



UNIVERSITÀ DEGLI STUDI DI PADOVA

FACOLTÀ DI SCIENZE MATEMATICHE, FISICHE E NATURALI

*Corso di Laurea Triennale in Informatica*

COMUNICAZIONE AUMENTATA/ALTERNATIVA PER  
GIOVANI OSPITI DELLA TERAPIA INTENSIVA  
PEDIATRICA

*Tesi di Laurea Triennale in Informatica*

RELATORE:  
PROF. PALAZZI CLAUDIO E.

PRESENTATA DA:  
BARON FEDERICO

ANNO ACCADEMICO 2010/11



# Indice

<b>1</b>	<b>Introduzione</b>	<b>1</b>
1.1	La Comunicazione Aumentativa/Alternativa (CAA) . . . . .	1
1.2	La Terapia Intensiva Pediatrica (TIPed) . . . . .	3
1.3	L'informatica Pediatrica . . . . .	3
1.4	Il progetto VOCA.TIO . . . . .	5
1.5	Mind Reader Squared . . . . .	6
<b>2</b>	<b>Analisi dei requisiti</b>	<b>9</b>
2.1	Descrizione generale . . . . .	9
2.1.1	Contesto d'uso del prodotto . . . . .	9
2.1.2	Caratteristiche degli utenti . . . . .	11
2.2	Casi d'uso . . . . .	14
2.3	Lista dei requisiti stage . . . . .	18
2.3.1	Requisiti funzionali (RF) . . . . .	18
2.3.2	Requisiti di qualità (RQ) . . . . .	19
2.3.3	Requisiti di vincolo (RV) . . . . .	20
2.4	Lista dei requisiti progetto VOCA.TIO . . . . .	20
2.4.1	Requisiti funzionali (RF) . . . . .	20
<b>3</b>	<b>Analisi tecnologica</b>	<b>23</b>
3.1	Valutazione tecnologie, strumenti e linguaggi utilizzabili . . . . .	23
3.2	Studio delle opzioni tecnologiche per l'input . . . . .	25
3.2.1	Eyetracking . . . . .	25
3.2.2	Brain-Computer Interface (BCI) . . . . .	26
<b>4</b>	<b>Progettazione e sviluppo di Mind Reader Squared</b>	<b>29</b>
4.1	Selezione dei concetti e delle immagini . . . . .	29
4.2	Decomposizione architetturale . . . . .	33

4.3 Database e logica di persistenza . . . . .	34
4.4 Activities . . . . .	38
4.4.1 Modalità 1 . . . . .	39
4.4.2 Configurazione simboli . . . . .	41
4.4.3 Configurazione utente . . . . .	46
4.4.4 Login . . . . .	48
4.4.5 Menu . . . . .	49
<b>5 Conclusioni</b>	<b>51</b>
5.1 Obiettivi raggiunti . . . . .	51
5.2 Sviluppi futuri . . . . .	52
5.3 Considerazioni personali . . . . .	53
<b>6 Glossario</b>	<b>55</b>
<b>Riferimenti Bibliografici</b>	<b>61</b>
<b>A Appendice - VOCA.TIO: resconto del test di venerdì 2/9</b>	<b>63</b>

# Elenco delle figure

1.1	Una tavola per la comunicazione alternativa low-tech che raffigura varie tipologie di alimenti. . . . .	2
1.2	Un dispositivo VOCA per la comunicazione attraverso simboli. . . . .	2
1.3	Slide della presentazione “Informatica pediatrica per interventi riabilitativi precoci” del 22/09/2011 di Roberto Mancin che spiega le principali differenze tra l’informatica generale e quella pediatrica. . . . .	5
2.1	Diagramma UML dei casi d’uso che descrive le funzionalità della Modalità 1 di MRS . . . . .	14
2.2	Diagramma UML dei casi d’uso che descrive le funzionalità di configurazione di MRS . . . . .	16
2.3	Diagramma UML dei casi d’uso che descrive la procedura di modifica di un simbolo in MRS. . . . .	17
3.1	Parte dell’occhio su cui si basano gli algoritmi di openEyes. . . . .	26
3.2	Cuffia per BCI Emotiv EPOC. . . . .	27
4.1	Diagramma UML delle classi che descrive l’architettura ad alto livello di Mind Reader Squared. . . . .	33
4.2	Schema che descrive il ciclo di vista di un activity nel framework Android. . . . .	38
4.3	Screenshot dell’applicazione mentre è in uso la modalita 1. . . . .	40
4.4	Interfaccia grafica di TabsFragment (screenshot). . . . .	43
4.5	Interfaccia grafica di OptionsFragment mentre viene scattata una foto (screenshot). . . . .	44
4.6	Interfaccia grafica di SelectImageFragment (screenshot). . . . .	45
4.7	Interfaccia grafica di SummaryFragment (screenshot). . . . .	45
4.8	Interfaccia grafica di ManageSymbolActivity mentre l’utente sta digitando un testo per modificare un simbolo (screenshot). . . . .	46

4.9	Interfaccia grafica di ManageProfileActivity di un utente che utilizza il russo come lingua preferita (screenshot). . . . .	47
4.10	Interfaccia grafica di ManageProfileActivity di un utente che utilizza l'italiano come lingua preferita (screenshot). . . . .	47
4.11	Interfaccia grafica di LoginActivity, form di login (screenshot). . . . .	48
4.12	Interfaccia grafica di LoginActivity, form di registrazione (screenshot). . . . .	48
4.13	Interfaccia grafica di MenuActivity (screenshot). . . . .	49
6.1	Immagine che descrive lo schema di interazioni tra le componenti del pattern architetturale MVC. . . . .	57

# Elenco delle tabelle

4.1	Tabella dei concetti e delle immagini selezionate per la modalità 1 di MRS. . . . .	33
4.2	Tabella degli attributi di ProfileDAO e SymbolDAO. . . . .	35
4.3	Descrizione dei metodi più importanti di ProfileLogic . . . . .	36
4.4	Descrizione dei metodi più importanti di SymbolLogic . . . . .	37





---

# 1 Introduzione

## 1.1 La Comunicazione Aumentativa/Alternativa (CAA)

La CAA riguarda forme di comunicazione capaci di sostituire, integrare, aumentare il linguaggio verbale orale. Il termine Aumentativa si riferisce alla capacità di incrementare le doti comunicative naturali della persona e Alternativa in quanto utilizza strategie comunicative diverse da quelle tradizionali.

La CAA si pone come obiettivo la compensazione di una menomazione (temporanea o permanente) delle strutture corporee necessarie per comunicare; vengono infatti create le condizioni affinché la persona con abilità alterate abbia l'opportunità di comunicare in modo efficace, ovvero di tradurre il proprio pensiero in una serie di azioni intelligibili automaticamente.

Il moderno utilizzo della CAA ha inizio intorno agli anni '50 con l'utilizzo sempre più massiccio del linguaggio dei segni e dei sistemi di comunicazione che si basano su simboli (**LIS** e **Bliss**) che divennero una vera e propria alternativa al linguaggio parlato per la comunicazione interpersonale. Con il rapido aumento del livello tecnologico, tra cui lo sviluppo di microcomputers e sintesi vocale, è stato presto possibile progettare dispositivi in grado di adattarsi alle esigenze delle persone e di produrre un output vocale.

Numerosi sono attualmente gli ausili e i metodi, disponibili sul mercato od oggetto di ricerca, sviluppati ed utilizzati per facilitare alle persone disabili la comunicazione, anche perché sono diversissime le abilità residue e le necessità di chi ne ha bisogno. Di seguito vengono presentate le caratteristiche dei principali sistemi di CAA che utilizzano dispositivi, elettronici e non, per la trasmissione e la ricezione dei messaggi.

**Low-tech:** Gli ausili per la comunicazione low-tech sono quelli che non necessitano

di batterie, elettricità o nulla di elettronico. Spesso si tratta di semplici tavole o libri in cui l'utente può selezionare lettere, parole, frasi, immagini e simboli per comporre i messaggi. Le possibilità per selezionare il messaggio, in base alle abilità dell'individuo, sono svariate, come ad esempio la direzione dello sguardo o l'utilizzo delle mani.

food 	hamburger 	Pizza 	Burrito 	Spaghetti 
bread 	cheese 	French fries 	hot dog 	popcorn 
fruit 	banana 	ice cream 	cookie 	candy 
drink 	water 	juice 	soft drink 	coffee 

*Figura 1.1: Una tavola per la comunicazione alternativa low-tech che raffigura varie tipologie di alimenti.*

**High-tech:** I sistemi high-tech possono essere sia dispositivi dedicati che componenti software di un computer. La loro caratteristica è quella di poter riprodurre la voce tramite registrazioni o sintesi vocale che offrono un potere comunicativo eccellente. Molti di questi ausili richiedono una configurazione particolarmente precisa e la necessità di un discreto allenamento, tanto da richiedere l'utilizzo parallelo delle tavole simboliche. Tra questi dispositivi ricordiamo i VOCA (voice output communication aids), sicuramente tra i più celebri ed utilizzati. In genere sono del tutto indipendenti dal Personal Computer, presentano una tastiera più o meno complessa sulla quale ogni tasto rappresenta un simbolo ed alla pressione rilascia il messaggio vocale corrispondente.



*Figura 1.2: Un dispositivo VOCA per la comunicazione attraverso simboli.*

## 1.2 La Terapia Intensiva Pediatrica (TIPed)

Un caso di particolare interesse per l'utilizzo della CAA è rappresentato dai bambini ricoverati presso la Terapia Intensiva Pediatrica (TIPed) di Padova. Questa struttura ospita frequentemente bambini che per patologie temporanee/permanenti, per età, o per lingua hanno difficoltà durante la degenza a comunicare verbalmente o graficamente in modo efficace ed efficiente col personale medico e paramedico.

La TIPed fa parte del Dipartimento di Pediatria Salus Pueri dell'Università di Padova, di conseguenza il personale della terapia intensiva collabora quotidianamente con i professori ed i ricercatori dell'Ateneo. Questo fa sì che l'ambiente della terapia intensiva di Padova sia unico nel suo genere, supportando l'attività clinica e le cure specialistiche con attività di ricerca ed apertura verso l'innovazione.

Il reparto è di dimensioni contenute ma costituisce l'eccellenza nel settore: ad esso vengono infatti trasferiti pazienti da diversi altri ospedali regionali ed extra-regionali. Ecco alcune note sulle attività prevalenti prese dal sito del dipartimento

<http://www.pediatria.unipd.it/>:

*Il reparto di Terapia Intensiva Pediatrica, comprensivo di 10 posti letto completamente attrezzati per il monitoraggio estensivo dei parametri vitali e per la ventilazione meccanica, si occupa dell'assistenza di pazienti con età compresa tra 0 e 18 anni con compromissione delle funzioni vitali o a rischio imminente di compromissione delle stesse.[...] Il reparto di Terapia Intensiva Pediatrica è stato istituito nel 1990; gli accessi sono progressivamente aumentati nel corso degli anni.*

## 1.3 L'informatica Pediatrica

Da [www.pedinfo.org](http://www.pedinfo.org): Pediatric Informatics is the study of health information used in the care of infants, children, or adolescents

L'apertura della Pediatria di Padova verso la ricerca di soluzioni innovative di cui si è parlato nel paragrafo precedente ha favorito da molti anni lo sviluppo e l'applicazione di sistemi informatici progettati per un'utenza pediatrica.

L'utenza pediatrica infatti ha peculiarità estremamente diverse e più complesse rispetto all'utenza standard. Non si tratta del bambino con menomazioni che deve adattarsi

allo strumento digitale come solitamente accade ma esattamente l'opposto.

L'informatica pediatrica adotta i paradigmi architettonici dell'**universal design** per creare sistemi digitali usabilmente accessibili e facilmente adattabili. Le abilità motorie, sensoriali e cognitive residue ed emergenti sono in rapida evoluzione in età evolutiva ed in TIPED in particolare. Per questo i sistemi multimediali studiati nell'informatica pediatrica devono essere anche multimodali per l'input e multisensoriali per l'output, in modo da permettere l'uso del canale migliore in ogni momento.

Troppo spesso il bambino intubato e cosciente ospite della terapia intensiva non ha voglia di far emergere la propria curiosità e volontà di comunicare perchè impaurito e disorientato in un ambiente altamente tecnologico con rumori di monitor, ventilatori ed allarmi acustici. Anche i genitori quando vedono i tubi e i fili ai quali il loro bambino è attaccato sono impauriti e silenti. Per questo un sistema digitale non deve essere solamente amichevole (user-friendly) ma anche stimolante, wireless e familiare. Una delle vie per raggiungere questi obiettivi è sicuramente quella di progettare il sistema in modo da assomigliare il più possibile ad un videogioco, con livelli sempre più stimolanti e interattivi.

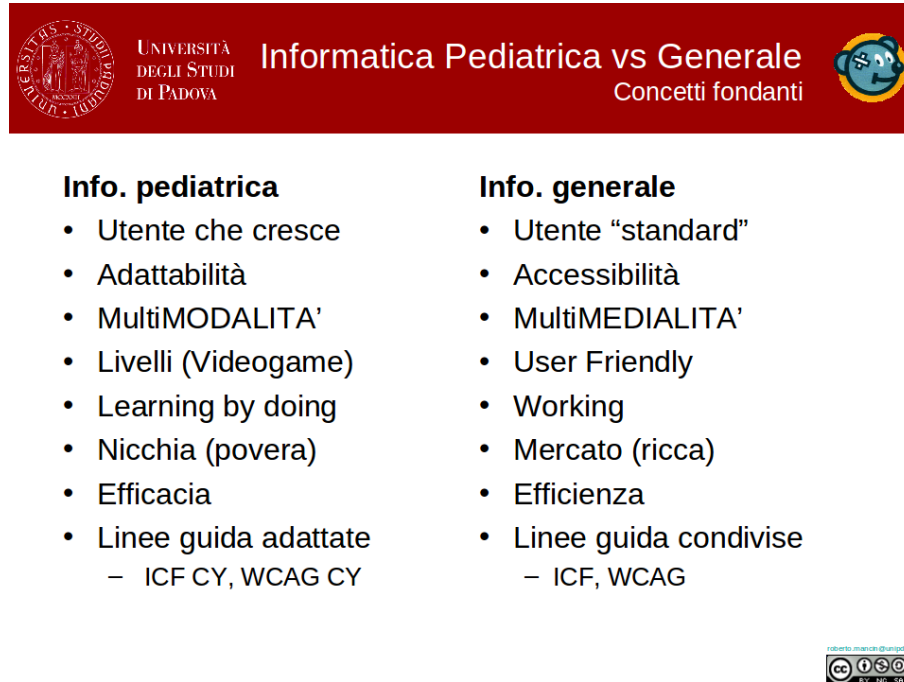
Mentre nell'informatica standard l'utente deve conoscere il sistema prima di usarlo, nell'informatica pediatrica il paradigma learning by doing è fondamentale. Il bambino, infatti, non ha tempo nè voglia di fare un corso e nemmeno di addestrare il sistema. Il gioco deve iniziare subito e solo mentre si gioca sarà possibile imparare cose nuove.

Fortunatamente pochi bambini hanno le necessità estreme che, per pochi giorni o settimane, sperimentano gli ospiti della TIPed; per cui l'informatica pediatrica nella quale si cercano soluzioni veramente efficaci è una branca di nicchia sconosciuta al mercato di massa. È vero però che tutti siamo temporaneamente abili ed invecchiando ci troviamo inevitabilmente ad avere necessità speciali. Per questo, da un pò di anni, anche nell'informatica "standard" si parla di accessibilità universale ai contenuti e si presta attenzione a chi utilizza tecnologie povere o sistemi mobili per accedere al web.

Il limite degli standard ISO, W3C e OMS (Organizzazione Mondiale della Sanità) come le **WCAG** o l'**ICF** è che non sono pensati per una persona che sta crescendo. Ad esempio non ha senso in un e-book di geometria per bambini delle elementari descrivere il disegno di un rombo con il tag alternativo "rombo" se questi non sanno

cosa sia un rombo, o bloccare un forum con un **CAPTCHA** multimediale insuperabile per un bambino.

Segue una tabella che riassume questi concetti.



Info. pediatrica	Info. generale
<ul style="list-style-type: none"> <li>• Utente che cresce</li> <li>• Adattabilità</li> <li>• MultiMODALITA'</li> <li>• Livelli (Videogame)</li> <li>• Learning by doing</li> <li>• Nicchia (povera)</li> <li>• Efficacia</li> <li>• Linee guida adattate               <ul style="list-style-type: none"> <li>– ICF CY, WCAG CY</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>• Utente "standard"</li> <li>• Accessibilità</li> <li>• MultiMEDIALITA'</li> <li>• User Friendly</li> <li>• Working</li> <li>• Mercato (ricca)</li> <li>• Efficienza</li> <li>• Linee guida condivise               <ul style="list-style-type: none"> <li>– ICF, WCAG</li> </ul> </li> </ul>

roberto mancini @unipad.it

*Figura 1.3: Slide della presentazione “Informatica pediatrica per interventi riabilitativi precoci” del 22/09/2011 di Roberto Mancini che spiega le principali differenze tra l’informatica generale e quella pediatrica.*

## 1.4 Il progetto VOCA.TIO

VOCA.TIO è il progetto multidisciplinare nato all’interno del Dipartimento di Pediatria nel quale si colloca il mio lavoro di stage. Dal primo giorno del mio servizio presso la TIPed ho assistito alla progettazione ed alla crescita di questo progetto di ricerca fino alla sua presentazione ufficiale all’Ateneo.

Seguire i primi passi di VOCA.TIO, o meglio “Vocal output communication aids for temporarily impaired owners: a feasibility study”, è stato molto utile ai fini dei processi di analisi e progettazione del software che ho sviluppato durante lo stage, poiché le due cose devono essere in totale sintonia. Infatti la ricerca proposta con VOCA.TIO si basa sull’utilizzo, presso la TIPed di Padova, di uno strumento soft-

ware che permetta la CAA, in particolare un Serious Game.

La relazione tra lo stage e VOCA.TIO è dunque molto stretta, il primo non è altro che un punto di partenza per un progetto multidisciplinare molto più grande.

## 1.5 Mind Reader Squared

*Negli ultimi vent'anni, il diffondersi degli studi sugli effetti della comunicazione e l'esigenza di condividere sempre più con il paziente e con la sua famiglia le decisioni in merito alle cure, hanno reso gli operatori sanitari più sensibili agli aspetti relazionali ed educativi del loro lavoro. [...]*

*Il bambino intubato e cosciente necessita non meno degli altri di comunicare con coloro che lo assistono. [...]*

*Garantire la possibilità di esprimersi anche in questa situazione è un dovere di ogni operatore sanitario che si relazioni con loro.*

Questa citazione, da “La comunicazione in terapia intensiva pediatrica” di Cristina Zaggia, coordinatore infermieristico della TIPed di Padova, evidenzia la necessità, ma soprattutto la volontà del personale della TIPed di porre grande attenzione all'aspetto comunicativo nella cura dei bambini. Questa loro caratteristica ha già portato in passato all'utilizzo di strumenti sperimentali nel reparto.

L'idea che sta alla base del mio stage presso la TIPed nasce dal desiderio di rendere più flessibile ed usabile uno strumento cartaceo prodotto da uno studio svolto per una tesi di laurea in Infermieristica [Caoduro 2009]. Questo strumento consiste in un libretto, facilmente sfogliabile, contenente un'immagine per ogni pagina. Queste immagini sono state accuratamente selezionate per rappresentare tutti i bisogni primari.

Seguendo la volontà dello staff della TIPed, si è pensato di trasferire su tecnologia mobile questo strumento, che sarà la base di numerose ricerche ed applicazioni informatiche per :

- migliorare la qualità di vita di chi si risveglia su un letto della TIPED e quindi favorire un riabilitazione precoce;
- migliorare e stimolare la comunicazione con i genitori e con lo staff medico ed infermieristico;
- migliorare la qualità dell'assistenza.

Questi sono anche gli obiettivi del progetto VOCA.TIO di cui, come si è detto, MRS è la base tecnologica. Al software così ipotizzato è stato dato il nome di Mind Reader Squared ed è stato il mio compito durante lo stage quello di svilupparlo, dallo studio di fattibilità alla validazione. Il nome Mind Reader deriva dai precedenti progetti portati avanti in questo contesto che hanno dato vita a due software:

- **Mind Reader 1:** è un text editor usabile anche da persone con gravi disabilità neuromotorie per scrivere minimizzano il numero di tasti. Questo è possibile grazie ad algoritmi di “word prediction” di tipo statistico e sintattico. L’applicazione è stata sviluppata in Delphi come oggetto di tesi in Informatica [Mancin R. Anno Accademico 1996/1997].
- **Mind Reader 2:** oltre a semplificare la grafica di MR 1 integra la “word prediction” sopra citata con un sistema di headtracking per controllare il mouse (input) e con la sintesi vocale (output). Mind Reader 2 permette di scrivere frasi di media lunghezza solamente con piccoli movimenti del capo [Mancin R. Anno Accademico 2006/2007].

MR 1 e 2, che sono utilizzabili solamente da individui capaci di leggere e scrivere in italiano e dotati di un controllo muscolare lento ma complesso, inoltre è richiesta all’utente una forte volontà di comunicare. Condizioni che difficilmente un ospite della TIPED soddisfa.

MRSquared invece è accessibile anche ai bambini non scolarizzati o di altra madrelingua con abilità motorie e sensoriali temporaneamente assai ridotte.

La parola squared vuole richiamare la definizione che è stata data nel 2009 all’attuale era del web: web squared (web al quadrato). Il tutto nasce dalla convergenza tra i servizi di web 2.0 e i sensori intelligenti posizionati sui dispositivi mobili di nuova generazione.

*Se la prima generazione di applicazioni 2.0 si basava solo su ciò che gli utenti scrivevano sulla tastiera, ora i dispositivi mobili non richiedono nemmeno l’intervento umano. Hanno i loro occhi, orecchie e senso del tatto*

Così Tim O’Reilly, inventore dei termini “web 2.0” e “web squared”, descrive la nuova era prendendo ad esempio quelle applicazioni che vanno sotto il nome di “realtà aumentata”. Mind Reader Squared si posiziona esattamente in questo contesto.





---

## 2 Analisi dei requisiti

### 2.1 Descrizione generale

#### 2.1.1 Contesto d'uso del prodotto

Il progetto nasce da una precisa necessità della Terapia Intensiva Pediatrica di Padova che ospita spesso bambini svegli e coscienti ma con gravi menomazioni temporanee ed importanti necessità comunicative. Il risveglio da uno stato alterato della coscienza (coma, anestesia totale) è un evento molto raro in età pediatrica ma relativamente frequente presso la TIPed. Limitare le paure tipiche di questo processo in cui il bambino è vigile ma non pienamente cosciente aprendo canali di comunicazione che amplificano le abilità residue ed emergenti è uno degli obiettivi principali del progetto VOCA.TIO. Un giovane ospite della TIPed può essere in una condizione di **locked in funzionale**, ovvero quella fase di ogni risveglio in cui nessun muscolo volontario può essere controllato, per altri motivi: ustioni, avvelenamenti, monitoraggio post-operatorio, insufficienze respiratorie, politraumi, sindrome da insufficienza multiorgano.

#### Utilizzo standard a supporto delle cure ospedaliere

Gli utenti del sistema non saranno solamente i bambini ma anche le loro famiglie ed il personale infermieristico.

I genitori potrebbero, ad esempio, desiderare di personalizzare lo strumento con suoni ed immagini esterne alla TIPed. Questo non dovrà interferire con l'attività assistenziale erogata dal personale medico ed infermieristico. È indubbio infatti che in ospedale venga data la priorità alla cura della malattia, alle terapie, alle visite mediche per cui il benessere degli ospiti e la qualità della loro vita vengano in secondo piano. La configurazione dell'applicazione quindi potrà essere fatta in qualsiasi momento e

da ogni adulto. Sono richieste le competenze che ha ogni adulto in grado di usare in modo elementare un telefonino.

L'utilizzo di questo strumento per la CAA comunque non nasce come necessità di un genitore, il quale non ha certo la lucidità necessaria, ma è frutto di un lungo processo promosso da alcuni membri dello staff e sposato dagli altri dato che tutti quotidianamente sperimentano la frustrante situazione di non poter fornire sempre un'adeguata assistenza a causa di banali problemi comunicativi. Con l'aumento di ospiti non madre lingua italiana questo problema diventa ogni anno più grave.

L'introduzione di una tecnologia innovativa in un contesto pediatrico è sempre fallimentare quando la volontà non è condivisa da infermieri, medici e personale tecnico. È possibile sviluppare l'applicazione più accessibile e divertente ma senza un supporto efficace ed efficiente rimarrà chiusa in un cassetto. Fra l'utente finale e lo sviluppatore ci sono molte professionalità che hanno avuto, hanno ed avranno un ruolo fondamentale per curare l'utilizzo e lo sviluppo di MRS e queste sono:

- i genitori che devono dare un consenso informato all'uso dello strumento e, se desiderato, personalizzarlo;
- lo staff del reparto che deve sapere quando è opportuno utilizzarlo e quando è meglio usare sistemi tradizionali;
- il gruppo informatico della pediatria che deve essere capace di risolvere in tempo breve problemi insoliti.

### **Utilizzo successivo alla dimissione del bambino**

Le intubazioni post operatorie, la causa più frequente in TIPED di locked-in funzionale, durano al massimo 48 ore. Successivamente il bambino può tornare a parlare e si può alzare dal letto. Questa evidenza potrebbe far sembrare assurdo parlare di inclusione e riabilitazione precoce. E' vero però che quasi ogni mese, solo a Padova, la TIPed accoglie un bambino in età scolare con gravissimi problemi neuro-motori causati da una grave lesione cerebrale. In questi casi non sono tubi e fili la causa del locked-in funzionale per cui la riabilitazione, quando possibile, successiva alla dimissione ospedaliera, spesso richiede mesi di logopedia e di terapia occupazionale.

Non è eticamente corretto ed economicamente opportuno togliere al bambino lo strumento per la CAA utilizzato durante il ricovero col rischio di interrompere il processo

di riabilitazione e reinserimento scolastico ne' e' possibile presupporre che i genitori abbiano la possibilità economica di acquistare immediatamente un tablet. Per queste ragioni è importante che MRS possa essere facilmente installato su altri dispositivi o dare, in comodato d'uso, al bambino lo strumento il giorno della dimissione.

### 2.1.2 Caratteristiche degli utenti

I giovani ospiti della TIPed hanno età compresa tra gli 0 e i 18 anni e presumibilmente non hanno alcuna esperienza riguardo all'utilizzo di dispositivi tablet o anche di PC (anche se questo, fortunatamente, è sempre meno vero). I bambini sono intubati a causa della ventilazione assistita e sono dunque costretti ad usare MRS in posizione supina o, in alcuni casi, seduta. Vi sono poi da considerare le abilità fisiche e cognitive che possono essere ridotte a causa di patologie temporanee/permanenti e della sedazione.

La mission di un buon informatico pediatrico patavino (ateneo il cui motto è *universa universis patavina libertas*), cioè rendere reale e possibile a tutti (*universis*) un efficace utilizzo di ogni soluzione (*universa*), fa sì che le caratteristiche degli utenti assumano un ruolo chiave, non solo nell'analisi ma durante tutto il progetto. È un'operazione tanto importante quanto difficile che richiede, oltre che una buona familiarità con l'ambiente della TIPed, anche buone conoscenze di medicina e di psicologia.

Durante l'attività di analisi che ho svolto con il prezioso aiuto dello Staff infermieristico medico ed informatico della clinica pediatrica di Padova, ho cercato di evidenziare alcuni fattori critici da seguire, poi, come linee guida nella progettazione di MRS.

**Età:** La differenza di età nei bambini rappresenta cambiamenti sostanziali nelle conoscenze e nelle capacità. In particolare vi è uno scalino netto tra i bambini in età prescolare e quelli invece alfabetizzati. Questo significa che l'applicazione deve consentire dalla CAA tramite semplici immagini che richiamano bisogni primari alla CAA tramite testi complessi con la possibilità, magari, di condividerli via web. Chiaramente devono esistere anche dei livelli intermedi.

**Provenienza:** TIPed a Padova ospita bimbi di ogni nazionalità, chiaramente di madrelingua diversa dall'italiano. Questo, oltre a costituire un'inevitabile difficoltà nella comunicazione tramite messaggi testuali, rende molto più difficile la comunicazione anche attraverso i simboli. È stato riscontrato come trovino difficoltà i bambini a riconoscere simboli che hanno elementi che non siano

tipici della propria cultura o della propria età, come possono essere ad esempio i lineamenti europei per un bambino asiatico o la rappresentazione di concetti astratti come dolore, paura e serenità. Per questo in collaborazione col pronto soccorso pediatrico si sta cercando di creare dei set di immagini adatte all'età e la cultura del bambino. Inoltre il sistema dovrà essere configurabile in ogni sua parte: testi, simboli, immagini e audio.

**Stato d'animo / interesse:** Abbiamo analizzato alcuni progetti di ricerca svolti in ospedali (non pediatrici) degli Stati Uniti che si sono occupati dello studio sull'utilizzo della CAA in una Terapia Intensiva (es. "Electronic voice-output communication aids for temporarily nonspeaking patients in a medical intensive care unit: a feasibility study", University of Pittsburgh). Tutte queste sperimentazioni, sebbene lontane dalle caratteristiche di VOCA.TIO, hanno evidenziato come prima barriera all'utilizzo di dispositivi per la CAA la mancanza di interesse del paziente a comunicare. Questo fattore è emerso anche dalle impressioni del personale medico e infermieristico di Padova, sicuramente molti dei bambini non avranno intenzione di scrivere lunghi testi tramite MRS, ma avranno in primis la necessità di esprimere i bisogni primari. Ecco perché abbiamo deciso di presentare MRS sotto forma di Serious Game; questo strumento deve essere in grado di favorire la comunicazione alternativa tramite un gioco con cui il bambino può confrontarsi autonomamente e di propria spontanea volontà: "protagonista del proprio processo di riabilitazione".

**Abilità cognitive:** Il risveglio parziale da un stato alterato della coscienza (anestesia totale, coma), il deterioramento delle condizioni di salute o più comunemente lo stato di sedazione possono alterare pesantemente le abilità cognitive dei bambini. Questo rende qualsiasi curva di apprendimento di MRS impraticabile, per questo il software deve prevedere la possibilità di permettere la CAA in maniera del tutto elementare e comprensibile. In particolare si dovrà controllare il numero di azioni necessarie per comunicare qualcosa, la disposizione degli input e la bontà dei simboli.

**Abilità motorie:** Ogni giovane ospite della TIPed possiede un controllo sul movimento differente: si va da chi non può alzarsi dal letto ma può vedere e scrivere, a chi può solo rispondere sì o no chiudendo gli occhi, muovendo un dito o stringendo la mano. La predisposizione del sistema di input deve essere quindi "universale" così da rendere il software utilizzabile con qualsiasi mezzo. Le possibili combinazioni di input possono comprendere tastiera, mouse, track-

ball, joystick, gamepad, bottoni, dispositivi per l'eyetracking o per il Brain-Computer Interace (BCI) (vedi paragrafo 3.2 per informazioni più dettagliate). Inoltre il software dovrà adattarsi ad essere usato con un diverso numero di input, ad esempio con 2 input (avanti, seleziona) o 3 input (avanti, indietro, seleziona), in base alle abilità del bambino. Un altro importantissimo fattore che entra in gioco è quello del posizionamento del tablet nei pressi del letto. Alcune soluzioni possibili sono quelle di utilizzare dei sostegni per sorreggerlo oppure di attaccarlo ad uno schermo più grande posto ai piedi del letto, in ogni caso utilizzando dispositivi di input blueTooth.

**Abilità visive:** Il tablet è facilmente spostabile quindi si può avvicinare ed allontanare in base alle capacità visive del bambino. Molto importante è che MRS permetta la CAA anche da chi non ci vede, condizione in cui molti dei pazienti si trovano durante i primi momenti del processo di riabilitazione.

**Abilità uditive:** Esiste la possibilità che gli utenti non possano sentire a causa di problemi all'udito o perché l'audio deve essere disabilitato per non disturbare il sonno di eventuali compagni di stanza. I messaggi sonori generati da MRS non dovranno essere altro che la verbalizzazione delle scritte, per cui se si perde l'uscita vocale la CAA verrà garantita attraverso quella grafica.

**Tempi di apprendimento:** Il tempo che è possibile dedicare all'apprendimento di MRS è estremamente limitato. Oltre ad essere molto provante per i bambini mantenere la concentrazione per un tempo prolungato bisogna considerare anche che la durata dei ricoveri mediamente non supera in totale i sette giorni: ecco i dati relativi al 2010 sulla permanenza dei bambini presso la TIPed su un campione di 400 ricoveri.

- >7 giorni: 103
- >15 giorni: 46
- >30 giorni: 20 (di cui solo 4 con più di 3 anni)

Questo fa sì che MRS, almeno per la modalità 1, debba essere utilizzato efficacemente sin dal primo utilizzo.

## 2.2 Casi d'uso

Di seguito vengono presentati i casi d'uso principali individuati durante il processo di analisi. I casi d'uso in questione aderiscono alla standard UML2.0.

### UC1: Mind Reader Squared - Modalità 1

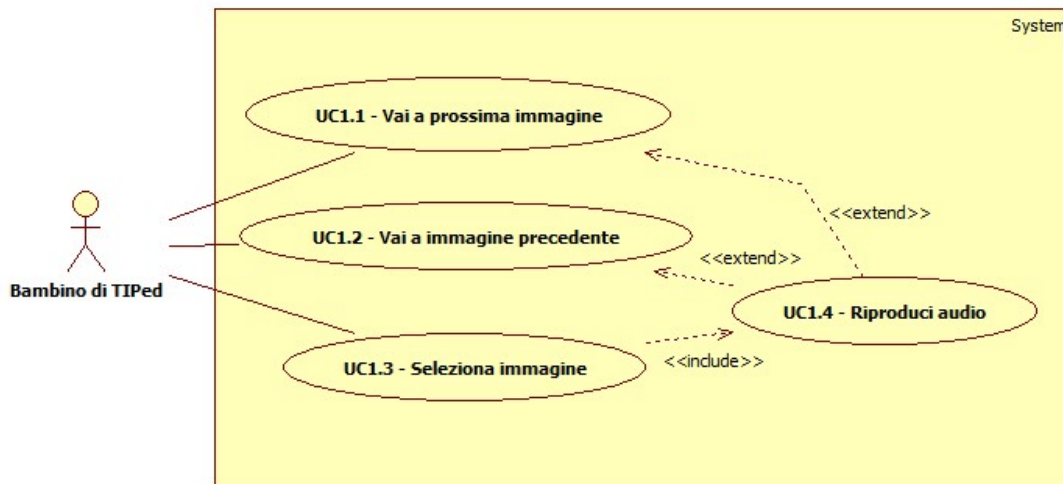


Figura 2.1: Diagramma UML dei casi d'uso che descrive le funzionalità della Modalità 1 di MRS

**Attore principale:** Bambino della TIPed

**Descrizione:** Questo caso d'uso rappresenta il funzionamento della modalità 1 all'interno del sistema Mind Reader Squared. Le funzionalità che questa modalità offre sono, e devono essere, assolutamente elementari.

**Pre-condizioni:**

L'applicazione è in esecuzione in uno qualsiasi dei suoi stati e l'utente decide di avviare l'attività Modalità 1. La configurazione che prevede la lettura di ogni testo è disattivata nello scenario principale mentre è attiva nello scenario secondario.

**Post-condizioni:**

L'applicazione ha risposto in modo adeguato a tutte le richieste dell'utente, ha salvato nel database i dati utili alla configurazione ed è pronta per avviare una qualsiasi tra le attività.

**Scenario principale:**

- Il bambino genera l'input che fa visualizzare l'immagine successiva o precedente (UC1.1,UC1.2);
- Ripete questa operazione, oppure quella di tornare indietro, a piacere (UC1.1,UC1.2);
- Il bambino seleziona l'immagine che sta visualizzando (tramite adeguato input) (UC1.3);
- MRS riproduce l'audio legato a quel simbolo (UC1.4).

**Scenari secondari:**

- Configurazione "leggi tutto"
  - Il bambino genera l'input che fa visualizzare l'immagine successiva o precedente (UC1.1,UC1.2);
  - MRS riproduce l'audio legato a quel simbolo (UC1.4);
  - Ripete queste operazioni a piacere (UC1.1,UC1.2,UC1.4).

## UC2: Mind Reader Squared - Gestione/Configurazione dei simboli

**Attore principale:** Familiare del bambino che utilizza MRS o infermiere della TIPed.

**Descrizione:**

Questo caso d'uso rappresenta in generale il processo di configurazione dei simboli in MRS.

**Pre-condizioni:**

L'applicazione è in esecuzione in uno qualsiasi dei suoi stati e l'utente decide di avviare l'attività di configurazione simboli.

**Post-condizioni:**

L'applicazione ha risposto in modo adeguato a tutte le richieste dell'utente, ha salvato

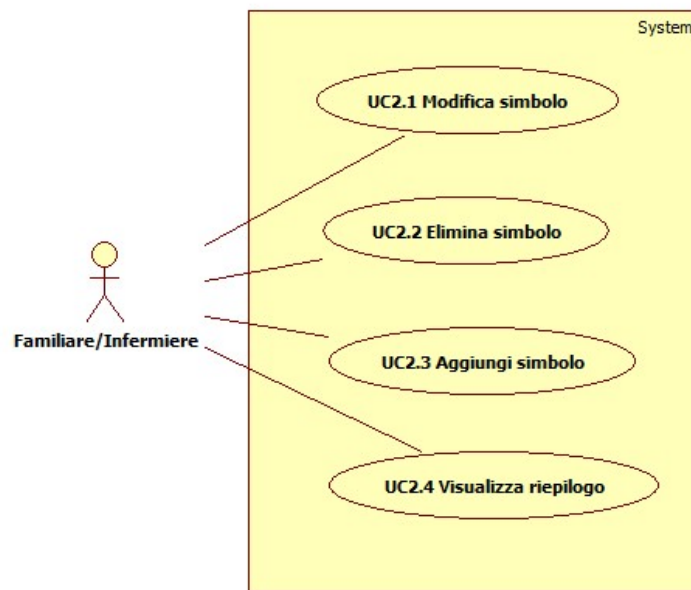


Figura 2.2: Diagramma UML dei casi d'uso che descrive le funzionalità di configurazione di MRS

nel database tutti i dati modificati ed è pronta ad avviare una qualsiasi tra le attività.

#### Scenario principale:

- L'utente esegue le operazioni consentite nell'ordine che preferisce fino a quando desidera (UC2.1,UC2.2,UC2.3,UC2.4).

#### UC2.1: Mind Reader Squared - Modifica simbolo

**Attore principale:** Familiare del bambino che utilizza MRS o infermiere della TIPed.

**Descrizione:** Questo caso d'uso descrive le operazione che l'utente esegue durante la modifica dei parametri di un simbolo: immagini, suoni, testi ecc. . .

#### Pre-condizioni:

L'applicazione sta eseguendo l'attività di configurazione e l'utente seleziona un sim-



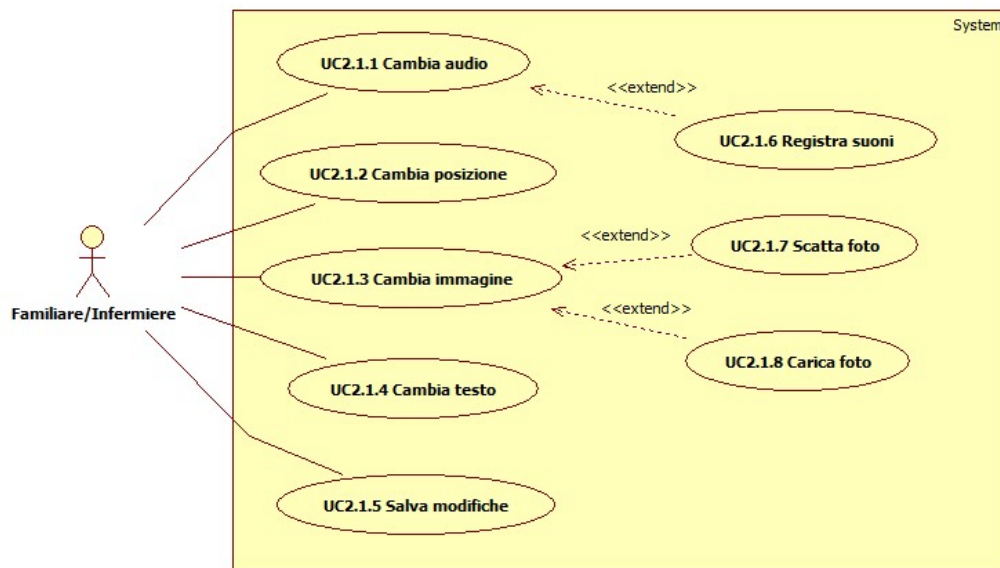


Figura 2.3: Diagramma UML dei casi d'uso che descrive la procedura di modifica di un simbolo in MRS.

bolo da modificare.

#### Post-condizioni:

L'applicazione ha risposto in modo adeguato a tutte le richieste dell'utente, ha salvato nel database tutti i dati modificati ed è pronta ad avviare una qualsiasi tra le attività.

#### Scenario principale:

- L'utente modifica i dati relativi al simbolo quali testo, immagine, suono, posizione (UC2.1.1,UC2.1.2,UC2.1.3,UC2.1.4).
- L'utente salva le modifiche effettuate, qualsiasi esse siano.

#### Scenari secondari:

- Registrazione audio
  - L'utente registra una traccia audio e continua con le modifiche.
- Scatto foto
  - L'utente scatta una foto tramite la videocamera e continua le modifiche.
- Carica immagine
  - L'utente carica un'immagine dal filesystem e continua le modifiche.

## 2.3 Lista dei requisiti stage

Lo sviluppo dei requisiti di seguito elencati porta alla prima versione di MRS che rappresenta la base per il Serious Game che verrà utilizzato in VOCA.TIO.

I requisiti sono suddivisi per tipologia (**F**unzionali, **Q**ualità, **V**incolo) e per componente (**M**odalità1, **C**onfigurazione, **D**atabase, **T**riage), le lettere finali del codice indicano invece la richiesta (**O**bligatorio, **D**esiderabile, **O**Pzionale).

### 2.3.1 Requisiti funzionali (RF)

#### **RFM1: Modalità 1 Applicazione equivalente allo strumento cartaceo già in uso**

L'applicazione deve garantire almeno le funzionalità, opportunamente digitalizzate, presenti nello strumento cartaceo da cui prende spunto. Questo strumento consiste in un libretto sfogliabile con una figura rappresentante uno stato/bisogno per ogni pagina.

- **RFM1.1-O:** Visualizzare la sequenza di immagini standard.
- **RFM1.2-O:** Selezionare una delle immagini.
- **RFM1.3-O:** Lettura di un testo che rappresenta la figura.
- **RFM1.4-D:** Gestire opportunamente l'ordine delle immagini.

#### **RFC: Configurazione e personalizzazione**

Componente fondamentale del sistema sarà la parte di configurazione che dovrà essere differenziata per ogni utente che utilizza il software permettendo così di creare una specie di profilo personale. Questa funzionalità verrà utilizzata probabilmente da infermieri e parenti dei bambini che poi faranno uso del programma.

- **RFC.1-O:** Inserimento dati personali dell'utente.
- **RFC.2-O:** Aggiungere/Modificare immagini e simboli.
- **RFC.3-D:** Impostare modalità di utilizzo preferita.
- **RFC.4-D:** Integrazione con strumenti di traduzione automatica

- **RFC.5-D:** Scattare foto da inserire come immagini
- **RFC.6-OP:** Caricare immagini dalla memoria del dispositivo
- **RFC.7-OP:** Registrazione di tracce audio.

### **RFT: Calibrazione/triage iniziale per automatizzare la scelta della modalità adatta**

La modalità di utilizzo del sistema e la calibrazione di alcuni parametri di accessibilità potranno essere determinati attraverso una sequenza di prove opportunamente studiate e presentate come gioco.

- **RFT.1-OP:** Divisione in tre livelli di difficoltà.
- **RFT.2-OP:** Valutazione delle capacità visive.
- **RFT.3-OP:** Valutazione delle capacità motorie.
- **RFT.4-OP:** Gestione del punteggio.

### **RFD: Gestione del database**

Tutte le informazioni di utilizzo e di configurazione acquisite dall'applicazione dovranno essere opportunamente persistite in un database.

- **RFD.1-O:** Salvataggio di un profilo personale per ogni utente con i dati personali e i dati di configurazione.
- **RFD.2-O:** Salvataggio di immagini e testi personalizzati.

## **2.3.2 Requisiti di qualità (RQ)**

### **RQI: Inclusività**

Verrà data particolare importanza all'analisi dei vari bisogni di tutti coloro che utilizzeranno questo sistema, sia per quanto riguarda le loro capacità durante il periodo di utilizzo sia per la necessità di utilizzare diversi dispositivi di input e output.

- **RQI.1-O:** Estrema versatilità nella gestione degli input.
- **RQI.2-D:** Accessibilità riguardo le dimensioni dello schermo, in particolare schermi di dimensione maggiore.
- **RQI.3-OP:** Portabilità della soluzione su più piattaforme.

### 2.3.3 Requisiti di vincolo (RV)

- **RV.1-O**: Il progetto dovrà essere sviluppato in modo open-source.

## 2.4 Lista dei requisiti progetto VOCA.TIO

Durante il lavoro di stage mi sono occupato anche svolgere una analisi dettagliata in vista del proseguimento del progetto MRS all'interno del progetto VOCA.TIO. Questa lista di requisiti descrive il prodotto che verrà utilizzato effettivamente all'interno del progetto di ricerca. Come si può notare i requisiti qui descritti rappresentano la continuazione del progetto di stage che ha prodotto il software di base.

L'elemento fondamentale che viene inserito è quello del Serious Game, che nella prima versione di MRS (prodotta durante il mio stage) non è ancora visibile.

I requisiti sono suddivisi per tipologia (**F**unzionali, **Q**ualità, **V**incolo) e per componente (**M**odalità1, **M**odalità2, **M**odalità3), le lettere finali del codice indicano invece la richiesta (**O**bligatorio, **D**esiderabile, **OP**zionale).

### 2.4.1 Requisiti funzionali (RF)

#### RFM1: Modalità 1

- **RFM1.1-O**: Inserire una scala di misurazione del dolore (scala di Eland).
- **RFM1.2-D**: Pacchetti aggiuntivi di immagini come bonus per il punteggio.
- **RFM1.3-OP**: Animazione delle immagini quando selezionate.

#### RFM2: Modalità 2 - Composizione di testo tramite simboli

In questa modalità verrà presentato un insieme di simboli che potranno essere selezionati in sequenza in modo da comporre dei pensieri/bisogni complessi. Ad ogni simbolo sarà associato anche un testo che lo rappresenta e sarà possibile vedere e sentire la frase fin'ora composta.

- **RFM2.1-O**: Visualizzare l'elenco dei simboli con relativo testo.
- **RFM2.2-O**: Visualizzare il testo (o i simboli) fin'ora composto.
- **RFM2.3-O**: Lettura del testo fino ad ora composto.

- **RFM2.4-D:** Trattare la sequenza di scelte in modo intelligente, utilizzando predizione.
- **RFM2.5-D:** Raggruppare i simboli per categorie (risposte/domande/affermazioni).
- **RFM2.6-O:** Memory (gioco).
- **RFM2.7-O:** Associazione (gioco).
- **RFM2.8-D:** Pacchetti aggiuntivi di immagini come bonus bonus per il punteggio.

### **RFM3: Modalità 3 - Composizione di testo tramite tastiera virtuale**

La modalità 3 offre la possibilità di utilizzare una tastiera virtuale semplificata per scrivere del testo di qualsiasi tipo. A questo si aggiungeranno funzionalità più complesse come la predizione o la trasmissione attraverso la rete.

- **RFM3.1-O:** Visualizzare una tastiera virtuale semplificata.
- **RFM3.2-O:** Lettura del testo finora scritto.
- **RFM3.3-D:** Utilizzare la predizione del testo.
- **RFM3.4-D:** Salvare/Caricare i testi.
- **RFM3.5-D:** Quizzettone (gioco).
- **RFM3.6-D:** Catena di parole (gioco).
- **RFM3.7-D:** Spelling (gioco).
- **RFM3.8-D:** Inviare/Ricevere testo scritto tramite SMS/eMail/Facebook.
- **RFM3.9-OP:** Tastiere virtuali aggiuntive come bonus bonus per il punteggio.
- **RFM3.10-OP:** Font aggiuntivi come bonus bonus per il punteggio.



---

## 3 Analisi tecnologica

### 3.1 Valutazione tecnologie, strumenti e linguaggi utilizzabili

Essendo alla mia prima esperienza nel settore delle applicazioni per dispositivi mobili mi sono avventurato nella ricerca dell'ambiente più adatto cercando di valutare ogni tipo di possibilità. In questa analisi ho privilegiato le soluzioni multiplatforma come richiesto dai requisiti di stage senza dover imboccare necessariamente il bivio Android, iOS. Di seguito vengono descritte le tecnologie e gli strumenti valutati:

**Appcelerator Titanium** è un progetto open-source nato nel 2006. Si è sviluppato molto velocemente soprattutto negli ultimi due anni ed ora nello sviluppo di applicazioni multiplatforma non ha rivali. Questa piattaforma mette a disposizione un **SDK** per Javascript che permette di scrivere applicazioni web, integrabili anche con HTML e CSS, che possono venire poi trasformate in applicazioni native mobili (Android e iOS) o Desktop (Windows, Mac, Linux). Le API di Titanium, infatti, permettono di accedere alle componenti tipiche dei dispositivi mobili come la videocamera e permettono anche l'utilizzo di alcune componenti dell'ambiente di sviluppo nativo come ad esempio i le Activity ed i Service di Android.

Se a questo aggiungiamo il fatto che sia estensibile con moduli del tutto indipendenti scritti in linguaggio nativo, che fornisce un ottimo IDE (Titanium Studio) come plug-in per **Eclipse** e che la comunità di sviluppatori è molto aperta e collaborativa diventa sicuramente la prima scelta per un progetto di questo tipo.

Purtroppo ho potuto constatare che il livello raggiunto nel trasferimento dell'applicazione in ambiente nativo, in particolare Android, non è ancora abbastanza efficace. Molte delle funzionalità previste producono bug indesiderati e la gestione della memoria è spesso fallimentare: bastano poche immagini per rendere l'applicazione inutilizzabile a causa di **memory leaks**. Comprensibile vista la

complessità e la continua evoluzione dei sistemi mobili, resta il fatto che Titanium non sia ancora pronto, e forse mai lo sarà, per permettere lo sviluppo di applicazioni che vadano oltre una semplice pagina web.

**HTML 5 con PhoneGap** come progetto è molto simile a Titanium: è open-source, ha un comunità molto attiva, e permette la creazione di applicazioni mobili multiplatforma a partire da pagine web. L'architettura di questa piattaforma è però più snella perché nel trasferimento dell'applicazione HTML, CSS, Javascript in ambiente nativo questa viene semplicemente inserita all'interno delle componenti grafiche adibite alla visualizzazione di contenuti web, `WebView` per Android e `UIWebView` per iOS. Le applicazioni PhoneGap si basano infatti su una pagina iniziale `index.html` in cui vengono incluse le API presenti in `phonegap.js`.

Questo limita sicuramente i bug di implementazione delle funzioni Javascript una volta eseguite nel dispositivo mobile e permette l'utilizzo dell'intero potenziale di HTML5 e CSS3. L'integrazione con le componenti dei sistemi mobili, invece, copre solamente le funzionalità fondamentali come ad esempio la videocamera e la gestione dei file ed inoltre queste API non sono abbastanza complete rispetto a quelle native. Anche per PhoneGap è prevista l'estensibilità tramite plugins ma, vista la scarsità della libreria (js), la quantità di codice da dover duplicare per Android e iOS sarebbe stata troppa. Così anche l'opzione di un'applicazione web in HTML5 multiplatforma non mi è sembrata adatta al progetto Mind Reader Squared.

**Android vs iOS** , scegliere non è stato molto difficile. Dovendo scegliere solamente uno tra i due sistemi ho identificato subito Android come candidato più adatto. Il fattore determinante della scelta deriva dalla scarsa, quasi nulla, compatibilità del sistema iOS con dispositivi di input di qualsiasi genere. Mentre iPad ha aperto solamente all'interfacciamento con ogni tipo di tastiera, Android ha aggiunto nel proprio SDK il supporto per molti dispositivi di input alternativo e senza alcuna limitazione di provenienza dei prodotti. La flessibilità massima nell'interfacciamento con input ed output è un elemento da cui MRS non può assolutamente prescindere.

**Android vs Android** . Con un sistema in continuo aggiornamento come quello di Android definire verso quali versioni indirizzare lo sviluppo di un'applicazione può essere molto delicato e difficile. Fortunatamente il contesto di utilizzo di



MRS è ristretto e ben definito e quindi si è potuto scegliere la distribuzione Android più adatta in base a quanto le sue caratteristiche coprissero i nostri requisiti. Trattandosi di un'applicazione per tablet il campo si è di gran lunga ristretto perché le versioni ottimizzate per tablet sono la 3.0 (Honeycomb) e tutte le successive. Dalla versione 3.1 sono state introdotte nel kit di sviluppo delle classi e dei metodi che permettono una gestione molto più profonda degli input alternativi come trackball e joystick. È stato deciso per questo motivo di sviluppare MRS per sistemi Android  $\geq 3.1$ .

## 3.2 Studio delle opzioni tecnologiche per l'input

Il mondo degli ausili informatici offre veramente una vasta gamma di prodotti tra cui poter trovare il dispositivo di input adatto ad ogni esigenza. Trackball, tastiere speciali, joystick e molto altro sono prodotti in ogni forma e dimensione e si interfacciano ai vari sistemi operativi come fossero normali mouse e tastiere. A questa categoria si possono fare due obiezioni: la prima riguarda la distribuzione ed il prezzo, almeno per quanto riguarda l'Italia sono poche le aziende che distribuiscono questi apparecchi ed i prezzi sono veramente esagerati. La seconda obiezione si riferisce al fatto che pochissimi di questi prodotti vengano venduti con interfaccia Bluetooth, tema che verrà trattato fra poche righe.

Maggiore interesse mi hanno suscitato le soluzioni tecnologicamente più avanzate la cui fase di ricerca e sviluppo non è che agli inizi e le cui potenzialità sono molto promettenti.

### 3.2.1 Eyetracking

L'eyetracking è il processo di misurazione della posizione e del movimento dello sguardo o dell'occhio. È facile immaginare come questa disciplina, oltre a molti altri utilizzi, sia importantissima nell'informatica pediatrica. Tra le possibilità vi è quella, ad esempio, di controllare con precisione un puntatore muovendo solamente gli occhi o la testa.

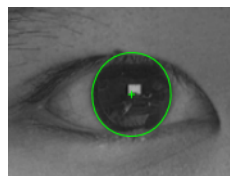
Anche se si sta ancora lavorando molto sullo sviluppo dell'eyetracking, molti risultati sono stati già raggiunti. I sistemi maggiormente utilizzati e che prenderemo in considerazione sono due: uno ha il grande vantaggio di essere minimamente invasivo e molto economico poiché si basa sul riconoscimento del movimento degli occhi tramite video (video analysis). Utilizzando una semplice webcam si possono rilevare

i movimenti dell'occhio con vari livelli di precisione dipendenti dalla complessità del software utilizzato ma anche dalla parte dell'occhio che viene analizzata, le possibilità in ordine di precisione sono: riflesso della cornea, pupilla, riflesso tra la cornea ed il cristallino ed infine i vasi sanguigni della retina.

L'altro interessante metodo per l'eyetracking ha bisogno di alcuni elettrodi posizionati intorno agli occhi poiché rileva i movimenti analizzando un segnale elettrico, Electrooculogram (EOG), da essi generato. Questa tecnica, sebbene più invasiva della precedente, ha il vantaggio di poter essere utilizzata anche con gli occhi chiusi o in condizioni di visibilità o posizionamento non ottimali.

Ecco alcuni esempi selezionati privilegiando soluzioni non commerciali:

**openEyes** è un toolkit open-source e open-hardware che comprende algoritmi di riconoscimento su video registrati ma anche un software per Linux/Windows che permette l'eyetracking real-time. Tutte queste componenti si basano sul riconoscimento del contorno tra l'iride e la sclera.



*Figura 3.1: Parte dell'occhio su cui si basano gli algoritmi di openEyes.*

**HeadMouse** è un'applicazione freeware per Windows prodotta all'Università di Lleida. Questa applicazione permette di utilizzare il mouse utilizzando una qualsiasi webcam ed i movimenti della testa. Può funzionare in collaborazione con una tastiera virtuale ed ha una vasta gamma di configurazioni possibili.

**ITU Gaze Tracker** è un eye-tracker open-source sviluppato da ITU GazeGroup presso l'Università di Copenhagen. Questo sistema molto preciso ha però bisogno di una videocamera a infrarossi con visione notturna.

### 3.2.2 Brain-Computer Interface (BCI)

La BCI rappresenta un mezzo di comunicazione diretta tra il cervello e un dispositivo esterno quale ad esempio un computer. Questa tecnologia ha come possibile obiettivo quello di controllare un PC utilizzando solo la mente.

In accordo con gli obiettivi di questo progetto, si è deciso di prendere in considerazione solamente quella branca della disciplina che prende il nome di “non-invasive BCI”

che prevede l'utilizzo di dispositivi che non richiedono alcun tipo di intervento chirurgico. La BCI non invasiva si basa sulla lettura degli elettroencefalogrammi (EEG). Nel passato sono già stati raggiunti ottimi risultati utilizzando elettroencefalografi professionali; ora la tecnologia ha raggiunto un livello tale da consentire l'utilizzo di dispositivi a basso prezzo ma che garantiscono una buona qualità.

I dispositivi portatili ed economici in questione sono:

**MindWave** è prodotto da NeuroSky, azienda leader nella produzione di chip per l'elaborazione di EEG che vengono utilizzati da altri produttori come PLX Devices e molti giochi. Questa cuffia si avvale di un solo sensore a secco ed ha un prezzo molto basso. Si presta soprattutto ai videogiochi in quanto il singolo sensore non permette grande flessibilità nella lettura dei dati ma è possibile gestire informazioni molto generali come ad esempio il livello di concentrazione.

**XWave** si avvale, come MindWave, del chip di NeuroSky. Le potenzialità ed il prezzo sono analoghe a quelle del prodotto precedente, questa cuffia possiede il vantaggio di avere l'interfacciamento con iPhone, iPad e Android tramite bluetooth. Fornisce inoltre gli SDK Android/iOS/PC/BB scaricabili gratuitamente.

**Emotiv EPOC (EE)** ha un prezzo leggermente più alto ma garantisce un EEG molto più preciso. Si avvale infatti di 14 sensori e permette anche l'utilizzo del giroscopio. Questa cuffia non prevede la comunicazione bluetooth diretta ma la trasmissione wireless avviene tramite un dongle USB e le SDK sono a pagamento.

Questo strumento sembra essere ottimo come dispositivo di input per Mind Reader Squared ma le SDK commerciali e l'interfaccia proprietaria fornita via USB e solo per Windows costituiscono dei limiti non da poco. Una possibile via per ottenere il funzionamento di EE su Android e MRS è costituita dall'utilizzo di Emokit, un progetto che tramite reverse engineering ha permesso di decriptare l'interfaccia proprietaria e rende disponibili da Windows, iOS e Linux i dati EEG grezzi. Uno strumento che potrebbe poi elaborare i dati grezzi trasformandoli in informazioni utili è OpenVibe.



*Figura 3.2: Cuffia per BCI Emotiv EPOC.*



# 4 Progettazione e sviluppo di Mind Reader Sqared

## 4.1 Selezione dei concetti e delle immagini

Per selezionare i concetti da inserire nella modalità 1 di Mind Reader Squared è stata innanzitutto stilata una lista da parte del personale della TIPed in cui sono raccolte le domande tipiche che vengono fatte ai bambini ricoverati.

1. vuoi la mamma?
2. vuoi il papà?
3. vuoi vedere i tuoi fratelli?
4. hai male?
5. dove hai male?
6. vuoi metterti seduto?
7. hai freddo?
8. vuoi che ti copriamo con una coperta?
9. hai caldo?
10. vuoi essere vestito?
11. vuoi vedere la televisione?
12. vuoi il tuo orsetto? la tua bambola? il tuo giocattolo preferito?
13. vuoi il ciuccio?
14. hai fame?

15. cosa vuoi mangiare?
16. Hai sete?
17. cosa vuoi bere?
18. ti scappa la pipì?
19. ti scappa la cacca?
20. Hai voglia di ascoltare della musica?
21. Come ti senti: triste, arrabbiato, stanco ?
22. Vuoi essere preso in braccio?
23. Hai sonno? Vuoi dormire?
24. Ti dà fastidio il tubo ET?
25. Ti dà fastidio il sondino ?
26. ti dà fastidio il catetere vescicale?

Da queste domande sono stati poi estratti i concetti che, attraverso un'immagine, dovrebbero essere identificati dal bambino. Dato il requisito di massima semplicità che la modalità 1 deve rispettare si è cercato di raggruppare i bisogni contenuti in questa lista in concetti più elementari possibile in modo da ridurre al massimo la possibilità di equivoci e confusione.

Per fare questa selezione è stato necessario coinvolgere anche un'infermiera della TIPed esperta di CAA ed una ricercatrice esperta in psicologia della riabilitazione. Si è deciso di accorpare domande simili come la 7 e la 8, dato che se il bambino vuole una coperta significa che ha freddo o la 17 e la 18 giacché non è facile e non ha senso discernere fra i due bisogni che comunque sono in assoluto i più importanti e frequenti per chi da anni non usa più il pannolino e vorrebbe farne a meno.

Le domande che prevedono una risposta più complessa del semplice sì/no o di localizzare un dolore o un fastidio sono state eliminate.

Questa decisione è stata presa principalmente per lasciare una separazione tra la complessità cognitiva dei vari livelli; infatti nelle modalità successive alla prima, compatibilmente alle abilità del bambino, sarà possibile espandere i concetti addentrando in sottocategorie più dettagliate. Ad esempio nella seconda modalità sarà prevista la

possibilità di completare la comunicazione di “mi fa male” con le scale di valutazione del dolore per indicare “quanto” ed un sistema per indicare “dove”, nozioni che appartengono ad uno stadio più avanzato di comprensione.

Il set di immagini, alla versione attuale, è unico anche se gli studi futuri portati avanti nel progetto VOCA.TIO porteranno alla creazione di set personalizzati rispetto alla nazionalità ed all’età del bambino.

Concetto	Immagine	Messaggio
Mamma		Voglio mia mamma
Papà		Voglio mio papà
Amici		Voglio i miei amici
Dolore/Fastidio		Mi fa male
Seduto		Voglio sedermi sulla sedia
Freddo		Ho freddo
Caldo		Ho caldo
<i>continua alla pagina successiva</i>		

Concetto	Immagine	Messaggio
Vestiti		Voglio vestirmi
Televisione		Voglio guardare la televisione
Giocattolo		Voglio i miei giocattoli
Ciuccio		Voglio il ciuccio
Fame		Ho fame
Sete		Ho sete
Bagno		Devo andare in bagno
Musica		Voglio ascoltare musica
Felicità		Sono felice
Tristezza		Sono triste
<i>continua alla pagina successiva</i>		






Concetto	Immagine	Messaggio
Paura		Ho paura
Stare in braccio		Voglio stare in braccio
Dormire		Voglio dormire

Tabella 4.1: Tabella dei concetti e delle immagini selezionate per la modalità 1 di MRS.

## 4.2 Decomposizione architetturale

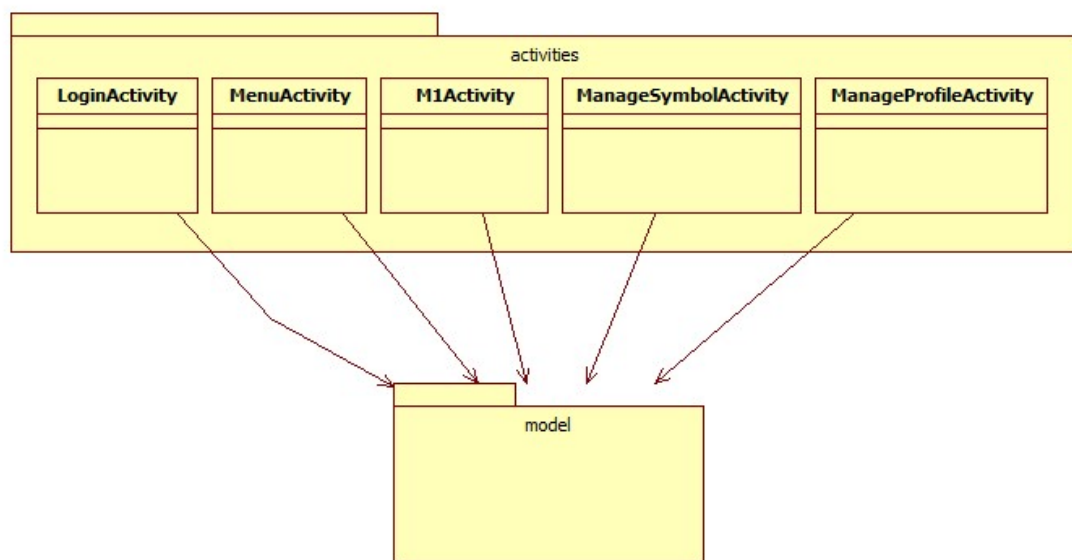


Figura 4.1: Diagramma UML delle classi che descrive l'architettura ad alto livello di Mind Reader Squared.

L'architettura delle applicazioni Android è relativamente semplice. Il framework mette a disposizione 4 componenti fondamentali con cui comporre il software, ognuna con funzioni diverse e ciclo di vita autonomo. Questo semplifica di molto la

progettazione, in particolare in MRS è stata utilizzata una sola tipologia di queste componenti: le activities. Un activity rappresenta una singola schermata con un'interfaccia utente, un'applicazione si compone di diverse activity che collaborano per rendere completa l'applicazione ma ognuna è indipendente dalle altre. Applicandole al design pattern **Model View Controller (MVC)** questi elementi architetturali previsti dal framework Android rappresentano l'insieme di view e controller. Per quanto riguarda il model ho previsto delle classi **Data Access Object (DAO)** per rappresentare le entità persistenti contenenti solamente i campi dati necessari in modo da essere utilizzate anche come oggetto leggero "da trasporto" mentre la logica di persistenza è stata separata in altre classi strutturate secondo il design pattern **singleton**.

Il sistema è composto quindi da 5 activity, o schermate, più una sesta da cui tutte le altre ereditano le procedure comuni del sistema. Altre sei classi costituiscono il model, 2 delle quali sono DAO, 3 singleton contenenti la logica di persistenza e 1 di gestione del database.

### 4.3 Database e logica di persistenza

Android fornisce completo supporto per database **SQLite**. In MRS si è deciso di gestire l'intera memoria persistente tramite un database le cui entità fondamentali sono i profili utente ed i simboli. Per gestire la creazione e gli accessi al db si utilizza la classe singleton **DatabaseHelper** che estende **SQLiteOpenHelper** e ridefinisce il metodo **onCreate()** per generare correttamente le due tabelle. Questa classe contiene inoltre alcuni metodi di utilità come ad esempio la criptazione/decriptazione **md5** e la pulizia delle stringhe da caratteri indesiderati.

#### ProfileDAO e SymbolDAO

Questi due oggetti DAO, molto utilizzati in tutte le activities, rispecchiano esattamente le entità del database. La loro struttura è essenziale, sono composti da: attributi, getters e setters ed un costruttore. Il costruttore ha come attributi i soli campi necessari ovvero quelli contrassegnati da "not null" nella definizione delle tabelle nel database.

Classe	Attributi
ProfileDAO	ID (int), password (String), firstName (String), lastName (String), birthDate (Date), language (String), preferredMod (int), registrationDate (Date), preferredLanguage (String), preferredTts (String), rating (int), openM2 (int), openM3 (int), readAll (int), virtualDpad (int), inputLeft (int), inputRight (int), inputSelect (int)
SymbolDAO	id (int), name (String), text (String), imageResource (int), imagePath (String), parent (int), m1 (boolean), m1Pos (int), m2 (boolean), m2Pos (int), soundPath (String), profileId (int), clicksM1 (int), soundChoise (int), clicksM2 (int)

Tabella 4.2: Tabella degli attributi di ProfileDAO e SymbolDAO.

Come si vede dalla tabella alcuni degli elementi che caratterizzano le due entità riguardano le modalità 2 e 3 non ancora implementate nella versione attuale di MRS.

ProfileDAO rappresenta un utente, con le proprie informazioni personali e le configurazioni desiderate. Questa entità sarà anche quella preposta a mantenere le impostazioni ed i dati relativi alle modalità di gioco (punteggi, statistiche).

SymbolDAO rappresenta un simbolo che può appartenere ad un solo profilo, questo perché anche se i simboli di default sono comuni a tutti gli utenti deve essere possibile personalizzarli in ogni momento. Lo sdoppiamento degli attributi per l'immagine è dovuto al fatto che le immagini di default si trovano tra le risorse dell'applicazione e sono reperibili tramite un id (int) mentre le foto scattate dall'utente si trovano nella memoria esterna del dispositivo e sono identificate da un URI (String). Uno di questi due campi si trova sempre a -1 e nel caricamento dell'immagine il sistema sceglierà sempre l'altro. Qui vengono anche salvati alcuni dati sull'utilizzo dell'applicazione come ad esempio i click per simbolo.

## ProfileLogic e SymbolLogic

Sono due classi singleton che forniscono tutti i metodi necessari alle activity per interagire con il database. Tutti i metodi di queste due classi contengono query al database e come attributi ed oggetti di ritorno hanno molto spesso i DAO.

Di seguito si descrive il funzionamento dei metodi più rilevanti.

ProfileLogic	
Metodo	Descrizione
<pre>public int registerProfile(Context context, ProfileDAO profile)</pre>	<p>Questo metodo si occupa di salvare nel database il nuovo profilo che è stato passato come parametro e successivamente generare e salvare tutti i simboli di default che saranno a disposizione del nuovo utente. L'id assegnato dal database al nuovo utente viene ritornato, oppure -1 in caso di errore. I simboli di default, immagini e testi, fanno parte delle risorse interne dell'applicazione così si rende necessario richiedere il passaggio come parametro del Context, oggetto molto pesante che racchiude tutti i dati di sistema relativi all'applicazione.</p>
<pre>public String[] queryWithId(int id, String[] columns)</pre>	<p>Questo metodo permette di svolgere query su un particolare profilo di cui si conosce l'id. I nomi delle colonne che sono passati in un array di stringhe vengono inseriti nel campo "select" della query ed i risultati vengono restituiti nello stesso ordine sempre all'interno di un array di stringhe. Questo evita di trasportare, quando non è necessario, l'intero oggetto DAO ma solamente le informazioni necessarie.</p>

*Tabella 4.3: Descrizione dei metodi più importanti di ProfileLogic*

SymbolLogic	
Metodo	Descrizione
<pre>public ArrayList[SymbolDAO] getM1SymbolList(int profile_id)</pre>	<p>Per prima cosa viene eseguita la query di ricerca dei simboli appartenenti alla modalità 1 del profilo desiderato. Per ogni simbolo restituito dalla query viene istanziato un equivalente oggetto DAO ed inserito poi in un <code>ArrayList</code>.</p> <p>L'array così ottenuto, prima di essere ritornato, viene ordinato secondo l'attributo "m1Pos" grazie ad un apposito <code>Comparator</code> definito come classe interna statica.</p>
<pre>public void deleteSymbol(final int symbol_id, final boolean m1, final boolean m2)</pre>	<p>I parametri m1 e m2 indicano se il simbolo deve essere cancellato rispettivamente dalla modalità 1 e/o dalla modalità 2 (la gestione dei simboli per la modalità 2 è funzionante anche se la modalità 2 non è prevista in questa versione di MRS). Prima della cancellazione del simbolo, sia per la modalità 1 che per la modalità 2, viene effettuato l'aggiornamento delle posizioni dei simboli successivi. Poi vengono aggiornati i campi m1 e m2 del simbolo in questione nel database e, se entrambi sono a -1, cioè il simbolo non è attivo per nessuna modalità, questo viene completamente cancellato dal database.</p>

Tabella 4.4: Descrizione dei metodi più importanti di *SymbolLogic*

## 4.4 Activities

In questa sezione verranno presentate nel dettaglio le activities che compongono MRS, sia nel layout grafico che nella logica di business.

Per comprendere a pieno il funzionamento di questi elementi è necessario analizzarne brevemente il ciclo di vita che, per ogni activity, compie diversi passaggi durante l'esecuzione dell'applicazione.

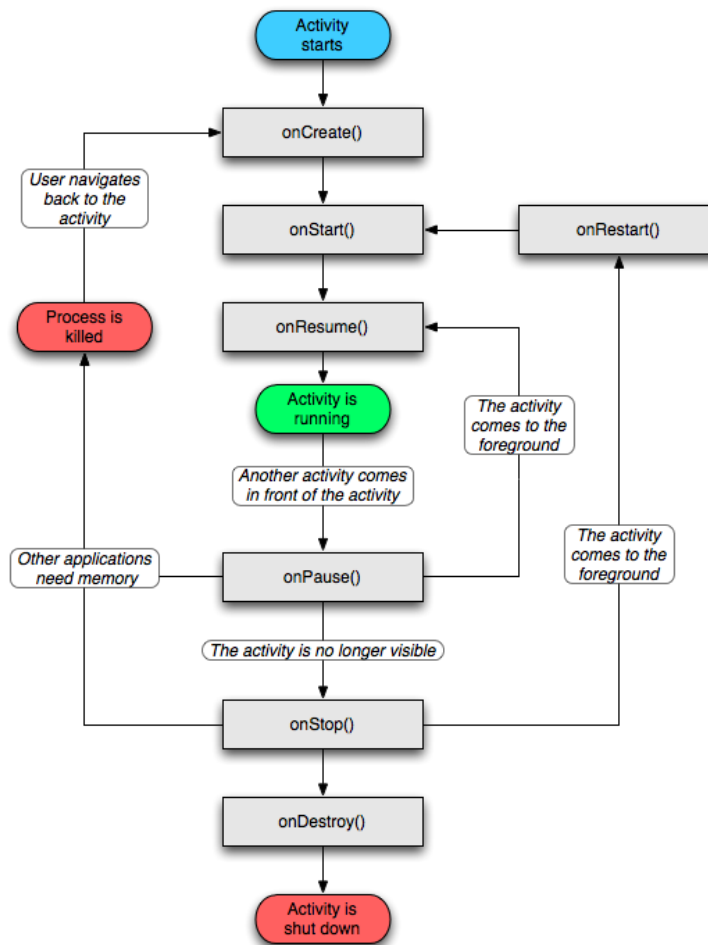


Figura 4.2: Schema che descrive il ciclo di vita di un activity nel framework Android.

Al lancio dell'applicazione viene visualizzata un'activity denominata "main activity". Questa activity, per svolgere altre operazioni, ha poi la possibilità di farne cominciare un'altra, e così via.

Ogni volta che un'activity inizia, quella precedente termina la sua esecuzione e viene messa in pausa e preservata all'interno di una pila (il "back stack"). Anche l'activity appena iniziata viene poi messa nella pila e successivamente visualizzata. Il "back stack" segue la politica del "last in, first out"; quando l'utente ha terminato

di interagire con l'ultima activity lanciata e decide di tornare indietro premendo il tasto "back" questa viene tolta dalla pila e viene risvegliata quella immediatamente sottostante.

La gestione del ciclo di vita delle activity, creazione - pausa - risveglio - distruzione, rappresenta il concetto fondamentale che sta alla base del framework Android che ne prevede la gestione tramite diverse **funzioni di callback**. Basti pensare che la maggior parte delle operazioni compiute da una activity avvengono all'interno del metodo `onCreate()`.

## BaseActivity

Prima di descrivere le activity visibili dall'interfaccia grafica è bene analizzare `BaseActivity` responsabile della logica comune a tutte le componenti del sistema. I compiti che questa classe deve gestire sono essenzialmente tre:

- Ottenere l'istanza dell'oggetto che serve a comunicare con il database;
- Recuperare i dati del profilo che lancia l'activity;
- Gestire il modulo di sintesi vocale (TTS).

Il funzionamento è quindi tanto semplice quanto importante. Ogni volta che viene creata una nuova activity questa, per prima cosa, ottiene l'istanza di `DataBaseHelper` (singleton) e successivamente la usa per richiedere al database i dati relativi all'utente che ha richiesto l'operazione. Per ultima cosa crea l'oggetto che permette di utilizzare il TTS e lo configura secondo le impostazioni dell'utente.

Ereditando da `BaseActivity` tutte le altre activity hanno accesso diretto ai dati del profilo corrente, posso fare richieste al database ed utilizzare il TTS già configurato a dovere.

### 4.4.1 Modalità 1

Questa activity, alla versione attuale di MRS, è l'unica che permette attraverso l'interfaccia grafica la CAA. Presenta molto semplicemente un'immagine a tutto schermo ed un testo che la descrive. L'immagine ed il testo sono inseriti rispettivamente negli elementi `ImageSwitcher` e `TextSwitcher`, che permettono di creare e visualizzare dinamicamente gli oggetti grafici con buona efficienza. La sequenza dei simboli (immagine più testo) segue un preciso ordine stabilito dall'utente tramite la configurazione e lo scorrimento avviene in modo circolare, dopo l'ultimo segue il primo. È

possibile interagire con questa activity attraverso tre operazioni: visualizza prossimo simbolo, visualizza simbolo precedente e seleziona simbolo.

Sono stati separati gli elementi di interfaccia da quelli di business logic inserendo la logica di rotazione dei simboli nella classe `SymbolList` appartenente al model. Questa classe, molto semplice, funge da interfaccia tra il database e l'activity elaborando la lista di simboli estratti dalla componente di persistenza tramite un iteratore in modo da fornire all'esterno solamente i metodi necessari: `start()`, `next()` e `previous()`, i quali, oltre a mantenere aggiornato lo stato dell'iteratore, ritornano il simbolo desiderato.



Figura 4.3: Screenshot dell'applicazione mentre è in uso la modalita 1.

Molto importante per questa activity è la gestione degli input, infatti deve essere in grado di rilevare tutte le diverse modalità previste dall'applicazione ma reagire solamente a quelle selezionate dall'utente durante la configurazione. Per fare ciò vengono utilizzati i callback propri dell'activity che agiscono quindi su tutto lo schermo e non solo su un elemento dell'interfaccia grafica: `onKeyDown`, `onKeyLongPress`, `onKeyUp` e `onTouchEvent`. Ognuna di queste funzioni lanciate dell'activity in reazione ad un evento generato dell'utente è necessaria per valutare di quale input si tratti tra quelli attualmente previsti da MRS, che sono:

- `INPUT_MOUSE_CLICK`: un click con il tasto sinistro di mouse e trackball o un tocco sul touch screen.
- `INPUT_MOUSE_LONG_CLICK`: un click prolungato con il tasto sinistro di mouse e trackball o un tocco prolungato sul touch screen.



- `INPUT_MOUSE_SLIDE_RIGHT`: spostarsi verso destra tenendo premuto il tasto sinistro di mouse e trackball o touch screen.
- `INPUT_MOUSE_SLIDE_LEFT`: spostarsi verso sinistra tenendo premuto il tasto sinistro di mouse e trackball o touch screen.
- `INPUT_KEY_CLICK`: premere uno qualsiasi dei tasti della tastiera.
- `INPUT_KEY_LONG_CLICK`: premere a lungo uno qualsiasi dei tasti della tastiera.
- `INPUT_KEY_LOCATION_LEFT_CLICK`: premere uno qualsiasi dei tasti presenti sul lato destro della tastiera (compreso il tastierino numerico).
- `INPUT_KEY_LOCATION_RIGHT_CLICK`: premere uno qualsiasi dei tasti presenti sul lato sinistro della tastiera.
- `INPUT_KEY_SPACE_CLICK`: premere il tasto “spazio”.
- `INPUT_KEY_LEFT_CLICK`: premere il tasto “freccia sinistra”.
- `INPUT_KEY_RIGHT_CLICK`: premere il tasto “freccia destra”.
- `INPUT_KEY_ENTER_CLICK`: premere il tasto “invio”.
- `INPUT_KEY_UP_CLICK`: premere il tasto “freccia in alto”.
- `INPUT_KEY_DOWN_CLICK`: premere il tasto “freccia in basso”.

In base alla configurazione scelta dall’utente è così possibile utilizzare questa activity con moltissime modalità di input. Ad esempio una modalità ad un solo tasto potrebbe essere: “prossimo simbolo” con click normale e “seleziona simbolo” con click lungo, permettendo la CAA con il solo utilizzo del tasto sinistro del mouse o di un tasto della tastiera o del tocco del touchscreen.

#### 4.4.2 Configurazione simboli

Per gestire la complessità di questa activity sono stati usati quattro fragment. I fragment, inseriti nella libreria di Android dalla versione 3.0, rappresentano una porzione dell’interfaccia utente nell’activity e possiedono un proprio ciclo di vita e ricevono i propri eventi di input. Questi fragment possono essere combinati fra di loro a formare l’intera interfaccia dell’activity con la possibilità di essere inseriti e rimossi mentre questa è in esecuzione.

Anche se le quattro classi `fragment` possiedono un ciclo di vita ed un'interfaccia indipendenti si è deciso di far confluire, tramite appositi `callback`, gli eventi principali da essi generati sull'activity `ManageSymbolActivity`. Questa classe quindi non gestisce un layout grafico ma è incaricata dell'implementazione dell'intera business logic che comprende anche il coordinamento dei quattro moduli che la compongono.

Per capirne meglio il funzionamento bisogna analizzare nel dettaglio i quattro `fragment`:

**TabFragment** è l'unico ad essere fisso ed occupa in ogni momento la metà sinistra della schermata. Questo elemento è composto da 3 schede, visualizzabili una alla volta, e consente di cambiare scheda in qualsiasi momento. La prima e la seconda scheda contengono la lista ordinata dei nomi dei simboli rispettivamente delle modalità 1 e 2. La terza scheda contiene una pagina vuota perché viene utilizzata come se fosse un bottone per segnalare all'activity di visualizzare il form per l'inserimento di un nuovo simbolo.

Poiché la gestione delle schede è già abbastanza complessa, l'implementazione delle liste della prima e della seconda scheda è affidata ad una classe interna `SymbolListManager`. Questa classe si occupa di costruire la grafica attraverso una `ListView` e mantenerla aggiornata gestendo un `ArrayAdapter`. La logica di gestione della lista è tutta contenuta in questa classe, che `TabFragment` istanzia due volte, una per gestire i simboli della modalità 1 e una per gestire i simboli della modalità 2.

Le operazioni possibili su questo `fragment` generano i seguenti `callback` gestiti da `ManageSymbolActivity`:

- **onChangeTab(int,int)** quando viene selezionata una scheda; i parametri indicano l'indice della nuova scheda e di quella precedente. `ManageSymbolActivity` lo gestisce verificando la presenza o meno di modifiche non salvate e, in caso vi siano, permette all'utente di decidere se mantenerle o scartarle. Fatto questo rimuove un eventuale `OptionFragment` presente sul lato destro dell'activity. Successivamente se la scheda selezionata è la prima o la seconda istanzia un `SummaryFragment` per la modalità corrispondente; se è la terza istanzia un `OptionFragment` per l'inserimento di un nuovo simbolo.
- **onSymbolSelected(int,int)** quando viene selezionato uno dei simboli dalla lista della prima o della seconda scheda; i parametri indicano l'indice

della scheda e quello del simbolo. Viene gestito facendo il controllo sulle modifiche e successivamente rimpiazzando, se presente, l'`OptionFragment` con un altro relativo al simbolo selezionato.



Figura 4.4: Interfaccia grafica di `TabsFragment` (screenshot).

**OptionsFragment** contiene il form di visualizzazione, modifica ed inserimento di un simbolo. È costituito da molti elementi grafici perché deve permettere la personalizzazione di tutti gli attributi di un simbolo. Molto importante è il controllo degli elementi grafici in questa classe perché le possibilità di interazione con l'utente sono moltissime, quasi tutto è cliccabile. Ogni intervento che l'utente fa su questo fragment provoca il cambiamento di uno o più attributi del simbolo, questi vengono mantenuti in una memoria cache e salvati solamente se viene premuto il bottone corrispondente; altrimenti è possibile ripristinare le impostazioni cancellando le modifiche provvisorie.

In **OptionsFragment** sono implementate anche alcune funzionalità abbastanza complesse come la possibilità di scattare foto con anteprima, il salvataggio di queste in memoria e la registrazione di tracce audio.

I callback generati da questo fragment e gestiti da `ManageSymbolActivity` sono:

- **onSave(SymbolDAO)** quando viene premuto il tasto “salva”. Viene gestito salvando le modifiche sia nel database che nella cache temporanea.
- **onDelete(int)** quando viene premuto il tasto “cancella”. Viene gestito cancellando il simbolo dal database solamente per la modalità/scheda selezionata. Viene inoltre rimosso l' `OptionFragment` dall'interfaccia grafica.

- **onMoveUp(int)** quando viene premuto il tasto “sposta verso l’alto”. Provoca l’aggiornamento della lista in `TabsFragment` e dell’attributo posizione nella cache. Queste modifiche sono reversibili.
- **onMoveDown(int)** quando viene premuto il tasto “sposta verso il basso”. La gestione è analoga a “sposta verso l’alto”.
- **onModify()** ogni qualvolta viene prodotta una modifica al simbolo corrente. Viene gestito rendendo cliccabile il tasto “salva” se prima non lo era.
- **onLoadImage(int)** quando viene premuto il tasto “carica immagine”. Viene gestito istanziando all’interno dell’activity un `SelectImageFragment`.



Figura 4.5: Interfaccia grafica di `OptionsFragment` mentre viene scattata una foto (screenshot).

`SelectImageFragment` estende la classe `DialogFragment` e quindi si tratta di un fragment che viene visualizzato come un dialogo. La sua funzione è quella di permettere all’utente di scegliere una tra le foto scattate in passato da inserire come immagine per il simbolo corrente.

Il suo contenuto viene generato prelevando l’immagine di default assegnata al simbolo corrente e le ultime cinque foto scattate dall’utente. Il dialogo conterrà così una lista a selezione singola con queste sei immagini. Per visualizzare nella lista del dialogo le immagini si è reso necessario sviluppare un’implementazione ad hoc della classe `BaseActivity` modificandone la creazione della grafica che di default sarebbe una semplice testo.



Figura 4.6: Interfaccia grafica di *SelectImageFragment* (screenshot).

**SummaryFragment** è puramente grafico e non permette interazioni con l'utente. Il suo scopo è quello di visualizzare i dati riassuntivi di una modalità di utilizzo di MRS. Contiene la lista di tutti i simboli della modalità selezionata in ordine per numero di clck e disposti in tabella.

Modalità 1 Info	
Testo	N. Click
Sono felice	7
Ho fame	6
Voglio stare in braccio	3
Voglio il ciuccio	2
Voglio mia mamma	2
Voglio dormire	1
Ho paura	1
Voglio ascoltare musica	1
Ho sete	1
Voglio i miei amici	1
Sono triste	0
Devo andare in bagno	0
Voglio i miei giocattoli	0

Figura 4.7: Interfaccia grafica di *SummaryFragment* (screenshot).



Figura 4.8: Interfaccia grafica di `ManageSymbolActivity` mentre l'utente sta digitando un testo per modificare un simbolo (screenshot).

### 4.4.3 Configurazione utente

Questa activity di configurazione ha una logica grafica abbastanza complessa ma non contiene fragment al contrario di `ManageSymbolActivity` perché è sostanzialmente statica.

Alla sua creazione vengono caricati tutti i dati dell'utente molti dei quali possono essere modificati attraverso questa schermata. Tutte le modifiche vengono salvate direttamente nel database senza bisogno di conferma.

In alto a sinistra si trova il pannello riassuntivo delle informazioni dell'utente. Qui troviamo anche due menù a tendina che permettono di impostare la lingua del sistema e la lingua del text-to-speech. I due metodi che gestiscono questi menù sono i più complessi per questa activity e compiono diverse operazioni. Il popolamento delle lingue disponibili per il TTS richiede che ne venga verificata la presenza nel dispositivo ad una ad una ed anche la presenza di un "motore" per il TTS deve essere accertata, altrimenti ne viene proposta l'installazione dall'**Android Market**. Viene mostrato un link anche per scaricare la voce nella madrelingua dell'utente nel caso questa non fosse già presente nel dispositivo. Molto importante è anche la gestione del cambio della lingua di sistema; quando questo avviene per prima cosa viene soppressa l'activity, modificata l'impostazione di localizzazione del dispositivo e subito riavviata la stessa activity. Questo fa sì che al riavvio tutte le risorse caricate siano quelle relative alla lingua appena selezionata dall'utente. Fatto ciò viene avviato un thread in background che aggiorna anche i testi di tutti i simboli in modo che l'applicazione per la CAA sia subito utilizzabile nella nuova lingua.

Sul lato destro vi sono le configurazioni relative ad input e output. La personalizzazione dell'output in questa versione di MRS prevede la selezione o meno dell'impostazione “leggi tutto” ed è modificabile tramite un piccolo bottone da spuntare. La configurazione dell'input presenta tre menù a tendina rispettivamente per le azioni “avanti”, “indietro” e “seleziona” che sono le operazioni fondamentali di ogni modalità progettata per MRS. Tutti i menù permettono di selezionare una qualsiasi tra le tipologie di input previste dal sistema (descritte nella sezione “Modalità 1”).

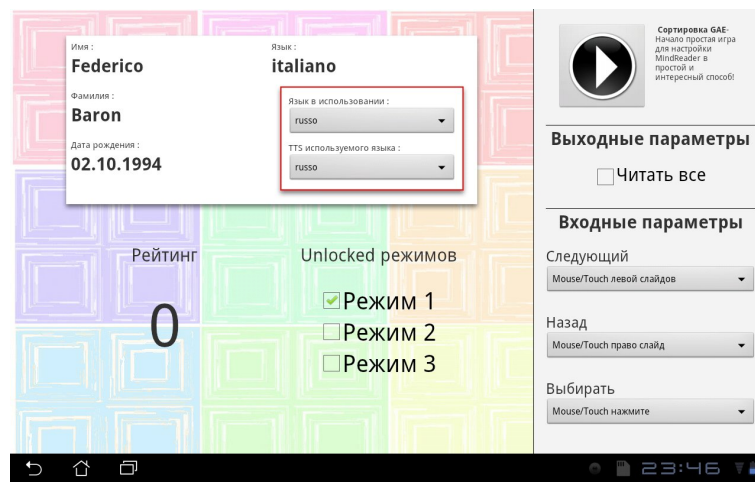


Figura 4.9: Interfaccia grafica di *ManageProfileActivity* di un utente che utilizza il russo come lingua preferita (screenshot).



Figura 4.10: Interfaccia grafica di *ManageProfileActivity* di un utente che utilizza l'italiano come lingua preferita (screenshot).

#### 4.4.4 Login

Questa semplice activity garantisce le funzionalità base di login e registrazione. I controlli eseguiti sui dati inseriti sono basilari poiché tutti i campi sono obbligatori. Un utente registrato viene immediatamente inserito nel database e può accedere all'applicazione con i propri dati. Le password vengono inviate al database e salvate con criptazione md5.

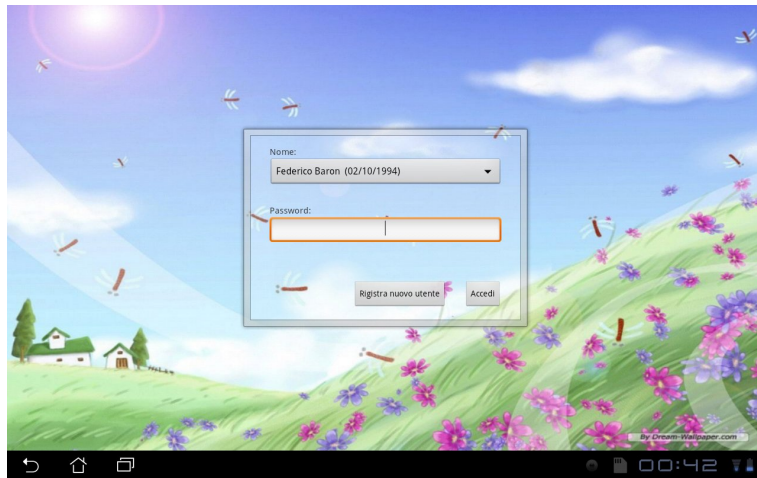


Figura 4.11: Interfaccia grafica di LoginActivity, form di login (screenshot).

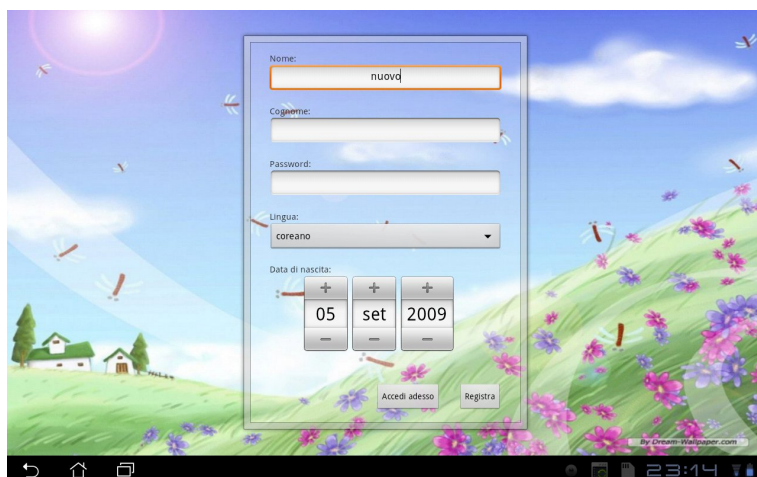


Figura 4.12: Interfaccia grafica di LoginActivity, form di registrazione (screenshot).



### 4.4.5 Menu

`MenuActivity` rappresenta il crocevia principale nei “traffici” di MRS. Questa activity, che viene lanciata sempre dopo il login, possiede solamente cinque bottoni che permettono di accedere alle cinque schermate dell’applicazione. Ogni volta che si esce da una schermata viene rilanciata `MenuActivity` da cui è possibile scegliere un’altra destinazione e così via.



*Figura 4.13: Interfaccia grafica di MenuActivity (screenshot).*



# 5 Conclusioni

## 5.1 Obiettivi raggiunti

Gli obiettivi prefissati sono stati tutti soddisfatti durante il lavoro di stage ad eccezione di due tra i requisiti opzionali anche se questi sono stati ampiamente discussi con il tutor di stage, e la loro esclusione dal software prodotto è stata opportunamente motivata.

**RFT-OP - Calibrazione/triage iniziale:** questa componente è stata analizzata in un breve studio di fattibilità che ha portato alla decisione, al termine della fase di progettazione, di escluderlo dalla prima versione di MRS, cioè quella da me sviluppata. La motivazione principale è stata che la sola modalità 1 non è sufficientemente configurabile per rendere veramente efficace un sistema di “trriage”. Inoltre la volontà di proporlo come gioco richiede un più accurato studio dell’interfaccia grafica.

**RQUI.3-OP - Portabilità della soluzione su più piattaforme:** questo requisito, nato precedentemente alla ricerca tecnologica svolta rispetto alle possibilità di sviluppo per multipiattaforma, è risultato impraticabile una volta imboccata la via dello sviluppo in ambiente nativo. Le considerazioni che hanno portato a questa scelta, sempre in accordo con il mio tutor, sono esposte nel paragrafo 3.1.

Rispetto al piano di lavoro vi è stato un leggero scostamento nel periodo iniziale dovuto alla necessità di spendere delle ore aggiuntive nell’analisi dei requisiti, a fronte alle 36 prefissate. Il progetto Mind Reader Squared è stato accolto di buon grado nell’ambiente della TIPed e in molti si sono resi disponibili ed hanno contribuito alla concretizzazione dell’idea iniziale da cui ha preso vita lo stage: trasferire su tablet uno strumento cartaceo preesistente. L’essermi adoperato per metabolizzare come funzioni la vita in una TIPed, sia per chi ci lavora che per i bambini che qui vengono

curati, non è stato affatto facile e sicuramente non una cosa da prendere alla leggera. Ogni singolo minuto speso presso il dipartimento, discutendo con il mio tutor e con lo staff infermieristico, partecipando ad assemblee, entrando di persona nel reparto TIPed, è stato fondamentale per poter svolgere una buona analisi dei requisiti anche se questo mi è costato un piccolo surplus nelle ore. Il tutor Roberto Mancin e le altre persone coinvolte nel progetto VOCA.TIO, a cui è stata sottoposta una prima versione dell'analisi dei requisiti, hanno espresso soddisfazione ed hanno introdotto il MRS così definito come base tecnologica del progetto.

In virtù del buon lavoro fatto precedentemente è stato possibile svolgere le fasi di progettazione e sviluppo in modo molto regolare e senza rilevanti cambi di direzione fino al raggiungimento degli obiettivi prefissati.

Un po' carente il processo di verifica e validazione a cui non sono riuscito a dedicare tutte le ore previste dal piano di lavoro. In ogni caso i test effettuati (anche su dispositivo reale e non solo su emulatore) hanno dato tutti esiti positivi e questa mancanza non ha minimamente compromesso il buon risultato ottenuto dallo stage.

## 5.2 Sviluppi futuri

Il progetto iniziato e portato avanti in questo lavoro di stage, come è stato chiaro fin dall'inizio, è un punto di partenza per qualcosa di molto più grande. Il software MRS è destinato a comprendere tutte le funzionalità e le qualità emerse dallo studio di analisi fatto nella prima fase di questo stage, descritti nel paragrafo 2.4, ed essere poi utilizzato a fini di ricerca nel progetto VOCA.TIO.

Credo che il buon lavoro fatto fin'ora possa lasciare spazio all'estensione del software, da me prodotto ed opportunamente documentato, da parte di qualsiasi programmatore senza dover perdere troppo tempo in costose correzioni o riprogettazioni. Inoltre i buoni rapporti instaurati con il personale del Dipartimento di Pediatria mi invogliano a non uscire completamente dal percorso iniziato ed a prolungare la mia disponibilità, se desiderato, nell'aiutare e collaborare con chi prenderà in mano il testimone per proseguire.

Il proseguo di Mind Reader Squared sarà estremamente interessante anche dal punto di vista tecnologico e dell'innovazione. Tra i primi aspetti da considerare, infatti, vi sarà la progettazione di una componente per la ricezione di dati via bluetooth provenienti da dispositivi di tipo BCI, ma più in generale dovrà essere ampliata la capacità di interfacciamento con altri strumenti per l'input. La progettazione di questa com-

ponente dovrà comprendere inoltre una strategia di lettura di dati grezzi provenienti da dispositivi, la maggior parte, che non forniscono il supporto per sistema Android. Un esempio è sicuramente Emotiv Epoc (EE) che, inoltre, trasmette tramite un protocollo proprietario ma sul quale è già stata fatta una buona analisi.

Senza dubbio, il meglio deve ancora venire!

## 5.3 Considerazioni personali

Sono molto contento di aver intrapreso questa esperienza di stage e sono convinto che ciò che mi porto a casa come conoscenza acquisita mi sarà certamente utile in futuro, sia dentro che fuori dall'ambito dell'informatica.

La cosa che più mi è piaciuta e che mi ha entusiasmato fin da subito è stata la completa dispinibilità, del personale della TIPed e del mio tutor Roberto, nel condividere ciò che avevano in mente e nel mettermi nelle condizioni di comprendere a pieno lo spirito, le necessità e gli obiettivi di questo progetto. Grazie a Roberto che, giorno dopo giorno, ha condiviso con me le sue conoscenze nel campo dell'informatica pediatrica e mi ha fatto conoscere l'ambiente della TIPed in ogni suo aspetto ho potuto imparare come l'accessibilità, non intesa come rispetto degli standard ma come "inclusività", sia uno strumento potentissimo con cui l'informatica ha la possibilità di fare molto.

Entrare in sintonia con l'ambiente, apprenderne i concetti e capirne i bisogni mi hanno aiutato enormemente durante tutte le fasi del progetto e, l'interazione con l'ambiente a cui è destinato credo sia un aspetto indispensabile dello sviluppo software che nella maggior parte dei casi viene trascurato.

Per portare a termine questo stage ho potuto fare affidamento su quanto imparato negli anni di studio. In particolare il corso di Ingegneria del Software, i cui concetti ho cercato di applicare, che mi ha permesso di gestire il progetto in modo efficace. Anche i corsi di programmazione mi sono stati indispensabili per comprendere, in tempi brevi, le possibilità tecnologiche disponibili e successivamente i meccanismi del framework Android. La quantità enorme di documentazione studiata per le varie attività di questo stage, dalle ricerche sulla CAA, agli algoritmi di eyetracking, passando per la documentazione di Android, mi ha permesso di apprendere molte cose

a me sconosciute e che, in tutti i casi, ho trovato molto interessanti.

Sono convinto che, dal punto di vista formativo, la scelta di questo stage sia stata molto buona e, se pur con caratteristiche molto differenti, non abbia nulla da invidiare ad uno stage presso un'azienda informatica.

L'opportunità di fare analisi critica su cosa e come possa fare l'informatica in una TIPed credo che non sia una cosa che si faccia dappertutto. Non comune è poi la possibilità, già verificatasi, di sperimentare sul campo il software e di partecipare nella primissima fase di un progetto universitario di ricerca importante come lo è VOCA.TIO.

La mia soddisfazione è anche un auspicio alla continuazione di questa collaborazione tra gli stage del corso di Informatica e la TIPed di Padova.

Ringrazio Roberto che, come mio tutor, ha fatto sicuramente di più di quello che gli era stato chiesto. Dedicando moltissimo del suo tempo del suo entusiasmo e delle sue conoscenze a questo progetto è stato sicuramente determinante nella buona riuscita dello stage.

Altri ringraziamenti vanno alle persone che ho conosciuto tra lo staff della TIPed e i membri del progetto VOCA.TIO che si sono rese disponibili ad aiutarmi in modo del tutto positivo.

## 6 Glossario

**Android Market:** Android Market è un sito web raggiungibile da

*<https://market.android.com/>* che offre un accesso facile e veloce a una vasta gamma di applicazioni sviluppate appositamente per la piattaforma Android. Tali applicazioni sono state create da sviluppatori di tutto il mondo e sono state votate da utenti di Android. Per registrarsi al portale è necessario accedervi con un dispositivo Android, smartphone o tablet.

**CAPTCHA:** Con l'acronimo inglese CAPTCHA si denota nell'ambito dell'informatica un test fatto di una o più domande e risposte per determinare se l'utente sia un umano e non un computer.

**Data Access Object (DAO) (Design Pattern):** Il DAO (Data Access Object) è fondamentalmente una classe che rappresenta un'entità tabellare di un database. L'utilizzo di questo design pattern permette di stratificare e isolare l'accesso ad una tabella dalla parte di business logic.

**Eclipse:** Eclipse è una piattaforma di sviluppo che offre potenti strumenti per la creazione, la pubblicazione e la gestione dei software durante tutto il loro ciclo di vita. La principale caratteristica di questo ambiente di sviluppo è l'estensibilità grazie all'enorme numero di plugins disponibili per ampliarne le funzionalità. L'intero progetto Eclipse è open-source.

**Funzione di callback:** In ambiente Android/Java i callback sono metodi che vengono invocati da oggetti posti a livello più basso nel software rispetto alla classe che li implementa. La dichiarazione di questi metodi viene inserita solitamente in delle interfacce che prendono il nome di "listeners" e i nomi dei metodi iniziano con "on". Alcuni esempi di callback appartenenti ad un "event listener" sono: `onTouchEvent()`, `onKeyEvent()`, `onMotionEvent()`.

**GRUPPO INFORMATICO di PEDIATRIA (GRINPED):** GRINPED è formato dallo staff del Servizio Informatico del Dipartimento di Pediatria Salus Pueri. Oltre

a svolgere attività di amministrazione del sistema informatico nell'ospedale pediatrico, si occupano anche di didattica facendo formazione rivolta ad infermieri e segreterie riguardo l'applicazione di soluzioni innovative per favorire la comunicazione telematica.

**International Classification of Functioning (ICF):** L'International Classification of Functioning, o Classificazione dello stato di salute, definisce lo stato di salute delle persone piuttosto che le limitazioni, dichiarando che l'individuo "sano" si identifica come "individuo in stato di benessere psicofisico" ribaltando, di fatto la concezione di stato di salute. Introduce inoltre una classificazione dei fattori ambientali.

Questa classificazione, commissionata negli anni '90 dall'OMS, si contrappone alla precedente ICIDH; in sostanza l'ICIDH valutava i fattori di disabilità iniziando dalla menomazione, mentre l'ICF valuta le abilità residue dell'individuo.

**Lis e Bliss:** LIS, la lingua dei segni italiana, è una forma di comunicazione che si basa su segni eseguiti con una o due mani. Ogni segno corrisponde ad uno o più significati precisi e codificati, come avviene per le parole.

Il Bliss è un linguaggio grafico internazionale che fu creato inizialmente, fra il 1942 e il 1965, per abbattere le barriere tra i popoli. È un sistema di simboli grafici basato sul significato e non sulla fonetica: pittografici (assomigliano a quello che rappresentano), ideografici (simboleggiano un'idea), internazionali (termini matematici, cifre), arbitrari (senza legame col concetto rappresentato).

**Locked in funzionale:** Locked in funzionale (o reattività muta (Alexander 1982)): è uno stato attraverso cui passa chiunque ogni volta che ritorna cosciente. La coscienza non risiede in un singolo neurone ed il risveglio della coscienza da sonno, sedazione o coma non è un processo on/off ma è graduale e può durare meno di un secondo (secchiata d'acqua gelida), ore (risveglio da anestesia totale o coma farmacologico) o giorni (risveglio da coma patologico). Alcuni ospiti della TIPed sono minimamente coscienti e non ancora in grado di controllare volontariamente tutti i muscoli però possono e devono poter comunicare i bisogni principali (fame, sete, freddo, ...).

**Message Digest algorithm 5 (MD5):** MD5 è un algoritmo crittografico di hashing che trasforma la stringa in input in un'altra lunga 128bit (MD5 Checksum). Le proprietà fondamentali di questo algoritmo e in generale di tutti gli algoritmi di hashing sono:



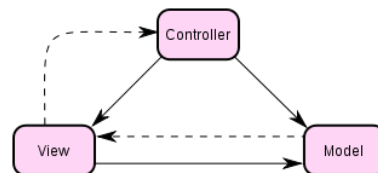
- bassissima probabilità che una stessa stringa in input generi la stessa checksum;
- velocità della conversione in andata;
- estrema difficoltà della conversione in ritorno (2 alla 128esima possibili soluzioni).

**Memory leak:** In informatica, un memory leak (“falla nella memoria”) è un particolare tipo di consumo non voluto di memoria dovuto alla mancata deallocazione dalla memoria di variabili/dati non più utilizzati da parte dei processi.

**Model View Controller (MVC) (Design Pattern):** Model-View-Controller è un pattern architetturale molto diffuso nello sviluppo di interfacce grafiche di sistemi software object-oriented. Il pattern è basato sulla separazione dei compiti fra i componenti software che interpretano tre ruoli principali:

- il model fornisce i metodi per accedere ai dati utili all’applicazione;
- il view visualizza i dati contenuti nel model e si occupa dell’interazione con utenti e agenti;
- il controller riceve i comandi dell’utente (in genere attraverso il view) e li attua modificando lo stato degli altri due componenti.

Questo schema, fra l’altro, implica anche la tradizionale separazione fra la logica applicativa (in questo contesto spesso chiamata logica di business), a carico del controller e del model, e l’interfaccia utente a carico del view.



*Figura 6.1: Immagine che descrive lo schema di interazioni tra le componenti del pattern architetturale MVC.*

**Serious game:** Si tratta di un vero e proprio gioco che non ha come finalità il semplice intrattenimento, il suo obiettivo è quello di sviluppare ed aumentare alcune abilità di chi lo utilizza. Questo tipo di programmi trova ora applicazione in moltissimi contesti.

Questo termine venne usato per la prima volta nel 1970 in un libro dal titolo “Serious Games” in cui, precedentemente all’inizio dell’era dei computers, ne venivano esposti i principi fondamentali.

**Singleton (Design Pattern):** Il Singleton è un design pattern creazionale che ha lo scopo di garantire che di una determinata classe venga creata una e una sola istanza, e di fornire un punto di accesso globale a tale istanza.

**Software Development Kit (SDK):** Il Software Development Kit è un insieme di strumenti per lo sviluppo e la documentazione di software. Le componenti fondamentali che compongono un SDK sono:

- un compilatore, per tradurre il codice sorgente in un eseguibile;
- librerie standard dotate di interfacce pubbliche dette API;

Oltre a questi elementi di base gli SDK contengono molti altri strumenti e documenti utili allo sviluppo ed alla gestione di applicazioni.

**SQLite:** Questo sistema di gestione di database SQL è caratterizzato dal fatto di poter leggere e scrivere su semplici file, a differenza di tutti gli altri che necessitano invece di un processo separato (server). È un database relazionale che permette la creazione di tabelle, triggers, indici, viste, tutto in un file ordinario. Il suo utilizzo è completamente gratuito poiché è stato affidato al dominio pubblico dagli autori.

**Unified Modeling Language (UML):** Unified Modeling Language è un linguaggio di modellazione e specifica basato sul paradigma object-oriented, utilizzato per descrivere soluzioni analitiche e progettuali in modo sintetico e comprensibile.

**Universal design:** Universal Design, in italiano Progettazione Universale, è il termine internazionale con cui ci si riferisce ad una metodologia progettuale di moderna concezione e ad ampio spettro che ha per obiettivo fondamentale la progettazione e la realizzazione di edifici, prodotti e ambienti che sono di per sé accessibili ad entrambe le categorie di persone sia che siano esse senza disabilità che con disabilità.

**Web Content Accessibility Guidelines (WCAG):** Le WCAG fanno parte delle iniziative intraprese dal WAI (Web Accessibility Initiative), che a sua volta fa parte del W3C (World Wide Web Consortium), con lo scopo di creare delle linee guida che gli sviluppatori (del Web) di contenuti, di strumenti di sviluppo e di strumenti di valutazione dell'accessibilità possono seguire, per poter creare e valutare che dei contenuti Web siano accessibili anche a persone disabili.

Queste indicazioni, la cui ultima versione risale al 2008 (WCAG2.0), sono tecniche applicabili a qualsiasi tecnologia web, abbastanza esplicite e semplici da capire.



## Riferimenti Bibliografici

- [1] Portale internazionale per l'informatica pediatrica  
<http://www.pedinfo.org/>  
(Citato a pag. 3)
- [2] Sito del Dipartimento di Pediatria Salus Pueri dell'Università di Padova  
- La terapia intensiva pediatrica  
<http://www.pediatria.unipd.it/on-line/Home/AssistenzaeCura/CentriSpecialistici/scheda6002854.html>  
(Citato a pag. 3)
- [3] Sito del Dipartimento di Pediatria Salus Pueri dell'Università di Padova  
- Scheda di Cristina Zaggia  
<http://www.pediatria.unipd.it/on-line/Home/Dipartimento/Staff/CoordinatoriInfermieristici/scheda6003388.html>  
(Citato a pag. 6)
- [4] Comunicazione con il bambino sottoposto a ventilazione assistita in terapia intensiva pediatrica: strumenti di comunicazione non verbali (Caoduro Federica, tesi di laurea in Infermieristica presso l'Università degli studi di Padova, A.A. 2008-2009)  
<http://www.slideshare.net/rmancin/il-bambino-afono>  
(Citato a pag. 6)
- [5] Un'applicazione per la predizione testuale, progettata per persone con disabilità neuromotoria (Mancin Roberto, tesi di laurea in Scienze dell'Informazione presso l'Università degli studi di Venezia, A.A. 1996-1997)  
<http://tesi.cab.unipd.it/9403/>  
(Citato a pag. 7)

- [6] Corriere della Sera - Dopo il Web 2.0, il “Web al quadrato”  
(Nicola Bruno, 25/09/2010)  
[http://www.corriere.it/scienze\\_e\\_tecnologie/09\\_settembre\\_25/  
web-quadrato\\_c61499b0-a9bf-11de-93d1-00144f02aabc.shtml](http://www.corriere.it/scienze_e_tecnologie/09_settembre_25/web-quadrato_c61499b0-a9bf-11de-93d1-00144f02aabc.shtml)  
(Citato a pag. 7)
- [7] Electronic voice-output communication aids for temporarily nonspeaking patients in a medical intensive care unit: a feasibility study. (Happ MB, Roesch TK, Garrett K, School of Nursing, University of Pittsburgh)  
(Citato a pag. 12)
- [8] Pediatric Informatics (Lehmann, Christoph; Kim, George R.; Johnson, Kevin B. (Eds.))
- [9] Documentazione ufficiale per sviluppatori Android  
<http://developer.android.com/>

# A Appendice - VOCA.TIO: resoconto del test di venerdì 2/9

Segue il resoconto del primo test sul campo effettuato con Mind Reader Squared (MRS), presso il reparto di terapia intensiva pediatrica dell'ospedale di Padova, a cui purtroppo non ho potuto partecipare.

*Non c'erano presso la Terapia Intensiva Pediatrica di Padova bambini vigili ma temporaneamente incapaci di comunicare verbalmente o graficamente. Comunque il dott. Divisic Antuan ha chiesto a un bambino di 9 anni di nome J., da qualche giorno non più intubato, di provare MRS. Come input abbiamo usato un mouse. Il test è andato benissimo dato che J. non ha disturbi cognitivi e visivi ma solo motori. 2 volte però ha premuto il tasto destro del mouse invece del sinistro per selezionare un'immagine e questo ha causato l'uscita da MRS. Problema tecnico facilmente risolvibile ma indicativo del fatto che è necessario lavorare non più sul software che è perfetto per J e presumibilmente per il 99% degli ospiti della Tiped ma sull'input affinché sia sempre possibile catturare la volontà del giovane "giocatore" espressa tramite le abilità residue ed emergenti al risveglio in tiped da coma/sedazione. Questo supera le richieste iniziali ma sarà certamente oggetto di sviluppi futuri.*

*Si segnala che il test è durato quasi un'ora. J. non era per nulla stanco, anzi avrebbe voluto usare il sistema per "giocare" ancora! Anche la curiosità, l'interesse e la disponibilità dello staff della Tiped mi ha piacevolmente stupito tanto che un'infermiere di turno ha messo a disposizione il suo volto per personalizzare, tramite una fotografia MRS. Questo ha reso il test più "caldo" e divertente.*



*Importanti le osservazioni a caldo di chi era presente che porteranno ad ulteriori perfezionamenti.*