEventArgs e)
    {
        //Implementation
    }
}
```

---

Listato 6.4: Esempio di implementazione di una strategia attiva

Le strategie reattive sono più complesse rispetto a quelle attive in quanto integrano l'azione/l'evento che deve verificarsi per abilitare tali strategie, anche se la base è pressoché identica. In particolare, le strategie reattive si differenziano per due aspetti:

- `SetAction()` e `RemoveAction()` aggiungono e rimuovono l'azione abilitante e non la funzionalità interna alla strategia;
- vi è un metodo aggiuntivo chiamato `Update()` che esegue la strategia. Questo viene chiamato nel momento in cui l'azione avviene nel modulo.

Lo scopo delle azioni è racchiudere un tipo di evento che può verificarsi sul modulo, per poi chiamare l'`Update()` della strategia reattiva a cui sono associate.

La chiamata dell'Update() applicherà la strategia reattiva che aggiornerà il modulo e, se è presente uno sharing con un'altra postazione, tale aggiornamento verrà propagato anche sull'altro schermo.

---

```
public class MouseLeftClickStrategy : AbstractActionStrategy
{
    public override void SetActionOnModule()
    {
        this.module.PreviewMouseLeftButtonDown += MouseLeftButtonDown;
    }

    public override void RemoveActionOnModule()
    {
        this.module.PreviewMouseLeftButtonDown -= MouseLeftButtonDown;
    }

    private void MouseLeftButtonDown(object sender,
        MouseButtonEventArgs e)
    {
        //Calls the Update() method of the reactive strategy that this
        //action is associated with
        this.strategy.Update();
    }
}
```

---

Listato 6.5: Esempio di implementazione di una Action

Espandendo rispetto alle architetture presentate nel capitolo sul design, successivamente sono elencate tutte le strategie e le azioni implementate nel prototipo:

- **Active Strategies:**

- *Single Touch Translate Strategy*: effettua una traslazione a seconda del movimento di un dito;
- *Mouse Left Click Translate Strategy*: effettua una traslazione a seconda del movimento del mouse mentre si preme il tasto sinistro;
- *Double Touch Scale Strategy*: ingrandisce se si allontanano le due dita o rimpicciolisce se si avvicinano;
- *Rotate Around Point Of Touch Strategy*: rotazione tramite due dita, la rotazione avviene attorno al dito che rimane fermo;
- *Rotate Around Center Strategy*: rotazione tramite due dita, la rotazione avviene sempre attorno al centro del modulo;



- **Reactive Strategies:**
  - *Change Opacity Strategy*: al verificarsi dell'azione desiderata, la trasparenza del modulo viene modificata;
  - *Change State Strategy*: al verificarsi dell'azione desiderata, il modulo effettua un cambio di stato;
  - *Rotate Degree Strategy*: al verificarsi dell'azione specificata, il modulo ruota in senso orario o antiorario dei gradi specificati;
- **Actions:**
  - *Tap Strategy*: si verifica al rilevamento di un tap sul modulo;
  - *Two Finger Tap Strategy*: si verifica al rilevamento di un doppio tap;
  - *Swipe Strategy*: è possibile decidere quale direzione dello swipe scaturisce l'update;
  - *Mouse Left Click Strategy*: avviene al click sinistro del mouse;
  - *Mouse Right Click Strategy*: si verifica al click destro del mouse;
  - *Mouse Wheel Click Strategy*: avviene al click della rotella del mouse.

## 6.3 Moduli Implementati

All'interno della *Module Page* può essere inserito un numero indefinito di moduli. Per differenziarli tra quelli dello stesso tipo, ogni modulo presenta un identificativo numerico che viene generato casualmente e controllato alla creazione.

Come spiegato nella sezione precedente, ogni modulo presenta una parte implementativa nella quale sono definiti gli stati e le transizioni e una parte grafica scritta in XAML [23].

Per l'**InkCanvasModule** si è scelto di mantenere un solo stato senza alcuna strategia al suo interno. Questa decisione è stata presa per coprire interamente la superficie della *ModulePage* permettendo di scrivere sopra di esse tramite il touch, ma anche di cancellare ed effettuare selezioni dei tratti per spostarli e ridimensionarli.

Più complessi sono invece **DocumentModule**, **MovingImageModule** e **BlockNotesModule** che presentano tutti la medesima struttura principale. Essi sono composti da due stati:

- il primo senza strategia, per avere l'elemento in solo stato di visualizzazione;

- il secondo, invece, con la possibilità ruotare il modulo tramite il touch, ma anche di scalarlo e muoverlo a piacimento sia tramite il touch sia con il mouse.

In tal modo questi contenitori possono essere posizionati e scalati nel modo ritenuto più opportuno per poi cambiare lo stato in sola visualizzazione.

L'utilizzo del *BlockNotesModule* è peculiare rispetto agli altri in quanto necessita un input testuale da parte dell'utente. Questo ha portato all'implementazione di una tastiera virtuale all'interno di ogni *Module Page* che viene abilitata quando il blocco note di una postazione ottiene il focus. Se sono presenti più moduli appartenenti a questa tipologia, la tastiera virtuale convertirà il touch dell'utente in input testuale solo nel block notes che ha ricevuto il focus per ultimo.

Le strategie reattive, con esclusione del Change State, vengono invece utilizzate nel **RotatingImageModule** che dà la possibilità, tramite un tap o click del mouse, di ruotare di esattamente 90 gradi in senso orario nel primo stato e in senso antiorario nel secondo.

Ogni modulo presenta inoltre un bordo che cambia colore a seconda dello stato in cui si trova; questo serve a dare un immediato riscontro all'utente sull'effettivo cambio di stato.

## 6.4 Publishing and Device Discovering

Per condividere i dati tra i due schermi è necessario che le applicazioni instaurino una connessione tra di loro, per fare ciò è richiesta una prima fase d'inizializzazione nella quale si ottiene l'indirizzo IP dell'altro schermo. Poiché i messaggi viaggiano attraverso la medesima rete tramite il protocollo TCP [24] e l'indirizzo è variabile, per non avere ritardi in un campo delicato come la cura di un paziente e la medicina, dovuti a una immissione manuale da parte degli specialisti a ogni accensione del software, tutta la parte riguardante il discovery degli indirizzi IP è fatta automaticamente senza la necessità di intervento da parte dell'utente. Per questa fase si è fatto uso del protocollo RSSDP (Really Simple Service Discovery Protocol) [25], ovvero un'implementazione .Net del protocollo Simple Service Discovery (SSDP) [26] che fa parte dello standard Universal Plug and Play (UPnP). SSDP consente di rilevare dispositivi e servizi su una rete (locale). Tramite questo protocollo si è in grado di eseguire il publish del proprio device che:

- presenta un codice univoco per distinguerlo dagli altri device;
- contiene al suo interno l'indirizzo IP della macchina.

In questo modo l'applicazione, durante la fase d'inizializzazione effettua una ricerca dell'altro dispositivo all'interno della sua rete. L'altro device viene rilevato controllando la presenza di un codice specifico, il quale è uguale per entrambi gli schermi. Se la ricerca da esito positivo, dalle informazioni del device trovato si ottiene l'IP dell'altra macchina e si fa partire una richiesta di connessione, altrimenti si effettua il publish di un nuovo device con il proprio indirizzo IP.

## 6.5 Session Management

Tutti le componenti di una sessione, esclusa la `ModulePage`, vengono gestite dal **SessionManager**, il quale è univoco per singola applicazione.

Visto che ogni sessione presenta un timer che parte da 0 e determina la durata totale della sessione stessa, la quale deve essere uguale in entrambi gli schermi, si è deciso di memorizzare solamente l'orario di partenza per poi dividerlo. La durata si ottiene semplicemente sottraendo l'ora attuale da quella iniziale.

Al termine di una sessione essa viene salvata per avere la possibilità di riprenderla in futuro e si produce un report finale.

Per ogni sessione vengono salvate, sotto forma di file:

- la durata totale;
- la diagnosi testuale del paziente;
- i nomi di tutti i membri del team partecipanti;
- la lista di tutti i nomi dei file aperti accoppiati con l'orario di apertura;
- il contenuto grafico delle `ModulePage` di tutte le postazioni.

In questo modo, alla ripresa di una sessione salvata, tutte le pagine in ogni postazione saranno uguali allo stato precedente, fatta eccezione per la `Browser Page` della quale non viene salvato il contenuto.

Le stesse informazioni, esclusi i contenuti grafici, vengono poi utilizzate per la produzione finale di un report pdf riguardante il paziente. Come si può vedere nell'esempio in figura 6.2, questo è il verbale d'equipe che il prototipo produce, e questo, in futuro, potrà essere espanso e messo a norma.



Figura 6.2: Esempio di verbale prodotto alla fine della sessione con un paziente.

## 6.6 Serializzazione e Deserializzazione

L'invio e ricezione dei messaggi tra le applicazioni aperte sui due schermi sono state effettuate utilizzando le librerie di sistema *System.IO* e *System.Net* con una applicazione che funge da server e l'altra da client. La comunicazione avviene da entrambi i lati attraverso stream di stringhe all'interno del canale. Per poter gestire lo scambio di informazioni in modo efficace e allo stesso tempo avere un'architettura facilmente espandibile, i dati sono racchiusi all'interno di strutture dati specifiche per ogni tipologia di messaggio. Queste vengono inizialmente convertite in stringhe e memorizzate all'interno della classe **Message** unitamente al loro tipo originale, in un campo `Data` e `TypeOfMessage` rispettivamente. Il procedimento iniziale perciò consiste nella creazione di un determinato tipo di messaggio che poi viene serializzato in stringa `Json` attraverso *System.Text.Json* e infine inserito in un oggetto `Message`. Successivamente, il `Message` viene a sua volta convertito in stringa `JSON` per poter essere inviato correttamente in rete.

Al momento della ricezione, il messaggio viene riconvertito in `JSON` e si tenta di istanziarlo all'interno di un `Message`. Se questo non porta ad una eccezione, si esegue una ulteriore estrazione della stringa contenuta per riottenere il tipo di messaggio suggerito dal campo `TypeOfMessage`.

Questi procedimenti avvengono tra il **Message Manager** che si occupa della creazione ed analisi dei messaggi ricevuti e il **Connection Manager**, adibito all'invio e ricezione dei dati sotto forma di stringhe.

La serializzazione e deserializzazione avviene anche per tutti i processi che riguardano il salvataggio e la ripresa di una sessione di un paziente. In modo simile, tutti i dati di una sessione vengono memorizzati all'interno di

un oggetto **SavedSession** che viene convertito in stringa JSON attraverso il **JsonSerializer** per poi essere salvata all'interno di un file di testo. Alla ripresa di una sessione, sulla base dell'id associato al paziente, si riottiene il file inerente alla sua ultima sessione e, dal testo in esso contenuto, si recupera l'oggetto **SavedSession** con tutte le informazioni in esso contenute.

## 6.7 Interfaccia Utente

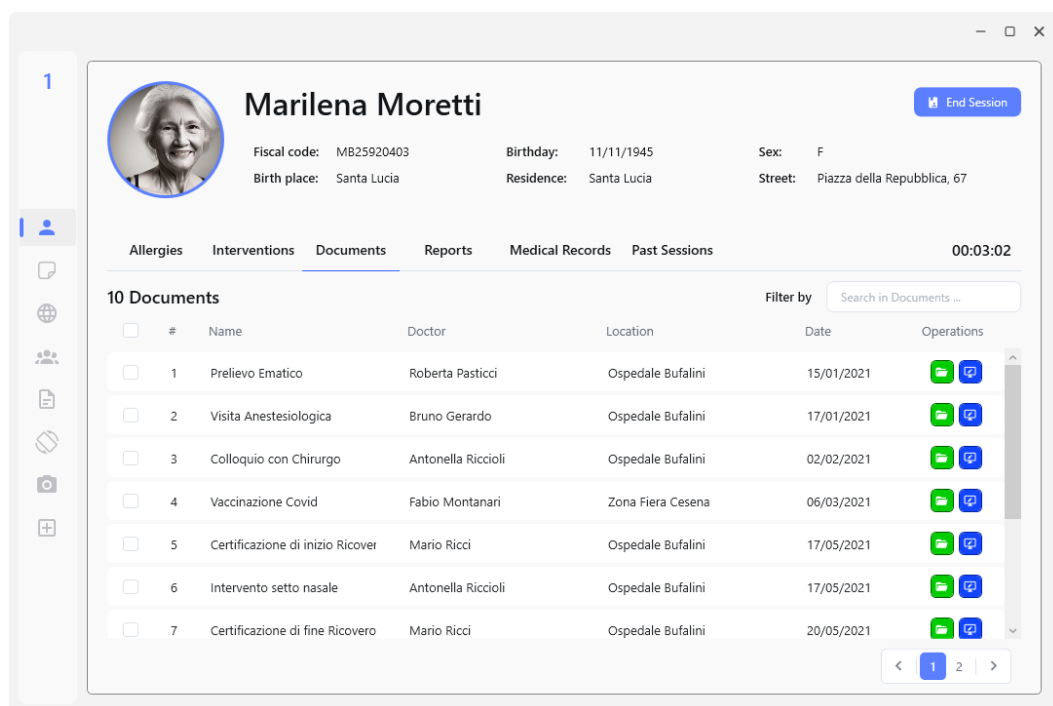


Figura 6.3: Grafica di esempio di una singola postazione.

Data l'area d'interesse per lo sviluppo del prototipo e il supporto nativo della gestione di eventi touch, data dal sistema operativo Windows, l'applicazione è stata implementata sul framework .Net utilizzando WPF (Windows Presentation Foundation) e scritta in C#.

Come base per l'applicazione, sono stati utilizzati diversi componenti del sottosistema grafico WPF, la cui appearance è stata modificata singolarmente all'interno dello *XAML* o attraverso l'uso di *Resource Dictionary*, per dare all'applicazione un look più moderno.

Tutte le icone sono state ottenute attraverso *Material Design Icons - Pictogrammers* e tramite il NuGet package *MahApps.Metro.IconPacks*. In segui-

to, ognuna di esse è stata ridimensionata e colorata nel modo ritenuto più opportuno.

Inoltre, com'è ravvisabile in figura 6.3, si è scelto di usare due tonalità di colori per l'intera applicazione: il bianco e il blu.

# Capitolo 7

## Utilizzo dell'Applicativo

Il prototipo consiste di una applicazione software utilizzabile sui sistemi operativi Windows. L'applicativo è lo stesso in entrambi gli schermi e, per un corretto utilizzo, è necessario che entrambi i dispositivi siano collegati alla stessa rete.

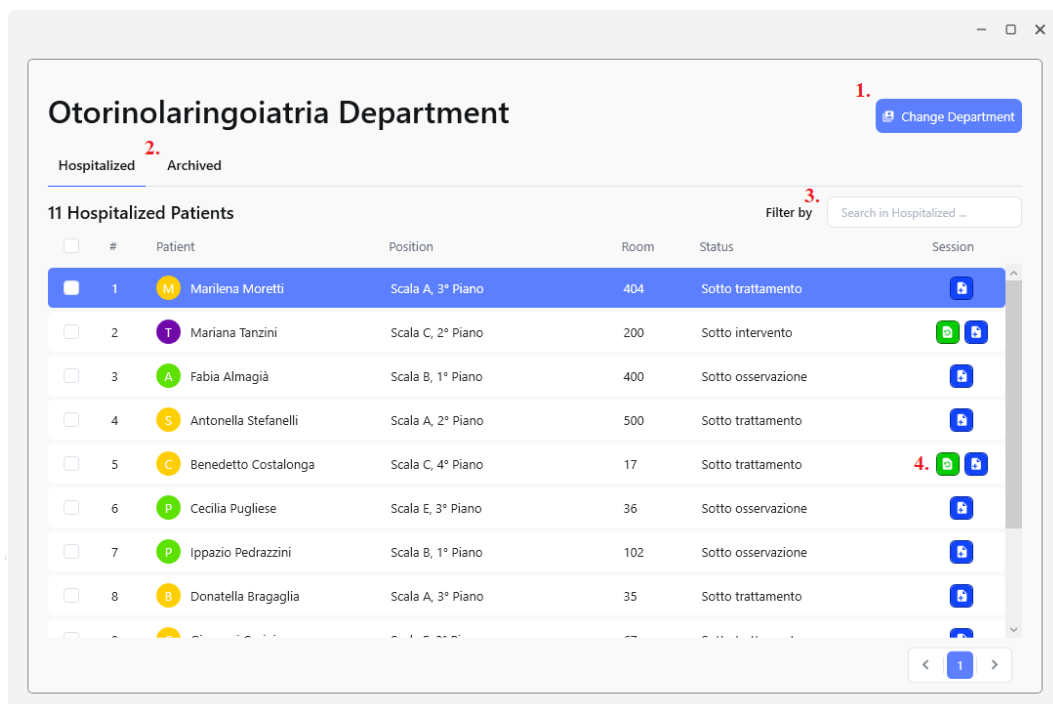


Figura 7.1: Pagina riservata a uno specifico dipartimento per la scelta del paziente su cui iniziare una sessione.

All'avvio dell'applicazione, dopo una prima fase di inizializzazione in cui il software controlla se l'applicazione nell'altro schermo è già attiva, il team si

trova davanti alla *Department Page*. Come è possibile vedere in fig. 7.1, questa pagina fornisce una prima vista sui pazienti bisognosi di cure all'interno di uno specifico dipartimento. Il team di medici da qui può:

1. cambiare dipartimento, per visualizzare i propri pazienti;
2. scegliere un paziente tra quelli attualmente in fase di cura o quelli archiviati;
3. cercare uno specifico paziente;
4. riprendere l'ultima sessione effettuata con un paziente o iniziarne una nuova.

Il paziente viene scelto solo all'interno del primo schermo, il secondo perciò, avendo avviato l'applicazione in un secondo momento, si mette in attesa dell'inizio di una sessione con un paziente.

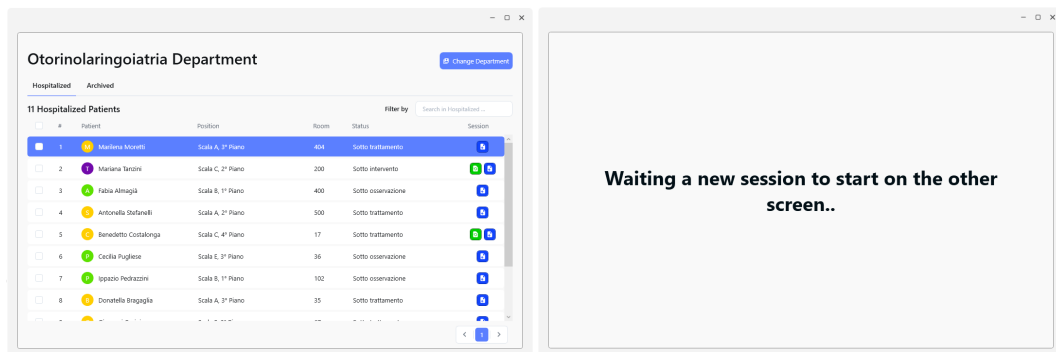


Figura 7.2: Applicazione avviata in entrambi gli schermi mentre si sta scegliendo il paziente.

Iniziata una sessione, in entrambi gli schermi sarà visualizzata la pagina con tutte le informazioni, immagini e documenti del paziente in cura, fig. 7.3. Nel menù posto a lato, è possibile cambiare pagina o effettuare altre azioni all'interno della propria postazione senza che avvenga alcuna variazione nelle altre. Le scelte che tale menù permette sono:

1. visualizzare le informazioni del paziente;
2. accedere alla pagina dalla quale manipolare immagini e documenti;
3. effettuare una ricerca su internet;
4. aggiungere i nominativi dei membri del team;



5. scrivere il verbale d'equipe;
6. ruotare la visualizzazione della postazione;
7. effettuare uno screenshot;
8. aggiungere una nuova postazione.

The screenshot shows a web interface for a patient's medical records. At the top, there is a patient profile for Marilena Moretti, including a photo and personal details like fiscal code, birthday, sex, birth place, residence, and street. Below this is a navigation menu with tabs for Allergies, Interventions, Documents, Reports, Medical Records, and Past Sessions. A timer shows 00:03:02. A table lists 10 documents with columns for #, Name, Doctor, Location, Date, and Operations. A search filter is available for the documents. Red annotations 1-6 point to the patient name, the navigation menu, the timer, the 'End Session' button, the filter dropdown, and the 'Operations' column of the document table respectively.

#	Name	Doctor	Location	Date	Operations
1	Prelievo Ematico	Roberta Pasticci	Ospedale Bufalini	15/01/2021	6. [Icons]
2	Visita Anestesiologica	Bruno Gerardo	Ospedale Bufalini	17/01/2021	[Icons]
3	Colloquio con Chirurgo	Antonella Riccioli	Ospedale Bufalini	02/02/2021	[Icons]
4	Vaccinazione Covid	Fabio Montanari	Zona Fiera Cesena	06/03/2021	[Icons]
5	Certificazione di inizio Ricovero	Mario Ricci	Ospedale Bufalini	17/05/2021	[Icons]
6	Intervento setto nasale	Antonella Riccioli	Ospedale Bufalini	17/05/2021	[Icons]
7	Certificazione di fine Ricovero	Mario Ricci	Ospedale Bufalini	20/05/2021	[Icons]

Figura 7.3: Pagina relativa alle informazioni e documenti del paziente.

Per quanto riguarda invece la pagina relativa al paziente:

1. sono presenti le informazioni anagrafiche del paziente;
2. è possibile cambiare sezione a seconda del tipo di dato da visualizzare;
3. viene mostrato un timer che indica da quanto tempo è iniziata la sessione;
4. vi è la possibilità di terminare la sessione in qualunque momento. In tal caso la sessione viene salvata e con essa i contenuti di ogni postazione, inoltre, sulla base del verbale redatto, viene prodotto il report del paziente. Successivamente, dalla Department Page, è possibile riprendere la sessione nel caso non sia stata terminata;

5. all'interno di una sezione si possono filtrare gli elementi per nome;
6. per ogni elemento sono disponibili due tipi di operazioni:
  - 6.1 l'apertura tramite un'applicazione esterna;
  - 6.2 l'apertura all'interno della *Module Page* dove è possibile manipolare l'elemento tramite il touch.

Nel caso sia necessario effettuare una ricerca su internet, per ottenere informazioni aggiuntive o accedere a siti medici specifici, si utilizzerà la *Browser Page*.

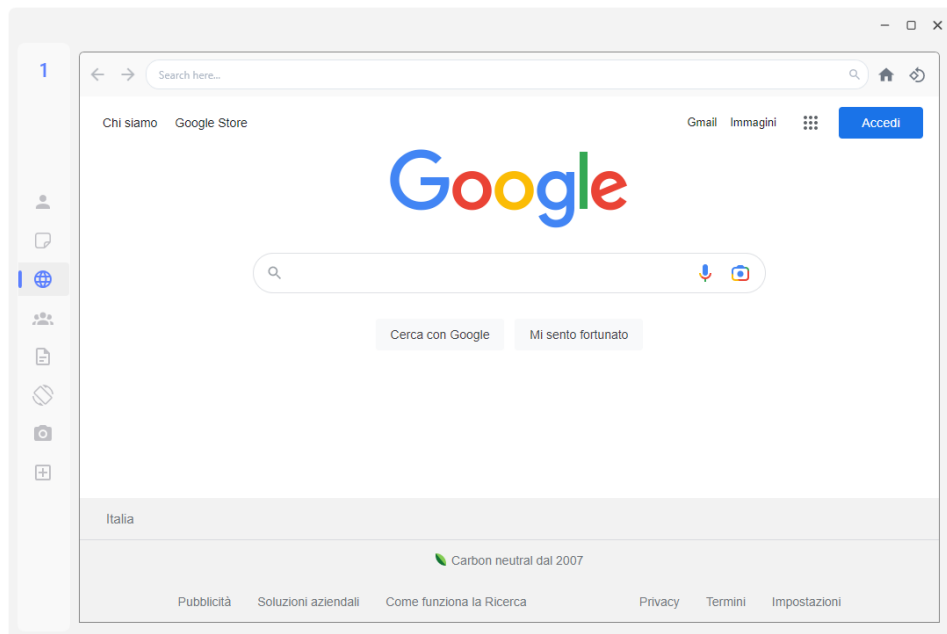


Figura 7.4: Piccolo browser incorporato all'interno del prototipo.

In fig. 7.4 è mostrata la homepage, nella quale inserire le parole chiave per effettuare le ricerche. Inoltre, da sinistra verso destra, nella barra orizzontale in alto, è possibile:

1. andare avanti o indietro nella ricerca, tramite le apposite frecce;
2. digitare l'url di un sito web per entrare direttamente nel sito, o una frase per una normale ricerca;
3. tornare alla homepage;
4. ruotare di 90 gradi la visualizzazione della pagina web corrente.

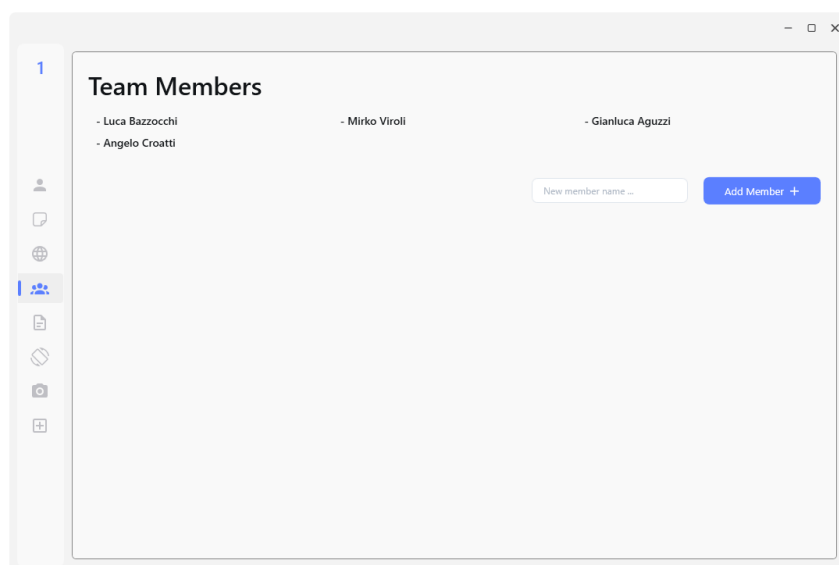


Figura 7.5: Pagina in cui sono indicati i nominativi dei membri del team.

All'interno del report finale sono presenti anche i nominativi di tutti gli specialisti che hanno preso parte alla sessione. Questi nominativi vengono inseriti all'interno della *Team Members Page*, la quale, come la *Report Page* è condivisa tra tutte le postazioni di entrambi gli schermi, in modo tale che, una eventuale modifica da una parte, sia condivisa con tutto il sistema.

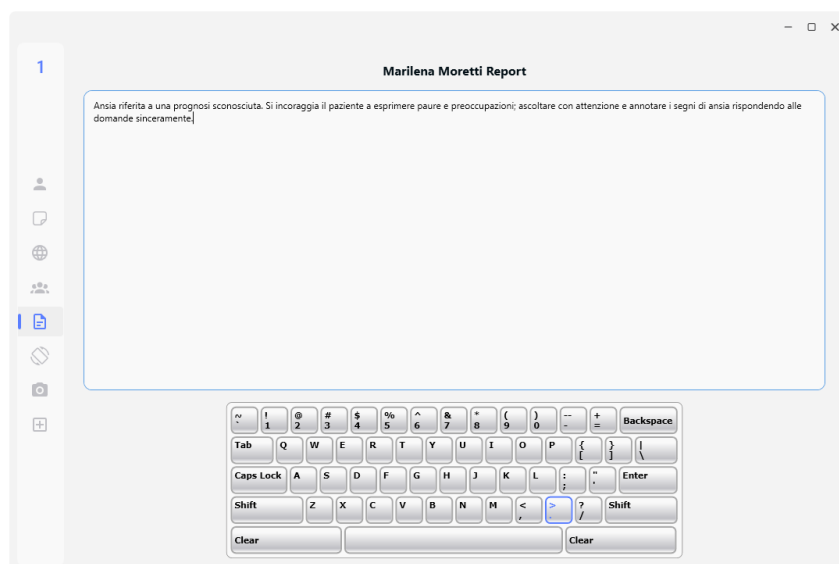


Figura 7.6: Pagina in cui vengono descritte tutte le scelte e decisioni sulla cura del paziente.

La parte che concerne la scrittura del verbale d'equipe è presente nella *Report Page*, fig. 7.6. Qui è possibile scrivere tramite una tastiera collegata allo schermo o direttamente attraverso la tastiera virtuale.

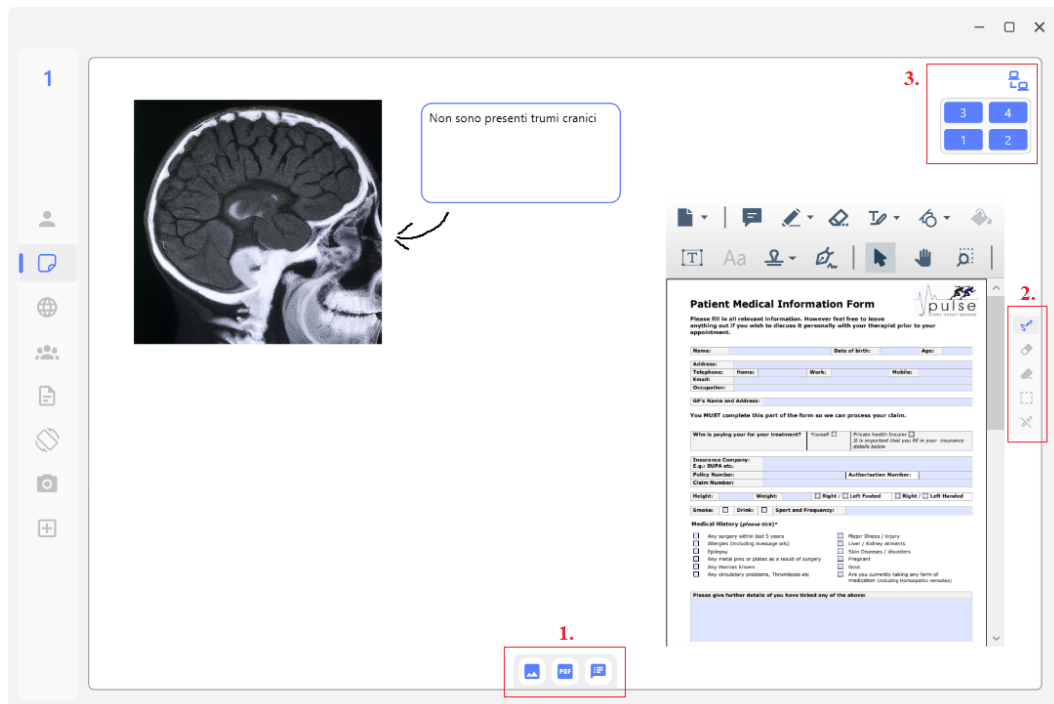


Figura 7.7: Pagina in cui si raggruppano e manipolano le immagini e i documenti necessari a prendere decisioni sulla cura del paziente.

In fig. 7.7 è raffigurata la *Module Page* dalla quale è possibile manipolare i moduli, dei contenitori di elementi (immagini, documenti e blocchi note). Nella pagina, escluso l'utilizzo del touch all'interno della parte bianca, è possibile:

1. inserire immagini, documenti .pdf o .docx attraverso una ricerca all'interno del dispositivo e aggiungere un blocco note;
2. modificare il tipo di input che avviene quando si passano le dita sulla tela bianca:
  - 2.1 *inchiostro*: permette la scrittura o disegno tramite il touch (come la freccia in fig. 7.7);
  - 2.2 *gomma*: cancella su un'area pari alla gomma, i segni effettuati in precedenza;
  - 2.3 *gomma piena*: cancella un tratto completo;

- 2.4 *selezione*: tramite il touch è possibile selezionare uno o più tratti a poi spostarli, ingrandirli e rimpicciolirli;
  - 2.5 *no stilo*: disabilita disegno e scrittura;
3. scegliere con quale postazione dell'altro schermo condividere i propri contenuti tramite il pannello in alto a destra.

La condivisione dei dati di una postazione, permette a due *Module Page* di schermi differenti, di condividere i propri contenuti in tempo reale e consente a entrambe di manipolare i dati. È perciò possibile, ad esempio, da un lato muovere la radiografia e contemporaneamente aggiornare il blocco note nell'altro schermo. I numeri all'interno del pannello indicano il numero delle postazioni mentre il colore, lo stato delle condivisioni dell'altro schermo. Il numero di una postazione è sempre presente in alto a sinistra, i pulsanti invece possono essere di tre colori: blu, è possibile connettersi, nero, tale postazione è già in condivisione con un'altra e verde quando si è in stato di condivisione. Dal pannello in fig. 7.7 si può notare come siano disponibili tutte e quattro le postazioni dell'altro schermo.

Ogni immagine e documento presenti nella pagina del paziente possono essere aperti all'interno di questa pagina, questo caso viene riportato in fig. 7.7 che mostra una radiografia del cranio e un documento che presenta informazioni mediche sul paziente. Trascinando il proprio dito sullo schermo, uno specialista ha disegnato una freccia per indicare a cosa il blocco note facesse riferimento.

Come da specifiche, tutti gli elementi all'interno della *Module Page*, detti moduli, sono manipolabili tramite il touch. Allo stato corrente del prototipo, ogni modulo presenta due stati:

1. **stato iniziale**: non manipolabile tramite il touch, visualizza a schermo l'elemento e permette di usare le sue funzionalità (ad esempio scrivere dentro un blocco note);
2. **stato secondario**: tramite il touch è possibile:
  - 2.1 traslare il modulo;
  - 2.2 fare zoom-in e zoom-out;
  - 2.3 ruotarlo attorno al suo centro.

Si può passare da uno stato all'altro attraverso un *double finger tap*, ovvero un tap con due dita, a seguito di tale azione, per indicare il passaggio di stato, il bordo del modulo si colora di verde e, se eseguito nuovamente, tornerà trasparente. In fig. 7.8 viene mostrato il passaggio di stato e la rotazione come esempio di manipolazione.

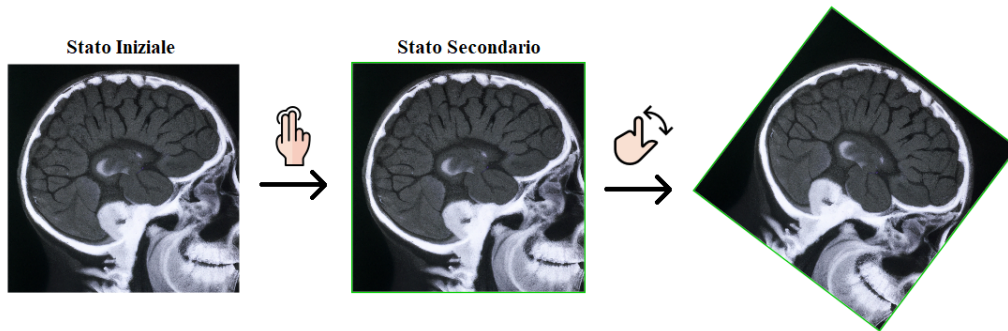


Figura 7.8: Esempio di manipolazione tramite il touch di un Image Module

Attribuzioni icone: freepik e kliwir art - Flaticon

In fig. 7.9 è presente uno stadio più avanzato della sessione rispetto alla precedente interfaccia. In questo caso lo specialista sta continuando ad annotare il blocco note attraverso la tastiera virtuale presente in ogni postazione, che si attiva ogni qualvolta un'area di testo ottiene il focus. Il pannello di controllo è invece illuminato di verde, ciò sta ad indicare che tale postazione è in condivisione con la postazione numero 1 dell'altro schermo e che tutte le altre non sono collegabili. Si può interrompere la condivisione in qualsiasi momento premendo sulla postazione del pannello a cui si è attualmente collegati.

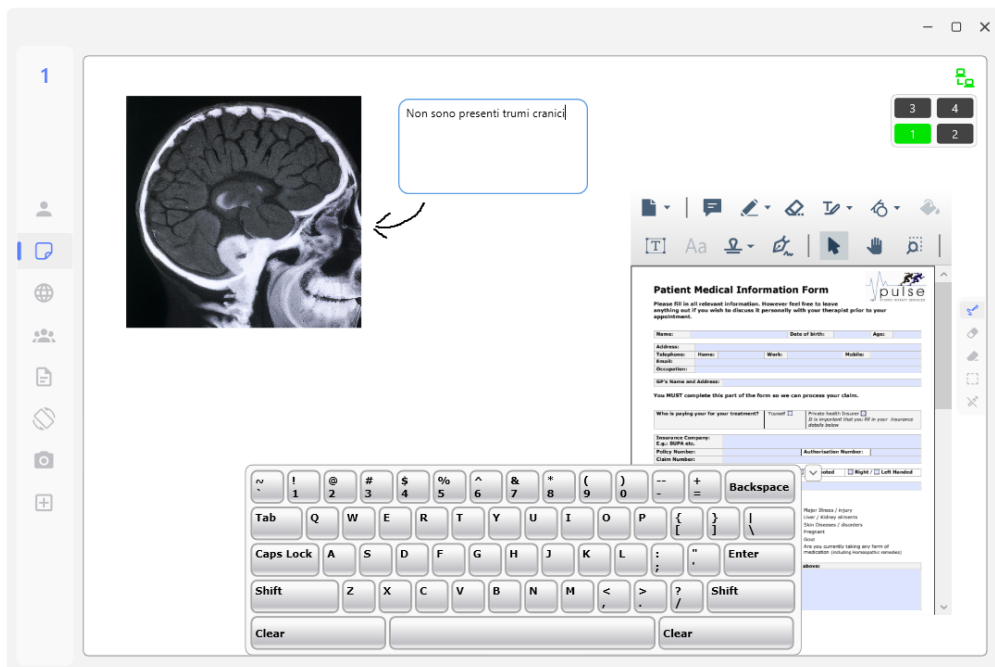


Figura 7.9: Module Page che sta effettuando una condivisione dei propri contenuti con l'altro schermo e presenta la tastiera virtuale aperta.

In ogni schermo cliccando sull'ultimo pulsante del menù laterale, è possibile aumentare il numero di postazioni fino a un massimo di quattro. Tutte le postazioni sono identificabili attraverso il numero in alto a sinistra e presentano le stesse funzionalità. Perciò, ogni specialista può utilizzare qualunque postazione e svolgere i propri compiti in contemporanea con gli altri.

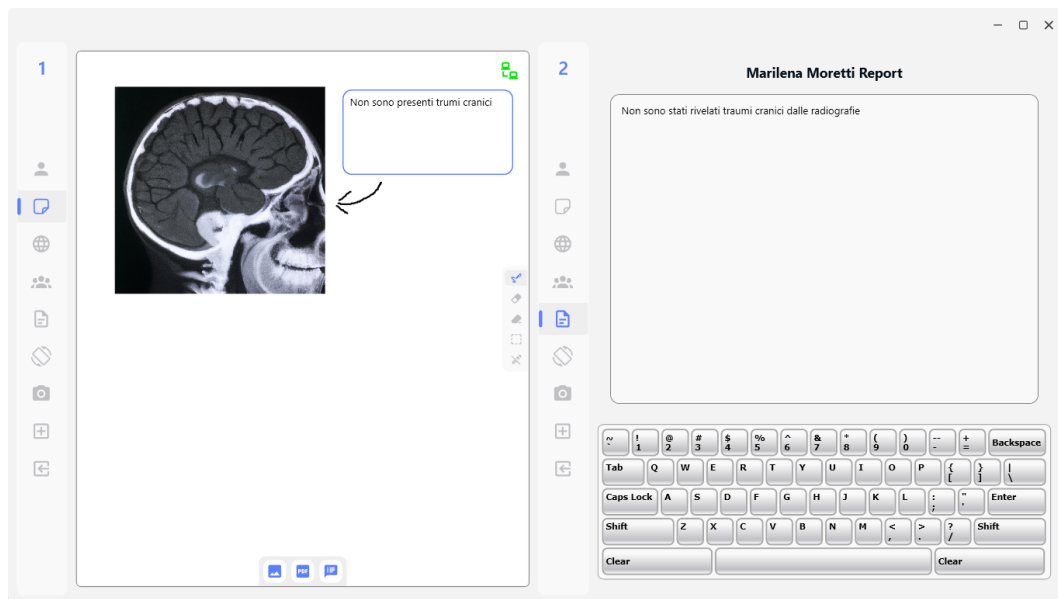


Figura 7.10: Esempio di utilizzo contemporaneo di due postazioni di lavoro nello stesso schermo.

La fig. 7.10 mostra l'utilizzo della *Module Page* e della *Report Page* nello stesso schermo da parte di due specialisti. Se nello stesso dispositivo si utilizzassero contemporaneamente due o più *Module Page*, i vari moduli possono essere spostati tra una pagina e l'altra semplicemente trascinandoli tramite il touch.

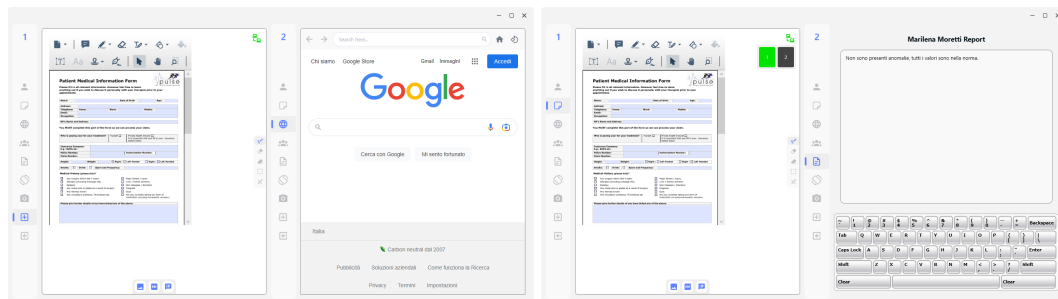


Figura 7.11: Esempio di utilizzo contemporaneo di due postazioni di lavoro in entrambi gli schermi.

Questa gestione delle postazioni porta all'utilizzo di due schermi in contemporanea per eseguire molteplici attività in modo concorrente ed efficace. Nell'esempio in fig. 7.11 è mostrata la condivisione della prima postazione in entrambi gli applicativi. Questa modalità può essere usata da due specialisti, ma anche come semplice visualizzazione sullo schermo a parete.

The screenshot displays a medical report for 'MARILENA MORETTI' dated '02/03/2023'. The report text states: 'Ansia riferita a una prognosi sconosciuta. Si incoraggia il paziente a esprimere paure e preoccupazioni; ascoltare con attenzione e annotare i segni di ansia rispondendo alle domande sinceramente.' Below this, the 'TEAM MEMBERS' are listed as Luca Bazzocchi, Mirko Viroli, Gianluca Aguzzi, and Angelo Croatti. An 'OPENED FILES' section lists three files: 'medical\_form.pdf' (00:05:02), 'head\_xray.png' (00:05:13), and 'chest\_xray.png' (00:05:31). The session time is noted as 00:06:02.

MARILENA MORETTI REPORT		DATE 02/03/2023
Ansia riferita a una prognosi sconosciuta. Si incoraggia il paziente a esprimere paure e preoccupazioni; ascoltare con attenzione e annotare i segni di ansia rispondendo alle domande sinceramente.		
TEAM MEMBERS		
- Luca Bazzocchi - Mirko Viroli - Gianluca Aguzzi - Angelo Croatti		
OPENED FILES		
medical_form.pdf		At time: 00:05:02
head_xray.png		At time: 00:05:13
chest_xray.png		At time: 00:05:31
Session time: 00:06:02		

Figura 7.12: Esempio di verbale prodotto alla fine della sessione con un paziente.

Infine, terminando la sessione dalla pagina del paziente, la sessione viene salvata e viene prodotto il verbale finale sulle modalità di cura del paziente, fig. 7.12. Successivamente lo schermo principale torna alla Department Page da cui è possibile selezionare il prossimo paziente, mentre l'altro rimane in stato di attesa.



## Conclusioni e sviluppi futuri

In questa tesi è stato progettato e implementato un prototipo per migliorare i *Daily Medical Team Briefings* portando alla stesura di un verbale d'equipe e una scelta per la cura dei pazienti. A seguito di un periodo di formazione sull'uso dell'applicativo e delle sue tecnologie, i medici saranno in grado di visualizzare e manipolare i dati in modo più intuitivo, efficace ed efficiente rispetto ai normali ambienti dotati solamente di un pc e un proiettore. Grazie a un totale massimo di otto postazioni, gli specialisti sono in grado di impiegare al meglio la loro conoscenza ed esperienza in maniera diretta senza dover aspettare, uno alla volta, di ricevere il materiale dagli altri.

Nonostante non ci siano state interazioni dirette con gli specialisti, i requisiti e i feedback ricevuti durante l'intero processo di sviluppo e release, è stato frutto di un'ampia interlocuzione con il personale e i clinici dell'AUSL della Romagna.

Al fine di rendere i moduli altamente riusabili ed espandibili, l'architettura e l'implementazione di tali componenti è stata la parte che ha richiesto il maggior carico di lavoro. Per quanto riguarda invece le pagine visualizzabili, la parte più complessa è stata la *Module Page*, in quanto fulcro delle interazioni tra i membri del team. Tutta la parte di gestione della condivisione dei dati tra le postazioni e il salvataggio di essi, ha richiesto uno sforzo non banale.

Il risultato finale di questa tesi, essendo la creazione di un prototipo che dimostri come sia possibile mettere in atto un'evoluzione delle tradizionali riunioni di equipe mediche, tutti i dati riguardanti documenti, immagini, radiografie e persone sono memorizzate all'interno dell'applicazione stessa. In futuro si prospetta un collegamento diretto con i servizi medici per il reperimento di tutti i file necessari all'avvio dei vari meeting tra gli specialisti.

Il progetto, la cui mole di lavoro è già molto ampia, non tiene conto di quali specialisti possano accedere ai dati personali di un paziente ma se sia corretto che tali dati possano essere visibili a tutto il team. Queste valutazioni di carattere sia medico che giudiziario, dovranno essere studiate più in dettaglio, comprendendo anche la possibilità che determinanti dati o postazioni debbano essere accessibili solo a specifici medici.

Durante le fasi iniziali in cui si è svolto un approfondimento del caso di studio e le tecnologie impiegabili, è stata svolta anche un'analisi approfondita di tutte le possibili componenti e funzionalità implementabili nel prototipo. Dall'elenco ottenuto, sulla base della loro priorità, difficoltà di implementazione e tempo disponibile, solo una sottoparte di esse è stata scelta e presentata nelle specifiche di questa tesi. Negli sviluppi futuri del progetto, una delle funzionalità più importanti è quella di annotare e disegnare direttamente sulle immagini tramite il touch e non solo affianco a esse. Questo sarebbe utile per far capire meglio le aree di interesse, ad esempio cerchiandole. Dato l'alto numero di pazienti ricoverati quotidianamente, l'efficienza dei processi è molto importante e il dover scrivere manualmente delle note potrebbe risultare un lavoro talvolta anche molto lento; quindi, l'aggiunta di un modulo/componente che registri la voce per riascoltarla in un secondo momento o che addirittura la trasponga in testo scritto, renderebbe molto più rapida l'intera sessione e le varie interazioni.

L'applicazione potrebbe essere inoltre integrata con altri linguaggi come Python, attraverso un supporto di sistemi e modelli di intelligenza artificiale si potrebbe aggiungere un assistente vocale che aiuti a reperire il materiale e le informazioni associate ai pazienti. Implementando tali funzionalità, potrebbero essere impartiti comandi come "Aprimi la cartella clinica di Mario Rossi sull'intervento per la correzione dei piedi piatti" o anche "Fammi sapere la data dell'ultima vaccinazione contro il tetano". Se invece si desidera fare ascoltare a tutti i presenti un determinato testo, si potrebbe aggiungere un componente che permetta di leggere i documenti con la voce del computer. Questo componente potrebbe essere poi espanso per ascoltare solo parti selezionate del documento o associarlo all'assistente vocale, il quale potrebbe aprire un documento richiesto e leggerlo.

Si potrebbe inoltre aggiungere un modulo che permetta la visualizzazione di modelli DICOM 3d con una integrazione tramite HoloLens. Per quanto riguarda invece la parte di condivisione, essa potrebbe essere espansa alla condivisione non solo nella rete locale ma anche in remoto. In questo modo si potrebbe condividere la voce, la singola postazione o lo schermo intero con uno specialista in remoto la cui partecipazione e feedback potrebbero essere essenziali per un più completo proseguimento della sessione.

In futuro si prospetta anche un'espansione di tutte le funzionalità e azioni applicabili ai moduli, ma anche un possibile differente utilizzo di quelle già implementate, come lo spostamento di un modulo da una postazione all'altra non solo tramite una traslazione, ma anche attraverso uno swipe/flick verso di essa.

Aspetto importante nel valutare la presente tesi è che, nel suo sviluppo, non si è voluto dimostrare l'importanza dell'ICT in ambito medico né era

volontà di trasformare un team di specialisti nel "team perfetto". L'obiettivo era, piuttosto, quello di creare un prototipo che mettesse a disposizione di una qualunque equipe medica, uno strumento che andasse a migliorare l'efficacia e l'efficienza dei briefing medici, permettendo ai professionisti che vi partecipano, di giungere a conclusioni in modo più veloce che, a loro volta, consentono di prendere decisioni più rapide e complete sulle metodologie di cure dei pazienti. Si presuppone comunque che le persone all'interno del team presentino rapporti di fiducia gli uni con gli altri e che siano già ampiamente adattabili e specializzati nelle varie aree di interesse, conoscendo o individuando rapidamente quali documenti possano essere necessari per il meeting.



# Ringraziamenti

Con questa tesi si concludono i miei studi ed il mio lungo percorso universitario in Ingegneria e Scienze Informatiche. Questi cinque anni sono stati molto importanti per me, ci sono stati momenti di gioia ma anche di tristezza, momenti di divertimento e di frustrazione, questi anni mi hanno formato, mi hanno maturato e fatto diventare la persona che sono ora. L'esperienza in questi anni universitari è stata molto bella e appagante; ho superato al primo colpo alcuni esami che sembravano impossibili mentre altri molto più semplici ho dovuto tentarli più volte, c'è anche stato un periodo molto buio a causa della propagazione del COVID-19 ma anche uno fatto da tante nuove esperienze in Erasmus. L'aver potuto passare un intero semestre in un paese come la Turchia ed in particolare una metropoli come Istanbul, con mode e usi completamente diversi da quelli italiani, dove ho potuto conoscere tante persone, di nazionalità e pensieri differenti, parlare sempre e solamente in inglese, provare cose nuove come andare in barca a vela, ballare, fare parapendio, visitare tanti posti nuovi e vivere in un campus universitario insieme a tanti altri studenti, ha cambiato il mio modo di vedere il mondo.

Un grande ringraziamento va alla mia famiglia che mi ha sempre supportato nel mio percorso formativo con gioia e senza mettermi pressioni. Quando mi è capitato di portare a casa anche solamente un 19 sono sempre stati felici per me e sapevano che nell'esame successivo avrei rialzato la media, mentre quando arrivava un 30 o 30 con lode si festeggiava. Mi ha sempre fatto piacere la grande comprensione nei miei confronti e questo mi ha aiutato a superare tutte le difficoltà con calma e secondo i tempi che mi sono imposto tramite il mio metodo di studio.

Ringrazio anche i miei amici e le altre persone che mi sono state accanto in questi anni e nell'ultimo periodo, ci siamo divertiti tanto in questi anni e ne abbiamo tanti altri da passare insieme. Alcuni amici sono fisicamente distanti, ma anche il solo sentirci di tanto in tanto per telefono o di persona in qualche stravagante viaggio ai confini del mondo, mi rende veramente contento.

Questo lavoro poi non sarebbe stato possibile senza l'aiuto e supporto del mio professore e relatore Mirko Viroli e del co-relatore Gianluca Aguzzi, che sono sempre stati disponibili nel rispondere ai miei dubbi ma sono anche stati

presenti quando mi servivano chiarimenti e aiuti, comprese le varie prove di utilizzo del prototipo. Il mio ultimo ringraziamento va infine a Angelo Croatti, colui che ha saputo indirizzare e far convergere la collaborazione e l'utilizzo di schermi multi-touch in ambito medico per la cura dei pazienti. Avendo avuto un riscontro diretto con i clinici, si è stati in grado di valutare in maniera adeguata le specifiche richieste per il prototipo e le migliorie da apportare ad ogni release.

E con questo vi lascio, spero a tutti di risentirvi e rivedervi in futuro; a partire da questo momento spero di riuscire a compiere grandi cose e che voi possiate essere ancora al mio fianco.

# Bibliografia

- [1] Paul W Mattessich and Barbara R Monsey. *Collaboration: what makes it work. A review of research literature on factors influencing successful collaboration*. ERIC, 1992.
- [2] W Keith Edwards. Policies and roles in collaborative applications. In *Proceedings of the 1996 ACM conference on Computer supported cooperative work*, pages 11–20, 1996.
- [3] Prasun Dewan. Architectures for collaborative applications. *Computer Supported Co-operative Work*, 7:169–193, 1999.
- [4] Joan Prades, Eline Remue, Elke Van Hoof, and Josep M Borrás. Is it worth reorganising cancer services on the basis of multidisciplinary teams (mdts)? a systematic review of the objectives and organisation of mdts and their impact on patient outcomes. *Health Policy*, 119(4):464–474, 2015.
- [5] Evert Schot, Lars Tummers, and Mirko Noordegraaf. Working on working together. a systematic review on how healthcare professionals contribute to interprofessional collaboration. *Journal of interprofessional care*, 34(3):332–342, 2020.
- [6] Scott Reeves, Ferruccio Pelone, Reema Harrison, Joanne Goldman, and Merrick Zwarenstein. Interprofessional collaboration to improve professional practice and healthcare outcomes. *Cochrane database of systematic reviews*, (6), 2017.
- [7] Joan Prades, Cristina Coll-Ortega, Lissandra Dal Lago, Karolien Goffin, Eugen Javor, Claudio Lombardo, Johan de Munter, Jordi Ponce, Daniele Regge, Ramón Salazar, et al. Use of information and communication technologies (icts) in cancer multidisciplinary team meetings: an explorative study based on eu healthcare professionals. *BMJ open*, 12(10):e051181, 2022.

- 
- [8] Myles Leslie and Elise Paradis. Is health information technology improving interprofessional care team communications? an ethnographic study in critical care. *Journal of Interprofessional Education & Practice*, 10:1–5, 2018.
- [9] Jos Aarts, Hans Doorewaard, and Marc Berg. Understanding implementation: the case of a computerized physician order entry system in a large dutch university medical center. *Journal of the American Medical Informatics Association*, 11(3):207–216, 2004.
- [10] Catharina Widmark, Christer Sandahl, Katarina Piuva, and David Bergman. Barriers to collaboration between health care, social services and schools. *International journal of integrated care*, 11(3), 2011.
- [11] Nicholas Goodwin. How important is information and communication technology in enabling interprofessional collaboration?, 2017.
- [12] Anna Janssen, Tracy Robinson, Melissa Brunner, Paul Harnett, Kylie E Museth, and Tim Shaw. Multidisciplinary teams and ict: a qualitative study exploring the use of technology and its impact on multidisciplinary team meetings. *BMC health services research*, 18:1–10, 2018.
- [13] Nick Goodwin and Albert Alonso. Understanding integrated care: the role of information and communication technology. In *Achieving effective integrated e-care beyond the silos*, pages 63–88. IGI Global, 2014.
- [14] Ryuichi Ohta and Yoshinori Ryu. Improvement in palliative care quality in rural nursing homes through information and communication technology-driven interprofessional collaboration. *Rural and Remote Health*, 21(2), 2021.
- [15] Yan Xiao. Artifacts and collaborative work in healthcare: methodological, theoretical, and technological implications of the tangible. *Journal of biomedical informatics*, 38(1):26–33, 2005.
- [16] M Krithikaa. Touch screen technology—a review. *International Journal of Trend in Research and Development (IJTRD)*, 3(1):2394–9333, 2016.
- [17] Mrudula Nimbarte. Multi-touch screen interfaces and gesture analysis: a study. *Advanced Computing*, 2(6):113, 2011.
- [18] Magdalyn Patyk, Sandra Gaynor, James Kelly, and Vivian Ott. Touch-screen computerized education for patients with brain injuries. *Rehabilitation Nursing*, 23(2):84–87, 1998.



- 
- [19] Ronald S Weinstein, Richard A McNeely, Michael J Holcomb, Leela Doppalapudi, Michael J Sotelo, Ana Maria Lopez, Kristine A Erps, Christopher J Martin, Elizabeth A Krupinski, Anna R Graham, et al. Technologies for interprofessional education: The interprofessional education-distributed. *Journal of Allied Health*, 39(3):238–245, 2010.
- [20] Anne Marie Piper and James D Hollan. Supporting medical communication for older patients with a shared touch-screen computer. *International journal of medical informatics*, 82(11):e242–e250, 2013.
- [21] David Nicholas, Paul Huntington, Peter Williams, and Pam Vickery. Health information: an evaluation of the use of touch screen kiosks in two hospitals. *Health Information & Libraries Journal*, 18(4):213–219, 2001.
- [22] Adam Nathan. *Windows presentation foundation unleashed*. Pearson Education, 2006.
- [23] Ashish Ghoda and Mamta Dalal. *XAML developer reference*. Pearson Education, 2011.
- [24] Gary R Wright and W Richard Stevens. *TCP/IP Illustrated, Volume 2 (paperback): The Implementation*. Addison-Wesley Professional, 1995.
- [25] Troy Willmot. Really simple service discovery protocol for .net, 2020.
- [26] Yaron Y Goland, Ting Cai, Paul Leach, Ye Gu, and Shivaun Albright. Simple service discovery protocol. *IETF Draft*, 1(3), 1999.