

Progettazione Applicazione Tirocinio

Lucia Fores

Indice

1	Descrizione del Sistema	4
1.1	Introduzione	4
1.2	Descrizione generale dell'applicazione	4
1.3	Requisiti	5
2	Studio di Fattibilità	7
2.1	Background e Formulazione del Problema	7
2.2	Soluzione al Problema	7
2.3	Vantaggi	7
2.4	Svantaggi	8
2.5	Tecnologie	8
2.6	Distribuzione	8
3	Glossario	10
4	Funzionalità del Sistema	12
4.1	Specifica dei requisiti	12
4.2	Modello dei Casi d'Uso	14
4.2.1	Attori	14
4.2.2	Diagramma UML dei Casi d'Uso	16
4.2.3	Casi d'Uso	17
4.2.3.1	Specifica Use-Case: Gestione Accesso	17
4.2.3.2	Specifica Use-Case: Gestione Profilo	22
4.2.3.3	Specifica Use-Case: Logout	23
4.2.3.4	Specifica Use-Case: Gestione Schede Pazienti	24
4.2.3.5	Specifica Use-Case: Gestione Piani Terapeutici	28
4.2.3.6	Specifica Use-Case: Gestione Tipologie e Abilità Esercizi	31
4.2.3.7	Specifica Use-Case: Gestione Implementazioni Esercizi	33
4.2.3.8	Specifica Use-Case: Gestione Informazioni Esercizi	34
4.2.3.9	Specifica Use-Case: EsecuzioneEsercizio	35
4.2.3.10	Specifica Use-Case: ConsultazioneDati	36
4.2.3.11	Specifica Use-Case: ConnessionePazienti	38
5	Specifiche Aggiuntive	42
6	Progettazione della Base Dati	45
6.1	Diagramma ER	45
6.2	Specifiche dei dati	46
6.3	Ristrutturazione del Diagramma ER e delle Specifiche dei Dati	57
6.3.1	Diagramma ER Ristrutturato	57
6.3.2	Specifiche dei Dati Ristrutturate	58

6.4	Tabelle della Base Dati	70
6.4.1	Ulteriori Vincoli Esterni	75

1 Descrizione del Sistema

1.1 Introduzione

Si vuole creare un'applicazione di supporto a caregiver di bambini con problematiche psico-motorie.

L'applicazione intende affermarsi come uno strumento di supporto al percorso di terapia che bambini affetti da tali problematiche seguono per recuperare i deficit che hanno.

1.2 Descrizione generale dell'applicazione

Il sistema sviluppato permetterà di svolgere le seguenti attività:

- permetterà ai **caregiver** del bambino e ai **terapisti** di:
 - **isciversi** nel sistema;
 - **preparare** particolari esecuzioni di un esercizio per i bambini che seguono;
 - **avere accesso** a tutti gli esercizi che sono stati implementati per lo specifico bambino che viene seguito sia per consultazione sia per permetterne l'esecuzione da parte del bambino;
 - **indicare** per ogni esercizio eseguito la **data di esecuzione** di tale esercizio.
- permetterà ai **terapisti** di:
 - **inserire** nuovi profili di bambini tenuti in cura;
 - **cancellare** profili di bambini non più tenuti in cura;
 - **creare** piani terapeutici per i bambini indicando quali esercizi vengono inseriti nel piano;
 - **modificare** piani terapeutici già esistenti;
 - **cancellare** piani terapeutici già esistenti;
 - **indicare** per ogni esercizio eseguito la **data di inserimento nel piano**, una tra la **data di apprendimento** o la **data di sospensione** dell'esercizio e, qualora l'esercizio sia stato appreso, un'eventuale **data di ripasso** di questo.
- permetterà ai **gestori di sistema** di:
 - **inserire** nuove tipologie di esercizio su richiesta dei terapeuti (le nuove tipologie verranno inserite insieme alla loro versione eseguibile nel sistema così da poter essere eseguiti dal bambino)
 - **inserire** nuove abilità su cui basare un singolo esercizio per un bambino

- permetterà ai **bambini** sotto supervisione dei **caregiver** di:
 - **eseguire** gli esercizi presenti nel piano terapeutico a loro assegnato

1.3 Requisiti

Gli utenti che potranno interagire con il sistema sono: **Utente non autenticato**, **Terapista**, **Caregiver non specializzato**, **Bambino** e **Gestore di Sistema**.

Il sistema garantisce le seguenti funzionalità per **Utente non autenticato**:

- Accesso al modulo di log-in;
- Accesso al modulo di recupero delle credenziali di accesso;
- Accesso al modulo di registrazione per iscriversi al sistema come Terapista o Caregiver.

Il sistema, in dettaglio, offre le seguenti funzionalità per un **Terapista**:

- Fare log-in e log-out;
- Creazione di profili per bambini che tiene in cura;
- Modifica di profili di bambini che tiene in cura;
- Cancellazione di profili di bambini che tiene in cura;
- Creazione di piani terapeutici per bambini che tiene in cura;
- Modifica dei piani terapeutici già creati;
- Cancellazione dei piani terapeutici già creati;
- Inserimento di nuove tipologie di esercizi;
- Creazione di nuove implementazioni di tipologie di esercizi presenti nel sistema;
- Aggiornamento delle informazioni sugli esercizi presenti nei piani terapeutici già creati;
- Accesso alla sezione di esecuzione degli esercizi con il bambino;
- Consultazione dei dati presenti in piattaforma, in particolare:
 - Consultazione di tutte le tipologie di esercizi che sono state inserite da tutti gli utenti del sistema;
 - Consultazione di tutti i dati inerenti ai bambini che tiene in cura (ossia i dati anagrafici, medici e del piano terapeutico assegnato).

Offre le seguenti funzionalità per un **Caregiver non specializzato**:

- Fare log-in e log-out;
- Accesso ai piani terapeutici dei bambini che segue;
- Creazione di nuove implementazioni per gli esercizi presenti nel piano terapeutico;
- Accesso alla sezione di esecuzione degli esercizi con il bambino;
- Parziale aggiornamento delle informazioni sugli esercizi presenti nei piani terapeutici già creati.

Offre le seguenti funzionalità per un **Bambino**:

- Accesso alla sezione di esecuzione degli esercizi.

Infine offre le seguenti funzionalità per un **Gestore di Sistema**:

- Inserimento di una nuova tipologia di esercizio con la correlata versione eseguibile
- Inserimento di una nuova abilità per gli esercizi nella lista delle abilità disponibili

2 Studio di Fattibilità

2.1 Background e Formulazione del Problema

Attualmente le famiglie con componenti che riportano ritardi di qualsiasi tipo sono spesso lasciate a loro stesse: esistono strutture e specialisti che si occupano della gestione di bambini con tali problematiche ma questo aiuto difficilmente riesce ad essere riportato costantemente in contesto familiare e scolastico.

Queste difficoltà sono state messe ancora più in luce durante il periodo della pandemia da Covid-19 in quanto, soprattutto durante i periodi di lockdown, tutte le attività di supporto a bambini con tali disabilità sono state interrotte lasciando le famiglie da sole a gestire situazioni che fino a quel momento non avevano mai vissuto con i propri figli.

2.2 Soluzione al Problema

Il sistema che si vuole sviluppare cerca di proporsi come strumento di supporto generale al percorso di terapia che viene svolto dai soggetti con disabilità psicomotorie (il sistema tuttavia vuole essere costruito in modo tale da essere scalabile ed adattabile a tutte le altre disabilità) e soprattutto come strumento di continua assistenza alle famiglie e agli educatori di bambini con tali problematiche, in modo tale da permettere un proseguimento del percorso di terapia iniziato con gli specialisti anche nei momenti in cui questi non possono essere fisicamente presenti con il bambino.

Il sistema non si vuole imporre come uno strumento a sostituzione della reale terapia, che deve comunque rimanere l'attività principale che il bambino deve svolgere per il recupero del deficit, ma vuole essere sviluppato in quanto si crede che possa fornire sia un supporto valido al percorso terapeutico del bambino anche in quei casi in cui il contesto più stringente della seduta terapeutica non permette al paziente di esercitarsi con la serenità che invece riesce ad avere in altri contesti sia un supporto valido in ambiente scolastico e familiare così da poter proseguire costantemente il lavoro di terapia specialistica.

2.3 Vantaggi

L'utilizzo del sistema permetterebbe da una parte uno snellimento del materiale che i terapisti producono per ogni paziente che seguono, andando a sostituire tutti supporti cartacei utilizzati per il mantenimento delle informazioni sul paziente e sul percorso terapeutico che sta seguendo, dall'altra permetterebbe un mantenimento e una comunicazione più agile del percorso terapeutico pensato per il bambino, in quanto gli esercizi potrebbero essere inseriti e inviati al paziente in qualsiasi momento, e soprattutto l'utilizzo del sistema permetterebbe una fruizione continua del percorso terapeutico da parte del paziente anche in situazioni che non permettono l'incontro diretto con i terapisti o in contesti complementari vissuti dal bambino come quello familiare e scolastico.

Si pensa che l'uso del sistema permetta inoltre di integrare l'attività terapeutica

anche in contesti che permettono al bambino di svolgere con più serenità gli esercizi assegnati e soprattutto che il sistema permetta un maggior coinvolgimento dei caregiver del bambino nel suo percorso di recupero dei deficit: il sistema sviluppato infatti permetterebbe ai caregiver di essere sempre aggiornati sulle attività che i bambini svolgono e permetterebbe anche ai caregiver di sapere l'obiettivo che si vuole raggiungere attraverso l'uso di ogni esercizio proposto al paziente nel percorso terapeutico.

Infine l'uso dell'applicazione in contesto scolastico può essere d'aiuto nei casi in cui il sostegno fornito al bambino non sia dato da figure specificatamente formate per trattare con pazienti affetti da particolari disabilità.

2.4 Svantaggi

Non tutti gli esercizi di terapia si prestano ad essere digitalizzati, basti pensare a tutti quelli che presuppongono l'utilizzo di oggetti fisici come punto cardine di svolgimento, inoltre l'esecuzione errata degli esercizi del piano terapeutico senza la supervisione di specialisti potrebbe non permettere una correzione tempestiva degli errori commessi.

2.5 Tecnologie

Il sistema richiede:

- un'interfaccia grafica per l'interazione con gli utenti;
- un editor per l'inserimento di nuove implementazioni di esercizi;
- un software per l'esecuzione degli esercizi da parte dei bambini in contesti diversi da quello della terapia;
- un server per gestire le richieste degli utenti;
- un database relazionale per mantenere e gestire i dati dei terapeuti, dei caregiver, dei pazienti e degli esercizi.

2.6 Distribuzione

L'applicazione è stata pensata per essere distribuita come applicazione web.

I vantaggi nella distribuzione come sito web consistono nel fatto di creare un'applicazione fruibile da qualunque dispositivo (sviluppando un sito *responsive* infatti si permette la corretta visualizzazione sia tramite computer sia tramite cellulari e tablet) in qualunque momento e situazione (a differenza di un'applicazione installabile infatti si elimina il rischio di non poterne usufruire per mancata memoria sul dispositivo).

Inoltre si è pensato ad un sito web soprattutto per la fruizione dell'applicazione in ambito scolastico: per l'esecuzione degli esercizi è infatti necessario permettere al bambino di utilizzare il *device* su cui si sta eseguendo l'applicazione, distribuendola come sito web è possibile quindi la fruizione da parte dei

bambini (che potrebbero non possedere un *personal device*) attraverso l'utilizzo dei *device* messi a disposizione dalle scuole nei laboratori senza che i caregiver cedano al bambino il proprio *personal device*.

3 Glossario

Vengono ora elencati i termini significativi del dominio e la loro descrizione.

Termine	Definizione	Note
Abilità	Capacità che si vuole insegnare ad un bambino	
Apprendimento	Attestazione dell'avvenuto apprendimento di un'abilità da parte del bambino	
Bambino	Individuo che riporta deficit psico-motori	Un suo sinonimo è <i>paziente</i>
Caregiver	Tutte le persone che si prendono cura di un bambino con problematiche psico-motorie; nella sua accezione più estesa indica sia i caregiver specializzati, ossia i terapeuti, sia i caregiver non specializzati, ossia i membri del nucleo familiare, gli educatori scolastici, ...	
Caregiver non specializzato	Figura che si prende cura dei bambini ma che non può essere considerata specialista, rientrano in questa categoria, ad esempio, i membri del nucleo familiare e gli educatori scolastici	Un suo sinonimo è <i>Caregiver non professionale</i>
Esercizio	Indica un'attività svolta nell'ottica di recupero del deficit del bambino	Un suo sinonimo è <i>attività</i>
Implementazione dell'esercizio	Particolare caratterizzazione di un esercizio pensata appositamente per l'insegnamento di un'abilità specifica per il bambino	Un suo sinonimo è <i>esecuzione di esercizio</i>

Termine	Definizione	Note
Problematica psico-motoria	Deficit nell'acquisizione delle competenze motorie, cognitive e comunicative in relazione all'età cronologica dell'individuo. Indica uno scollamento tra lo sviluppo e quelli che sono considerati gli standard specifici per l'età dell'individuo	Suoi sinonimi sono <i>deficit psico-motorio</i> , <i>disabilità psico-motoria</i> , <i>ritardo psico-motorio</i>
Percorso di terapia	Percorso di terapia che segue il bambino per il recupero dei deficit psico-motori	Suoi sinonimi sono <i>percorso terapeutico</i> , <i>piano terapeutico</i> , <i>programma</i>
Ripasso	Esecuzione dell'esercizio o con le stesse modalità utilizzate in fase di apprendimento o con modalità diverse	
Routine	Insieme di esercizi da far svolgere ad un paziente in una seduta di esecuzione di esercizi	
Sospensione	Attestazione del non corretto funzionamento dell'esercizio per il bambino	
Strumento di supporto	Strumento per supportare la terapia per il recupero di un deficit psico-motorio	
Terapista	Figura specialista che segue il bambino nel suo percorso di terapia	Suoi sinonimi sono <i>caregiver specializzato</i> , <i>caregiver professionale</i>
Utente non autenticato	Utente che non ha ancora eseguito l'accesso al sistema	

4 Funzionalità del Sistema

In questa sezione vengono descritte le funzionalità previste per il sistema prima attraverso una loro descrizione e in seguito attraverso i modelli dei casi d'uso.

4.1 Specifica dei requisiti

ID	REQ_1_Effettua_Registrazione
Descrizione	Il Sistema dovrà permettere ad un nuovo utente di registrarsi o come Terpista o come Caregiver Non Specializzato
ID	REQ_2_Recupero_DatiDiAccesso
Descrizione	Il Sistema dovrà permettere ad un utente di recuperare i dati necessari all'autenticazione
ID	REQ_3_Effettua_Autenticazione
Descrizione	Il Sistema dovrà permettere ad un utente iscritto di autenticarsi, facendo distinzione tra Terapisti e Caregiver Non Specializzati
ID	REQ_4_Modifica_DatiDiAccesso
Descrizione	Il Sistema dovrà permettere ad un utente autenticato di modificare alcuni suoi dati necessari all'autenticazione
ID	REQ_5_Effettua_Logout
Descrizione	Il Sistema dovrà permettere ad un utente autenticato di eseguire il logout
ID	REQ_6_Inserisci_Nuova_SchedaPaziente
Descrizione	Il Sistema dovrà permettere ad un terapeuta di inserire la scheda di un nuovo paziente
ID	REQ_7_Modifica_SchedaPaziente
Descrizione	Il Sistema dovrà permettere ad un terapeuta di modificare alcuni dati della scheda di un paziente
ID	REQ_8_Elimina_SchedaPaziente
Descrizione	Il Sistema dovrà permettere ad un terapeuta di eliminare una scheda paziente inserita
ID	REQ_9_Inserisci_Nuovo_Programma
Descrizione	Il Sistema dovrà permettere ad un terapeuta di inserire un nuovo piano terapeutico per un bambino che tiene in cura

ID	REQ_10_Modifica_Programma
Descrizione	Il Sistema dovrà permettere ad un terapeuta di modificare il piano terapeutico per un bambino che tiene in cura
ID	REQ_11_Elimina_Programma
Descrizione	Il Sistema dovrà permettere ad un terapeuta di eliminare il piano terapeutico per un bambino che tiene in cura
ID	REQ_12_Inserisci_Tipologia_Esercizio
Descrizione	Il Sistema dovrà permettere ad un gestore di sistema di inserire una nuova tipologia di esercizio nel sistema
ID	REQ_13_Inserisci_Abilita
Descrizione	Il Sistema dovrà permettere ad un gestore di sistema di inserire una nuova abilità nel sistema
ID	REQ_14_Crea_Implementazione_Esercizio
Descrizione	Il Sistema dovrà permettere ad un caregiver di inserire un'implementazione per un esercizio che fa parte del piano terapeutico del bambino
ID	REQ_15_Aggiorna_Informazioni_EsecuzioneEsercizio
Descrizione	Il Sistema dovrà permettere ad un caregiver di aggiornare le informazioni riguardanti l'esecuzione di un esercizio da parte di un bambino, facendo distinzione tra terapisti e caregiver non specializzati
ID	REQ_16_Esecuzione_Esercizio
Descrizione	Il Sistema dovrà permettere ad un caregiver di accedere alla sezione di esecuzione degli esercizi e dovrà permettere al bambino di eseguirli
ID	REQ_17_Consultazione_Dati
Descrizione	Il Sistema dovrà permettere ad un terapeuta di consultare i dati presenti al suo interno; in particolare, dovrà permettere di consultare tutte le tipologie di esercizi presenti nel sistema e di consultare tutti i dati inerenti i bambini che tiene in cura (ossia i dati anagrafici, medici e del piano terapeutico, intendendo per quest'ultimo sia i dati del piano stesso sia i dati di esecuzione degli esercizi presenti al suo interno)

4.2 Modello dei Casi d'Uso

4.2.1 Attori

Vengono qui di seguito riportati gli attori che prendono parte ai casi d'uso del sistema che si sta progettando.

ID	A.01
Nome	UtenteNonAutenticato
Genitore	/
Tipo	Primario
Stereotipo	Umano
Descrizione	Rappresenta un generico visitatore dell'applicazione probabilmente prossimo all'autenticazione o alla registrazione

ID	A.02
Nome	UtenteAutenticato
Genitore	/
Tipo	Primario
Stereotipo	Umano
Descrizione	Rappresenta un generico visitatore dell'applicazione che ha terminato le operazioni di autenticazione

ID	A.02.01
Nome	Terapista
Genitore	A.02 UtenteAutenticato
Tipo	Primario
Stereotipo	Umano
Descrizione	Rappresenta un utente di tipo Terapista, gestore delle schede dei pazienti e delle tipologie di esercizi

ID	A.02.02
Nome	CaregiverNonSpecializzato
Genitore	A.02 UtenteAutenticato
Tipo	Primario
Stereotipo	Umano
Descrizione	Rappresenta un utente di tipo CaregiverNonSpecializzato, persona che si occupa del bambino

ID	A.03
Nome	Bambino
Genitore	/
Tipo	Primario
Stereotipo	Umano
Descrizione	Rappresenta un utente di tipo Bambino, ossia un utente che ha accesso alla sola esecuzione degli esercizi

ID	A.04
Nome	Gestore di Sistema
Genitore	/
Tipo	Primario
Stereotipo	Umano
Descrizione	Rappresenta un gestore del sistema

4.2.2 Diagramma UML dei Casi d'Uso

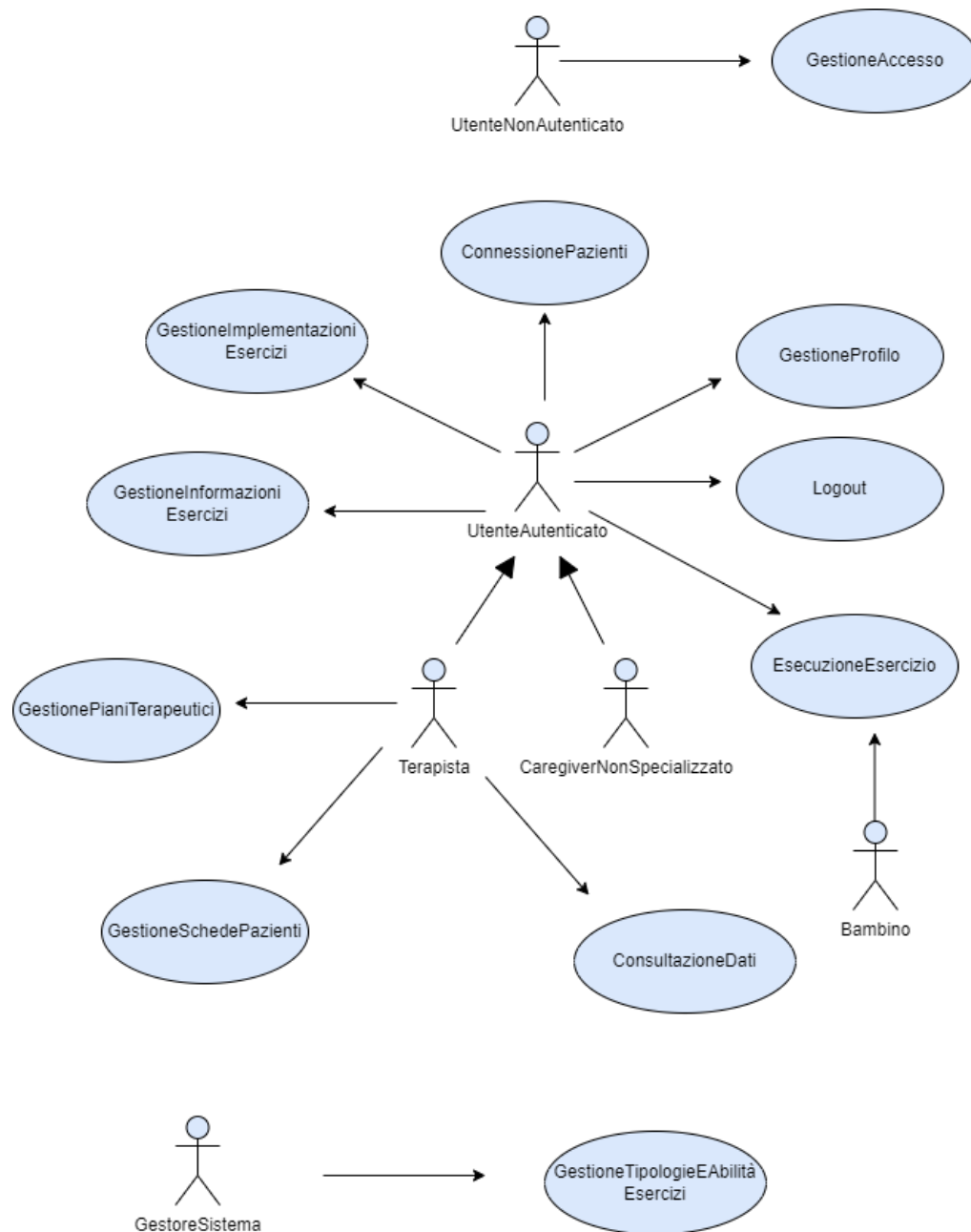


Figura 1: Diagramma UML degli Use Case del sistema

4.2.3 Casi d'Uso

4.2.3.1 Specifica Use-Case: Gestione Accesso

ID Use Case	UC.1.1
Nome	EffettuaRegistrazioneCaregiver
Attori	UtenteNonAutenticato (rinominato <i>Utente</i> in questo caso d'uso)
Descrizione	Registrazione di un nuovo utente di tipo Caregiver
Pre-condizioni	Il Sistema non contiene una sessione attiva per l'Utente
Flusso Principale	<ol style="list-style-type: none">1. L'Utente chiede al Sistema di dare avvio alla procedura di registrazione;2. Il Sistema chiede di inserire i dati relativi alla registrazione (<i>Specifiche Aggiuntive SPEC.01</i>);3. L'Utente compila e conferma i dati inseriti;4. Il Sistema verifica che tutti i campi obbligatori siano stati compilati e controlla che i dati inseriti siano validi (<i>Specifiche Aggiuntive SPEC.09</i>);5. Il Sistema controlla che non ci sia un altro utente registrato con lo stesso indirizzo di posta elettronica;6. Il Sistema invia all'indirizzo di posta elettronica indicata una notifica di registrazione effettuata correttamente, nella quale è presente il codice identificativo univoco assegnato all'Utente appena iscritto.
Post-condizioni	Il Sistema contiene un nuovo Caregiver
Flusso Alternativo #1	<p>*.a. In qualsiasi momento l'Utente interrompe l'operazione:</p> <ol style="list-style-type: none">1. Il Sistema scarta eventuali dati inseriti e reindirizza l'Utente alla pagina da cui ha avviato la procedura.
Flusso Alternativo #2	<p>4.a. Il Sistema rileva che <i>non</i> tutti i campi obbligatori sono stati compilati o che i dati inseriti <i>non</i> sono validi:</p> <ol style="list-style-type: none">1. Il Sistema segnala all'Utente l'errore e chiede di inserire nuovamente i dati mancanti o non validi;2. Il flusso riprende dal punto 3. del flusso principale.

Flusso Alternativo #3	<p>5.a. Il Sistema rileva che esiste già un altro utente con lo stesso indirizzo di posta elettronica:</p> <ol style="list-style-type: none"> 1. Il Sistema segnala all'Utente l'errore e lo reindirizza alla pagina da cui ha avviato la procedura.
------------------------------	---

ID Use Case	UC.1.2.
Nome	RecuperaIDUnivoco
Attori	UtenteNonAutenticato (rinominato <i>Utente</i> in questo caso d'uso)
Descrizione	Recupero dell'ID univoco necessario all'autenticazione
Pre-condizioni	Il Sistema non contiene una sessione attiva per l'Utente
Flusso Principale	<ol style="list-style-type: none"> 1. L'Utente chiede al Sistema di dare avvio alla procedura di recupero dei dati dimenticati; 2. Il Sistema chiede di inserire l'indirizzo di posta elettronica specificato in fase di registrazione; 3. L'Utente compila e conferma i dati inseriti; 4. Il Sistema verifica che tutti i campi obbligatori siano stati compilati; 5. Il Sistema controlla ci sia un utente registrato con l'indirizzo di posta elettronica specificato; 6. Il Sistema invia all'indirizzo di posta elettronica indicata una notifica di recupero delle credenziali effettuata correttamente, nella quale è presente il codice identificativo univoco assegnato all'Utente che lo ha richiesto.
Post-condizioni	Nessuna
Flusso Alternativo #1	<p>*.a. In qualsiasi momento l'Utente interrompe l'operazione:</p> <ol style="list-style-type: none"> 1. Il Sistema scarta eventuali dati inseriti e reindirizza l'Utente alla pagina da cui ha avviato la procedura
Flusso Alternativo #2	<p>4.a. Il Sistema rileva che <i>non</i> tutti i campi obbligatori sono stati compilati:</p> <ol style="list-style-type: none"> 1. Il Sistema segnala all'Utente l'errore e chiede di inserire nuovamente i dati mancanti; 2. Il flusso riprende dal punto 3. del flusso principale.

Flusso Alternativo #3	<p>5.a. Il Sistema rileva che non esiste un utente registrato con l'indirizzo di posta elettronica specificato:</p> <ol style="list-style-type: none"> 1. Il Sistema segnala all'Utente l'errore e lo reindirizza alla pagina da cui ha avviato la procedura.
------------------------------	--

ID Use Case	UC.1.3.
Nome	RecuperaPassword
Attori	UtenteNonAutenticato (rinominato <i>Utente</i> in questo caso d'uso)
Descrizione	Recupero della password necessaria all'autenticazione
Pre-condizioni	Il Sistema non contiene una sessione attiva per l'Utente
Flusso Principale	<ol style="list-style-type: none"> 1. L'Utente chiede al Sistema di dare avvio alla procedura di recupero dei dati dimenticati; 2. Il Sistema chiede di inserire l'indirizzo di posta elettronica specificato in fase di registrazione o l'ID univoco collegato all'Utente; 3. L'Utente compila e conferma i dati inseriti; 4. Il Sistema verifica che tutti i campi obbligatori siano stati compilati; 5. Il Sistema controlla che ci sia un utente registrato con l'indirizzo di posta elettronica specificato o con l'ID specificato; 6. Il Sistema invia all'indirizzo di posta elettronica indicata una notifica di recupero delle credenziali effettuata correttamente, nella quale è presente la password dell'Utente che l'ha richiesta.
Post-condizioni	Nessuna
Flusso Alternativo #1	<p>*.a. In qualsiasi momento l'Utente interrompe l'operazione:</p> <ol style="list-style-type: none"> 1. Il Sistema scarta eventuali dati inseriti e reindirizza l'Utente alla pagina da cui ha avviato la procedura.
Flusso Alternativo #2	<p>4.a. Il Sistema rileva che <i>non</i> tutti i campi obbligatori sono stati compilati:</p> <ol style="list-style-type: none"> 1. Il Sistema segnala all'Utente l'errore e chiede di inserire nuovamente i dati mancanti; 2. Il flusso riprende dal punto 3. del flusso principale.
Flusso Alternativo #3	<p>5.a. Il Sistema rileva che non esiste un utente registrato con l'indirizzo di posta elettronica specificato o con l'ID specificato:</p> <ol style="list-style-type: none"> 1. Il Sistema segnala all'Utente l'errore e lo reindirizza alla pagina da cui ha avviato la procedura

ID Use Case	UC.1.4.
Nome	EffettuaAutenticazione
Attori	UtenteNonAutenticato (rinominato <i>Utente</i> in questo caso d'uso)
Descrizione	L'Utente si connette al Sistema
Pre-condizioni	Il Sistema non contiene una sessione attiva per l'Utente
Flusso Principale	<ol style="list-style-type: none"> 1. L'Utente chiede al Sistema di dare avvio alla procedura di connessione; 2. Il Sistema chiede all'Utente i dati relativi all'autenticazione (<i>Specifiche Aggiuntive SPEC.10</i>); 3. L'Utente compila e conferma i dati inseriti; 4. Il Sistema verifica che tutti i campi obbligatori siano stati compilati e controlla che i dati inseriti siano validi (<i>Specifiche Aggiuntive SPEC.09</i>); 5. Il Sistema controlla che esista un profilo associato ai dati inseriti dall'Utente in fase di connessione; 6. Il Sistema connette l'Utente e gli segnala che la procedura è andata a buon fine.
Post-condizioni	Il Sistema contiene una sessione attiva per l'Utente
Flusso Alternativo #1	<p>*.a. In qualsiasi momento l'Utente interrompe l'operazione:</p> <ol style="list-style-type: none"> 1. Il Sistema scarta eventuali dati inseriti e reindirizza l'Utente alla pagina da cui ha avviato la procedura.
Flusso Alternativo #2	<p>4.a. Il Sistema rileva che <i>non</i> tutti i campi obbligatori sono stati compilati o che i dati inseriti <i>non</i> sono validi:</p> <ol style="list-style-type: none"> 1. Il Sistema segnala all'Utente l'errore e chiede di inserire nuovamente i dati mancanti o non validi; 2. Il flusso riprende dal punto 3. del flusso principale.
Flusso Alternativo #3	<p>4.a. Il Sistema rileva che <i>non</i> esiste un profilo associato ai dati inseriti dall'Utente:</p> <ol style="list-style-type: none"> 1. Il Sistema segnala all'Utente l'errore e lo reindirizza alla pagina da cui ha avviato la procedura.

4.2.3.2 Specifica Use-Case: Gestione Profilo

ID Use Case	UC.2.1.
Nome	ModificaPassword
Attori	UtenteAutenticato (rinominato <i>Utente</i> in questo caso d'uso)
Descrizione	Modifica della password necessaria all'autenticazione
Pre-condizioni	Il Sistema contiene una sessione attiva per l'Utente
Flusso Principale	<ol style="list-style-type: none"> 1. L'Utente chiede al Sistema di dare avvio alla procedura di modifica della password; 2. Il Sistema chiede di inserire la password corrente dell'Utente e la nuova password dell'Utente; 3. L'Utente compila e conferma i dati inseriti; 4. Il Sistema verifica che tutti i campi obbligatori siano stati compilati e controlla che i dati inseriti siano validi (<i>Specifiche Aggiuntive SPEC.09</i>) 5. Il Sistema controlla che la vecchia password corrisponda a quella dell'Utente; 6. Il Sistema segnala all'Utente che la procedura è avvenuta correttamente.
Post-condizioni	Il Sistema contiene una nuova password per l'Utente (e la precedente viene eliminata)
Flusso Alternativo #1	<p>*.a. In qualsiasi momento l'Utente interrompe l'operazione:</p> <ol style="list-style-type: none"> 1. Il Sistema scarta eventuali dati inseriti e reindirizza l'Utente alla pagina da cui ha avviato la procedura.
Flusso Alternativo #2	<p>4.a. Il Sistema rileva che <i>non</i> tutti i campi obbligatori sono stati compilati o che i dati inseriti <i>non</i> sono validi:</p> <ol style="list-style-type: none"> 1. Il Sistema segnala all'Utente l'errore e chiede di inserire nuovamente i dati mancanti o non validi; 2. Il flusso riprende dal punto 3. del flusso principale.
Flusso Alternativo #3	<p>5.a. Il Sistema rileva che la vecchia password inserita non corrisponde a quella dell'Utente:</p> <ol style="list-style-type: none"> 1. Il Sistema segnala all'Utente l'errore all'Utente; 2. Il flusso riprende dal punto 2. del flusso principale.

4.2.3.3 Specifica Use-Case: Logout

ID Use Case	UC.3.1.
Nome	Logout
Attori	UtenteAutenticato (rinominato <i>Utente</i> in questo caso d'uso)
Descrizione	L'Utente si disconnette dal Sistema
Pre-condizioni	Il Sistema contiene una sessione attiva per l'Utente
Flusso Principale	<ol style="list-style-type: none">1. L'Utente chiede al Sistema di dare avvio alla procedura di disconnessione;2. Il Sistema chiede conferma all'Utente;3. L'Utente dà conferma positiva;4. Il Sistema disconnette l'Utente e gli comunica che la procedura è avvenuta con successo.
Post-condizioni	Il Sistema <i>non</i> contiene una sessione attiva per l'Utente
Flusso Alternativo #1	*.a. In qualsiasi momento l'Utente interrompe l'operazione: <ol style="list-style-type: none">1. Il Sistema mantiene la sessione attiva per l'Utente.
Flusso Alternativo #2	3.a. L'Utente dà conferma negativa: <ol style="list-style-type: none">1. Il Sistema mantiene la sessione attiva per l'Utente.

4.2.3.4 Specifica Use-Case: Gestione Schede Pazienti

ID Use Case	UC.4.1.
Nome	InserisciNuovaSchedaPaziente
Attori	Caregiver il cui ruolo è professionale (soprannominato <i>Terapista</i> in questo caso d'uso)
Descrizione	Il Terapista inserisce la scheda di un nuovo paziente nel sistema
Pre-condizioni	Non esiste una scheda per il paziente nel Sistema
Flusso Principale	<ol style="list-style-type: none"> 1. Il Terapista chiede al Sistema di dare avvio alla procedura di inserimento di una nuova scheda paziente; 2. Il Sistema chiede di inserire i dati relativi al paziente (<i>Specifiche Aggiuntive SPEC.02</i>); 3. L'Utente compila e conferma i dati inseriti; 4. Il Sistema verifica che tutti i campi obbligatori siano stati compilati; 5. Il Sistema controlla che non ci sia un'altra scheda paziente registrata con gli stessi dati; 6. Il Sistema segnala al Terapista che la procedura si è conclusa correttamente e mostra il codice identificativo univoco assegnato alla scheda paziente.
Post-condizioni	Il Sistema contiene una nuova Scheda Paziente
Flusso Alternativo #1	<p>*.a. In qualsiasi momento il Terapista interrompe l'operazione:</p> <ol style="list-style-type: none"> 1. Il Sistema scarta eventuali dati inseriti e reindirizza il Terapista alla pagina da cui ha avviato la procedura.
Flusso Alternativo #2	<p>4.a. Il Sistema rileva che <i>non</i> tutti i campi obbligatori sono stati compilati:</p> <ol style="list-style-type: none"> 1. Il Sistema segnala al Terapista l'errore e chiede di inserire nuovamente i dati mancanti; 2. Il flusso riprende dal punto 3. del flusso principale.
Flusso Alternativo #3	<p>5.a. Il Sistema rileva che esiste già una scheda per il paziente che si vuole inserire:</p> <ol style="list-style-type: none"> 1. Il Sistema segnala al Terapista l'errore.

ID Use Case	UC.4.2.
Nome	ModificaSchedaPaziente
Attori	Caregiver il cui ruolo è professionale (soprannominato <i>Terapista</i> in questo caso d'uso)
Descrizione	Il Terapista modifica la scheda di un paziente già presente nel sistema
Pre-condizioni	Esiste una scheda per il paziente nel Sistema
Flusso Principale	<ol style="list-style-type: none"> 1. Il Terapista chiede al Sistema di dare avvio alla procedura di modifica di una scheda paziente; 2. Il Sistema chiede di inserire il codice fiscale del paziente o il suo ID univoco; 3. Il Sistema chiede di inserire i dati relativi ai campi che devono essere modificati; 4. L'Utente compila e conferma i dati inseriti; 5. Il Sistema verifica che tutti i campi obbligatori siano ancora compilati (<i>Specifiche Aggiuntive SPEC.02</i>); 6. Il Sistema segnala al Terapista che la procedura si è conclusa correttamente.
Post-condizioni	La Scheda del paziente è stata aggiornata con i nuovi dati
Flusso Alternativo #1	<p>*.a. In qualsiasi momento il Terapista interrompe l'operazione:</p> <ol style="list-style-type: none"> 1. Il Sistema scarta eventuali dati inseriti e reindirizza il Terapista alla pagina da cui ha avviato la procedura.
Flusso Alternativo #2	<p>2.a. Il Sistema rileva che <i>non</i> esiste una scheda paziente per il codice fiscale inserito o per l'ID univoco inserito:</p> <ol style="list-style-type: none"> 1. Il Sistema segnala al Terapista l'errore e lo reindirizza alla pagina di inserimento delle schede pazienti.
Flusso Alternativo #3	<p>5.a. Il Sistema rileva che <i>non</i> tutti i campi:</p> <ol style="list-style-type: none"> 1. Il Sistema segnala al Terapista l'errore e chiede di inserire nuovamente i dati mancanti; 2. Il flusso riprende dal punto 4. del flusso principale.

ID Use Case	UC.4.3.
Nome	EliminaSchedaPaziente
Attori	Caregiver il cui ruolo è professionale (soprannominato <i>Terapista</i> in questo caso d'uso)
Descrizione	Il Terapista elimina una scheda paziente già presente nel sistema
Pre-condizioni	Esiste una scheda per il paziente nel Sistema
Flusso Principale	<ol style="list-style-type: none"> 1. Il Terapista chiede al Sistema di dare avvio alla procedura di eliminazione di una scheda paziente; 2. Il Sistema chiede di inserire il codice fiscale del paziente o il suo ID univoco; 3. L'Utente compila e conferma i dati inseriti; 4. Il Sistema controlla che sia presente una scheda per il paziente con il codice fiscale o l'ID univoco inserito; 5. Il Sistema chiede conferma al Terapista per la cancellazione della scheda; 6. Il Terapista dà conferma positiva; 7. Il Sistema segnala al Terapista che la procedura si è conclusa correttamente.
Post-condizioni	La Scheda del paziente è stata eliminata dal Sistema
Flusso Alternativo #1	<p>*.a. In qualsiasi momento il Terapista interrompe l'operazione:</p> <ol style="list-style-type: none"> 1. Il Sistema scarta eventuali dati inseriti e reindirizza il Terapista alla pagina da cui ha avviato la procedura.
Flusso Alternativo #2	<p>5.a. Il Sistema rileva che <i>non</i> esiste una scheda paziente per il codice fiscale inserito o per l'ID univoco inserito:</p> <ol style="list-style-type: none"> 1. Il Sistema segnala al Terapista l'errore e lo reindirizza alla pagina da cui ha avviato la procedura.
Flusso Alternativo #3	<p>6.a. Il Terapista dà conferma negativa:</p> <ol style="list-style-type: none"> 1. Il Sistema annulla l'operazione e reindirizza il Terapista alla pagina da cui ha avviato la procedura.

ID Use Case	UC.4.4.
Nome	InserisciNuovaDisabilita
Attori	Caregiver il cui ruolo è professionale (soprannominato <i>Terapista</i> in questo caso d'uso)
Descrizione	Il Terapista inserisce la scheda di una disabilità non presente nel Sistema
Pre-condizioni	Non esiste una disabilità con le stesse caratteristiche nel sistema
Flusso Principale	<ol style="list-style-type: none"> 1. Il Terapista chiede al Sistema di dare avvio alla procedura di inserimento di una nuova disabilità; 2. Il Sistema chiede di inserire i dati relativi alla disabilità (<i>Specifiche Aggiuntive SPEC.03</i>); 3. L'Utente compila e conferma i dati inseriti; 4. Il Sistema controlla che tutti i campi obbligatori siano stati compilati; 5. Il Sistema segnala al Terapista che la procedura si è conclusa correttamente.
Post-condizioni	Il Sistema contiene la scheda di una nuova disabilità
Flusso Alternativo #1	<p>*.a. In qualsiasi momento il Terapista interrompe l'operazione:</p> <ol style="list-style-type: none"> 1. Il Sistema scarta eventuali dati inseriti e reindirizza il Terapista alla pagina da cui ha avviato la procedura.
Flusso Alternativo #2	<p>4.a. Il Sistema rileva che <i>non</i> tutti i campi obbligatori sono stati compilati:</p> <ol style="list-style-type: none"> 1. Il Sistema segnala al Terapista l'errore e chiede di inserire nuovamente i dati mancanti; 2. Il flusso riprende dal punto 3. del flusso principale.

4.2.3.5 Specifica Use-Case: Gestione Piani Terapeutici

ID Use Case	UC.5.1.
Nome	InserisciPianoTerapeutico
Attori	Caregiver il cui ruolo è professionale (soprannominato <i>Terapista</i> in questo caso d'uso)
Descrizione	Il Terapista inserisce un nuovo piano terapeutico per un paziente
Pre-condizioni	Non esiste un piano terapeutico per il paziente nel Sistema
Flusso Principale	<ol style="list-style-type: none"> 1. Il Terapista, che si trova nella scheda di un paziente, chiede al Sistema di dare avvio alla procedura di inserimento di un piano terapeutico per il paziente; 2. Il Sistema chiede quali tipologie di esercizio inserire nel piano; 3. Il Terapista indica quali tipologie di esercizi inserire; 4. Per ogni tipologia di esercizio, il Sistema chiede di inserire una implementazione per l'esercizio; 5. Il Terapista segnala al Sistema che vuole concludere l'operazione di compilazione del piano terapeutico; 6. Il Sistema chiede conferma al Terapista; 7. Il Terapista dà conferma positiva; 8. Il Sistema segnala al Terapista che la procedura si è conclusa correttamente.
Post-condizioni	Il Sistema contiene un nuovo piano terapeutico per il paziente
Flusso Alternativo #1	<p>*.a. In qualsiasi momento il Terapista interrompe l'operazione:</p> <ol style="list-style-type: none"> 1. Il Sistema scarta eventuali dati inseriti e reindirizza il Terapista alla pagina da cui ha avviato la procedura.
Flusso Alternativo #2	<p>7.a. Il Terapista dà conferma negativa:</p> <ol style="list-style-type: none"> 1. Il Sistema annulla l'operazione e reindirizza il Terapista alla pagina da cui ha avviato la procedura.

ID Use Case	UC.5.2.
Nome	ModificaPianoTerapeutico
Attori	Caregiver il cui ruolo è professionale (soprannominato <i>Terapista</i> in questo caso d'uso)
Descrizione	Il Terapista modifica il piano terapeutico di un paziente
Pre-condizioni	Esiste un piano terapeutico per il paziente nel Sistema
Flusso Principale	<ol style="list-style-type: none"> 1. Il Terapista, che si trova nella scheda di un paziente, chiede al Sistema di dare avvio alla procedura di modifica del piano terapeutico del paziente; 2. Il Sistema chiede quali tipologie di esercizio inserire nel piano; 3. Il Terapista indica quali tipologie di esercizi inserire; 4. Per ogni tipologia di esercizio, il Sistema chiede di inserire una implementazione per l'esercizio; 5. Per ogni implementazione di esercizio già presente nel piano che il Terapista vuole aggiornare, il Sistema chiede al Terapista quali informazioni aggiornare; 6. Il Terapista compila i campi che vuole aggiornare; 7. Il Sistema controlla che tutti i campi obbligatori siano ancora compilati; 8. Il Terapista segnala al Sistema che vuole concludere l'operazione di modifica del piano terapeutico; 9. Il Sistema chiede conferma al Terapista; 10. Il Terapista dà conferma positiva; 11. Il Sistema segnala al Terapista che la procedura si è conclusa correttamente.
Post-condizioni	Il piano terapeutico del paziente è stato aggiornato con le nuove informazioni
Flusso Alternativo #1	<p>*.a. In qualsiasi momento il Terapista interrompe l'operazione:</p> <ol style="list-style-type: none"> 1. Il Sistema scarta eventuali dati inseriti e reindirizza il Terapista alla pagina da cui ha avviato la procedura

Flusso Alternativo #2	7.a. Il Sistema rileva che <i>non</i> tutti i campi obbligatori sono stati compilati: 1. Il Sistema segnala al Terapista l'errore e chiede di inserire nuovamente i dati mancanti; 2. Il flusso riprende dal punto 6. del flusso principale.
Flusso Alternativo #3	10.a. Il Terapista dà conferma negativa: 1. Il Sistema scarta eventuali dati inseriti e reindirizza il Terapista alla pagina da cui ha avviato la procedura.

ID Use Case	UC.5.3.
Nome	EliminaPianoTerapeutico
Attori	Caregiver il cui ruolo è professionale (soprannominato <i>Terapista</i> in questo caso d'uso)
Descrizione	Il Terapista elimina il piano terapeutico di un paziente
Pre-condizioni	Esiste un piano terapeutico per il paziente nel Sistema
Flusso Principale	1. Il Terapista, che si trova nella scheda di un paziente, chiede al Sistema di dare avvio alla procedura di eliminazione del piano terapeutico del paziente; 2. Il Sistema chiede conferma al Terapista; 3. Il Terapista dà conferma positiva; 4. Il Sistema segnala al Terapista che la procedura si è conclusa correttamente.
Post-condizioni	Il piano terapeutico del paziente è stato eliminato dal Sistema
Flusso Alternativo #1	*.a. In qualsiasi momento il Terapista interrompe l'operazione: 1. Il Sistema annulla l'operazione e reindirizza il Terapista alla pagina da cui ha avviato la procedura.
Flusso Alternativo #2	3.a. Il Terapista dà conferma negativa: 1. Il Sistema annulla l'operazione e reindirizza il Terapista alla pagina da cui ha avviato la procedura.

4.2.3.6 Specifica Use-Case: Gestione Tipologie e Abilità Esercizi

ID Use Case	UC.6.1.
Nome	InserisciTipologiaEsercizio
Attori	GestoreSistema
Descrizione	Il Gestore del Sistema inserisce nel Sistema una nuova tipologia di esercizio
Pre-condizioni	Nel Sistema non esiste una tipologia di esercizio con gli stessi dati
Flusso Principale	<ol style="list-style-type: none"> 1. Il Gestore chiede al Sistema di dare avvio alla procedura di inserimento di una nuova tipologia di esercizio; 2. Il Sistema chiede al Gestore di inserire i dati relativi alla tipologia di esercizio (<i>Specifiche Aggiuntive SPEC.04</i>); 3. Il Gestore compila e conferma i dati inseriti; 4. Il Sistema verifica che tutti i campi obbligatori siano stati compilati; 5. Il Sistema controlla che non ci sia un'altra tipologia di esercizio nel Sistema con gli stessi dati; 6. Il Sistema segnala al Gestore che la procedura si è conclusa correttamente.
Post-condizioni	Il Sistema contiene una nuova tipologia di esercizio
Flusso Alternativo #1	<p>*.a. In qualsiasi momento il Gestore interrompe l'operazione:</p> <ol style="list-style-type: none"> 1. Il Sistema annulla l'operazione e reindirizza il Gestore alla pagina da cui ha avviato la procedura.
Flusso Alternativo #2	<p>4.a. Il Sistema rileva che <i>non</i> tutti i campi obbligatori sono stati compilati:</p> <ol style="list-style-type: none"> 1. Il Sistema segnala al Gestore l'errore e chiede di inserire nuovamente i dati mancanti; 2. Il flusso riprende dal punto 3. del flusso principale
Flusso Alternativo #3	<p>5.a. Il Sistema rileva che esiste già una tipologia di esercizio con gli stessi dati:</p> <ol style="list-style-type: none"> 1. Il Sistema segnala al Gestore l'errore e lo reindirizza alla pagina da cui ha avviato la procedura

ID Use Case	UC.6.2.
Nome	InserisciAbilita
Attori	GestoreSistema
Descrizione	Il Gestore del Sistema inserisce nel Sistema una nuova abilità per gli esercizi
Pre-condizioni	Nel Sistema non esiste una abilità con gli stessi dati
Flusso Principale	<ol style="list-style-type: none"> 1. Il Gestore chiede al Sistema di dare avvio alla procedura di inserimento di una nuova abilità; 2. Il Sistema chiede al Gestore di inserire i dati relativi alla abilità da inserire (<i>Specifiche Aggiuntive SPEC.14</i>); 3. Il Gestore compila e conferma i dati inseriti; 4. Il Sistema verifica che tutti i campi obbligatori siano stati compilati; 5. Il Sistema controlla che non ci sia un'altra abilità nel Sistema con gli stessi dati; 6. Il Sistema segnala al Gestore che la procedura si è conclusa correttamente.
Post-condizioni	Il Sistema contiene una nuova abilità
Flusso Alternativo #1	<p>*.a. In qualsiasi momento il Gestore interrompe l'operazione:</p> <ol style="list-style-type: none"> 1. Il Sistema annulla l'operazione e reindirizza il Gestore alla pagina da cui ha avviato la procedura.
Flusso Alternativo #2	<p>4.a. Il Sistema rileva che <i>non</i> tutti i campi obbligatori sono stati compilati:</p> <ol style="list-style-type: none"> 1. Il Sistema segnala al Gestore l'errore e chiede di inserire nuovamente i dati mancanti; 2. Il flusso riprende dal punto 3. del flusso principale
Flusso Alternativo #3	<p>5.a. Il Sistema rileva che esiste già una abilità con gli stessi dati:</p> <ol style="list-style-type: none"> 1. Il Sistema segnala al Gestore l'errore e lo reindirizza alla pagina da cui ha avviato la procedura

4.2.3.7 Specifica Use-Case: Gestione Implementazioni Esercizi

ID Use Case	UC.7.1.
Nome	InserisciImplementazioneEsercizio
Attori	UtenteAutenticato (rinominato <i>Utente</i> in questo caso d'uso)
Descrizione	L'Utente inserisce una nuova implementazione per la tipologia di esercizio scelta
Pre-condizioni	Nel Sistema esiste la tipologia di esercizio che si vuole implementare
Flusso Principale	<ol style="list-style-type: none"> 1. L'Utente, che si trova nel piano terapeutico del paziente, chiede al Sistema di dare avvio alla procedura di inserimento di una nuova implementazione per un esercizio presente nel piano; 2. Il Sistema chiede all'Utente di inserire i dati relativi alla implementazione dell'esercizio (<i>Specifiche Aggiuntive SPEC.05</i>); 3. L'Utente compila e conferma i dati inseriti; 4. Il Sistema verifica che tutti i campi obbligatori siano stati compilati; 5. Il Sistema controlla che non ci sia un'altra implementazione per l'esercizio nel piano terapeutico per la stessa abilità; 6. Il Sistema segnala al Terapista che la procedura si è conclusa correttamente.
Post-condizioni	Il piano terapeutico contiene una nuova implementazione per un esercizio del piano stesso
Flusso Alternativo #1	<p>*.a. In qualsiasi momento il Terapista interrompe l'operazione:</p> <ol style="list-style-type: none"> 1. Il Sistema annulla l'operazione e reindirizza il Terapista alla pagina da cui ha avviato la procedura.
Flusso Alternativo #2	<p>4.a. Il Sistema rileva che <i>non</i> tutti i campi obbligatori sono stati compilati:</p> <ol style="list-style-type: none"> 1. Il Sistema segnala al Terapista l'errore e chiede di inserire nuovamente i dati mancanti; 2. Il flusso riprende dal punto 3. del flusso principale.
Flusso Alternativo #3	<p>5.a. Il Sistema rileva che esiste una implementazione per l'esercizio con gli stessi dati nel piano terapeutico:</p> <ol style="list-style-type: none"> 1. Il Sistema segnala al Terapista l'errore e lo reindirizza alla pagina da cui ha avviato la procedura.

4.2.3.8 Specifica Use-Case: Gestione Informazioni Esercizi

ID Use Case	UC.8.1.
Nome	AggiornaInformazioniEsercizi
Attori	Caregiver
Descrizione	Il Caregiver aggiorna le informazioni di una implementazione di esercizio inserita in un piano terapeutico di un paziente
Pre-condizioni	Nel Sistema esiste un piano terapeutico per il paziente in cui è presente almeno un'implementazione di esercizio
Flusso Principale	<ol style="list-style-type: none"> 1. Il Caregiver, che si trova nel piano terapeutico del paziente, chiede al Sistema di dare avvio alla procedura di aggiornamento delle informazioni di una implementazione per un esercizio presente nel piano; 2. Il Sistema chiede al Caregiver di inserire i dati da aggiornare (se il ruolo del caregiver è <i>professionale</i> allora <i>Specifiche Aggiuntive SPEC.12</i>, altrimenti se il ruolo del caregiver è <i>non professionale</i> allora <i>Specifiche Aggiuntive SPEC.13</i>); 3. Il Caregiver compila e conferma i dati inseriti; 4. Il Sistema verifica che tutti i campi obbligatori siano stati compilati e controlla che i dati inseriti siano validi (<i>Specifiche Aggiuntive SPEC.07</i>); 6. Il Sistema segnala al Caregiver che la procedura si è conclusa correttamente;
Post-condizioni	Il piano terapeutico contiene le informazioni aggiornate per l'implementazione di esercizio scelta
Flusso Alternativo #1	<p>*.a. In qualsiasi momento il Caregiver interrompe l'operazione:</p> <ol style="list-style-type: none"> 1. Il Sistema annulla l'operazione e reindirizza il Caregiver alla pagina da cui ha avviato la procedura.
Flusso Alternativo #2	<p>4.a. Il Sistema rileva che <i>non</i> tutti i campi obbligatori sono stati compilati o che i dati inseriti <i>non</i> sono validi:</p> <ol style="list-style-type: none"> 1. Il Sistema segnala al Caregiver l'errore e chiede di inserire nuovamente i dati mancanti o non validi; 2. Il flusso riprende dal punto 3. del flusso principale.

4.2.3.9 Specifica Use-Case: EsecuzioneEsercizio

ID Use Case	UC.9.1.
Nome	SceltaEserciziDaEseguire
Attori	UtenteAutenticato (rinominato <i>Utente</i> in questo caso d'uso), Bambino
Descrizione	L'Utente sceglie quale esercizio far svolgere al Bambino
Pre-condizioni	Nel Sistema esiste un piano terapeutico per il paziente in cui è presente almeno un'implementazione di esercizio
Flusso Principale	<ol style="list-style-type: none"> 1. L'Utente chiede al Sistema di dare avvio alla procedura di esecuzione degli esercizi presenti nel piano terapeutico del paziente; 2. Il Sistema chiede all'Utente quali esercizi del piano inserire nella routine di esecuzione degli esercizi; 3. L'Utente sceglie dal piano le implementazioni di esercizi da eseguire; 4. Il Sistema avvia l'esecuzione degli esercizi da far svolgere al Bambino; 5. Per ogni esercizio svolto, il Sistema salva le informazioni di esecuzione dell'esercizio da parte del Bambino;
Post-condizioni	Il Sistema contiene il risultato dell'esecuzione di una nuova routine di esercizi da parte del Bambino
Flusso Alternativo #1	<p>*.a. In qualsiasi momento l'Utente interrompe l'operazione:</p> <ol style="list-style-type: none"> 1. Il Sistema annulla l'operazione e reindirizza l'Utente alla pagina da cui ha avviato la procedura.

4.2.3.10 Specifica Use-Case: ConsultazioneDati

ID Use Case	UC.10.1.
Nome	ConsultaTipologieEsercizi
Attori	Caregiver il cui ruolo è professionale (soprannominato <i>Terapista</i> in questo caso d'uso)
Descrizione	Il Terapista consulta le tipologie di esercizi presenti nel Sistema
Pre-condizioni	Nel Sistema esiste almeno una tipologia di esercizi
Flusso Principale	<ol style="list-style-type: none"> 1. Il Terapista chiede al Sistema di consultare le tipologie di esercizi presenti; 2. Il Sistema mostra al Terapista tutte le tipologie di esercizi presenti; 3. Il Terapista sceglie la tipologia di esercizio di cui vuole consultare le informazioni; 4. Il Sistema mostra al Terapista i dati relativi alla tipologia di esercizio scelta.
Post-condizioni	Nessuna
Flusso Alternativo #1	*.a. In qualsiasi momento l'Utente interrompe l'operazione: <ol style="list-style-type: none"> 1. Il Sistema annulla l'operazione e reindirizza l'Utente alla pagina da cui ha avviato la procedura.

ID Use Case	UC.10.2.
Nome	ConsultaRisultatiEsercizi
Attori	Caregiver il cui ruolo è professionale (soprannominato <i>Terapista</i> in questo caso d'uso)
Descrizione	Il Terapista consulta i risultati dell'esecuzione delle implementazioni degli esercizi da parte del Bambino
Pre-condizioni	Nel Sistema esiste almeno una esecuzione di esercizio da parte del Bambino

Flusso Principale	<ol style="list-style-type: none"> 1. Il Terapista chiede al Sistema di consultare i risultati delle esecuzioni degli esercizi di un paziente; 2. Il Sistema mostra al Terapista tutte le routine di esecuzione degli esercizi che il paziente ha eseguito; 3. Il Terapista sceglie l'esecuzione di cui vuole che siano mostrati gli esercizi; 4. Il Sistema mostra al Terapista i dati relativi alla routine scelta.
Post-condizioni	Nessuna
Flusso Alternativo #1	<p>*.a. In qualsiasi momento l'Utente interrompe l'operazione:</p> <ol style="list-style-type: none"> 1. Il Sistema annulla l'operazione e reindirizza l'Utente alla pagina da cui ha avviato la procedura.

4.2.3.11 Specifica Use-Case: ConnessionePazienti

ID Use Case	UC.11.1.
Nome	ConnessionePazientiProfessionale
Attori	Caregiver il cui ruolo è professionale (soprannominato <i>Terapista</i> in questo caso d'uso)
Descrizione	Il Terapista chiede di connettersi come caregiver ad un Paziente di cui non è caregiver
Pre-condizioni	Nel Sistema esiste il Terapista e il Paziente cui vuole connettersi e i due non sono collegati
Flusso Principale	<ol style="list-style-type: none"> 1. Il Terapista chiede al Sistema di connettersi al Paziente a cui non è connesso; 2. Il Sistema chiede al Terapista di inserire il codice fiscale del Paziente; 3. Il Terapista inserisce e conferma il codice fiscale del Paziente; 4. Il Sistema verifica che tutti i campi obbligatori siano stati compilati e controlla che i dati inseriti siano validi; 5. Il Sistema collega il Terapista al Paziente.
Post-condizioni	Il Terapista e il Bambino sono connessi
Flusso Alternativo #1	*.a. In qualsiasi momento l'Utente interrompe l'operazione: <ol style="list-style-type: none"> 1. Il Sistema annulla l'operazione e reindirizza l'Utente alla pagina da cui ha avviato la procedura.
Flusso Alternativo #2	4.a. Il Sistema rileva che <i>non</i> tutti i campi obbligatori sono stati compilati o che i dati inseriti <i>non</i> sono validi: <ol style="list-style-type: none"> 1. Il Sistema segnala al Caregiver l'errore e chiede di inserire nuovamente i dati mancanti o non validi; 2. Il flusso riprende dal punto 2. del flusso principale.

ID Use Case	UC.11.2.
Nome	ConnessionePazientiNonProfessionale
Attori	Caregiver il cui ruolo è nonProfessionale (soprannominato <i>Terapista</i> in questo caso d'uso)
Descrizione	Il Terapista chiede di connettersi come caregiver ad un Paziente di cui non è caregiver

Pre-condizioni	Nel Sistema esiste il Terapista e il Paziente cui vuole connettersi e i due non sono collegati
Flusso Principale	<ol style="list-style-type: none"> 1. Il Terapista chiede al Sistema di connettersi al Paziente a cui non è connesso; 2. Il Sistema chiede al Terapista di inserire il codice fiscale del Paziente; 3. Il Terapista inserisce e conferma il codice fiscale del Paziente; 4. Il Sistema verifica che tutti i campi obbligatori siano stati compilati e controlla che i dati inseriti siano validi; 5. Il Sistema invia al caregiver tutore del Paziente una notifica contenente la richiesta del Terapista per essere accettato come Caregiver del Paziente.
Post-condizioni	Nel Sistema esiste una richiesta inviata al Tutore del Bambino per effettuare la connessione tra Caregiver e Bambino
Flusso Alternativo #1	<p>*.a. In qualsiasi momento l'Utente interrompe l'operazione:</p> <ol style="list-style-type: none"> 1. Il Sistema annulla l'operazione e reindirizza l'Utente alla pagina da cui ha avviato la procedura.
Flusso Alternativo #2	<p>4.a. Il Sistema rileva che <i>non</i> tutti i campi obbligatori sono stati compilati o che i dati inseriti <i>non</i> sono validi:</p> <ol style="list-style-type: none"> 1. Il Sistema segnala al Caregiver l'errore e chiede di inserire nuovamente i dati mancanti o non validi; 2. Il flusso riprende dal punto 2. del flusso principale.

ID Use Case	UC.11.3.
Nome	TutoreAccettaRichiesta
Attori	Caregiver il cui ruolo è nonProfessionale, tutore del paziente (soprannominato <i>Tutore</i> in questo caso d'uso)
Descrizione	Il Tutore accetta la richiesta di connessione di un caregiver non professionale per un bambino di cui è tutore

Pre-condizioni	Nel Sistema esiste una notifica per la richiesta da parte di un caregiver non professionista di diventare caregiver di un bambino di cui il Tutore è tutore
Flusso Principale	<ol style="list-style-type: none"> 1. Il Tutore apre la notifica inerente la richiesta di connessione del caregiver ad un bambino di cui è tutore; 2. Il Sistema mostra i dati del caregiver che ha effettuato la richiesta e del bambino cui si vuole connettere come caregiver; 3. Il Tutore conferma di voler accettare la richiesta; 4. Il Sistema chiede al tutore di inserire i codici fiscali del caregiver che si vuole connettere al bambino e del bambino per confermare la scelta; 5. Il Tutore inserisce e conferma i codici fiscali richiesti; 6. Il Sistema verifica che tutti i campi obbligatori siano stati compilati e controlla che i dati inseriti siano validi; 5. Il Sistema invia al caregiver che voleva connettersi al Paziente una notifica di avvenuta connessione.
Post-condizioni	Il Caregiver che aveva inviato la richiesta è collegato al Bambino richiesto
Flusso Alternativo #1	<p>*.a. In qualsiasi momento l'Utente interrompe l'operazione:</p> <ol style="list-style-type: none"> 1. Il Sistema annulla l'operazione e reindirizza l'Utente alla pagina da cui ha avviato la procedura.
Flusso Alternativo #2	<p>3.a. Il Tutore non accetta la richiesta:</p> <ol style="list-style-type: none"> 1. La notifica viene chiusa;
Flusso Alternativo #3	<p>6.a. Il Sistema rileva che <i>non</i> tutti i campi obbligatori sono stati compilati o che i dati inseriti <i>non</i> sono validi:</p> <ol style="list-style-type: none"> 1. Il Sistema segnala al Caregiver l'errore e chiede di inserire nuovamente i dati mancanti o non validi; 2. Il flusso riprende dal punto 4. del flusso principale.

ID Use Case	UC.11.4.
Nome	CaregiverCancellaNotifiche

Attori	Caregiver
Descrizione	Il Caregiver cancella una delle sue notifiche
Pre-condizioni	Nel Sistema esiste una notifica per il Caregiver
Flusso Principale	<ol style="list-style-type: none"> 1. Il Caregiver chiede di cancellare una sua notifica; 2. Il Sistema chiede conferma al Caregiver; 3. Il Caregiver conferma di voler cancellare la notifica; 4. Il Sistema cancella la notifica del Caregiver
Post-condizioni	La notifica cancellata non esiste più nel Sistema
Flusso Alternativo #1	<p>*.a. In qualsiasi momento l'Utente interrompe l'operazione:</p> <ol style="list-style-type: none"> 1. Il Sistema annulla l'operazione e reindirizza l'Utente alla pagina da cui ha avviato la procedura.
Flusso Alternativo #2	<p>3.a. Il Caregiver non conferma di voler eliminare la notifica:</p> <ol style="list-style-type: none"> 1. La notifica viene chiusa e non eliminata;

5 Specifiche Aggiuntive

Vengono ora riportate delle specifiche aggiuntive che non sono state inserite nei casi d'uso in quanto contengono dettagli a grana fine che non influiscono sull'esecuzione dei casi d'uso ma di cui si deve comunque tenere in considerazione durante l'implementazione di questi ultimi.

Identificativo della specifica	Descrizione	ID del relativo caso d'uso
SPEC.01	I dati necessari per la registrazione di un utente sono (<i>i campi sono tutti obbligatori</i>): Nome, Cognome, Ruolo, Email, Codice Fiscale	UC.1.1.
SPEC.02	I dati necessari per la registrazione di un nuovo paziente non presente nel sistema sono (<i>i campi sono tutti obbligatori</i>): Data di nascita, Età, Problematica riscontrata	UC.4.1. UC.4.2.
SPEC.03	I dati necessari per la registrazione di una nuova disabilità non presente nel sistema sono (<i>i campi sono tutti obbligatori</i>): Nome, Descrizione	UC.4.4.
SPEC.04	I dati necessari per la registrazione di una nuova tipologia di esercizio non presente nel sistema sono (<i>i campi sono tutti obbligatori</i>): Nome, Descrizione, Risultati Attesi, Modalità di Esecuzione	UC.6.1.
SPEC.05	I dati necessari per una nuova implementazione di esercizio sono (<i>i campi con l'asterisco sono obbligatori</i>): Abilità insegnata(*), Data di inizio (*), Data di apprendimento, Data di ripasso, Data di sospensione, Modalità di Risposta Accettata	UC.7.1.
SPEC.06	I dati necessari per l'inserimento di una implementazione di esercizio in un piano terapeutico sono (<i>i campi sono tutti obbligatori</i>): Data di inserimento	UC.5.1. UC.5.2.

Identificativo della specifica	Descrizione	ID del relativo caso d'uso
SPEC.07	<ul style="list-style-type: none"> • la <i>data di inserimento</i> è considerata valida solo se uguale o successiva alla data corrente • la <i>data di inizio</i> è considerata valida solo se uguale o successiva alla data di inserimento • la <i>data di apprendimento</i> è considerata valida solo se uguale o successiva alla data di inizio • le <i>date di esecuzione</i> sono considerate valide solo se uguali o successive alla data di inizio e precedenti alla data di apprendimento • la <i>data di sospensione</i> è considerata valida solo se uguale o successiva alla data di inizio • la <i>data di ripasso</i> è considerata valida solo se successiva alla data di apprendimento 	UC.8.1.
SPEC.08	Per ogni implementazione di esercizio può esserci solo una tra <i>data di apprendimento</i> e <i>data di sospensione</i>	UC.7.1.
SPEC.09	La password è considerata valida se contiene minimo 8 caratteri e massimo 16 caratteri (non accentati) in cui compaiono almeno un carattere minuscolo, un carattere maiuscolo, un carattere accentato e un carattere speciale	UC.1.1. UC.1.4. UC.2.1.
SPEC.10	I dati necessari all'autenticazione sono: Numero Identificativo (che viene assegnato dal sistema in fase di registrazione) o Email (indicata in fase di registrazione) e Password	UC.1.4.
SPEC.11	L'unico dato modificabile da un utente è la password	UC.2.1.
SPEC.12	I dati relativi ad un'implementazione di esercizio che possono essere modificati da un Terapista sono: Data di apprendimento, Data di ripasso, Data di sospensione, Modalità di Risposta Accettata	UC.8.1.

Identificativo della specifica	Descrizione	ID del relativo caso d'uso
SPEC.13	I dati relativi ad un'implementazione di esercizio che possono essere modificati da un Caregiver Non Specializzato sono: Data di esecuzione	UC.8.1.
SPEC.14	I dati necessari per la registrazione di una nuova abilità non presente nel sistema sono (<i>i campi sono tutti obbligatori</i>): Codice, Descrizione	UC.6.2.

6 Progettazione della Base Dati

6.1 Diagramma ER

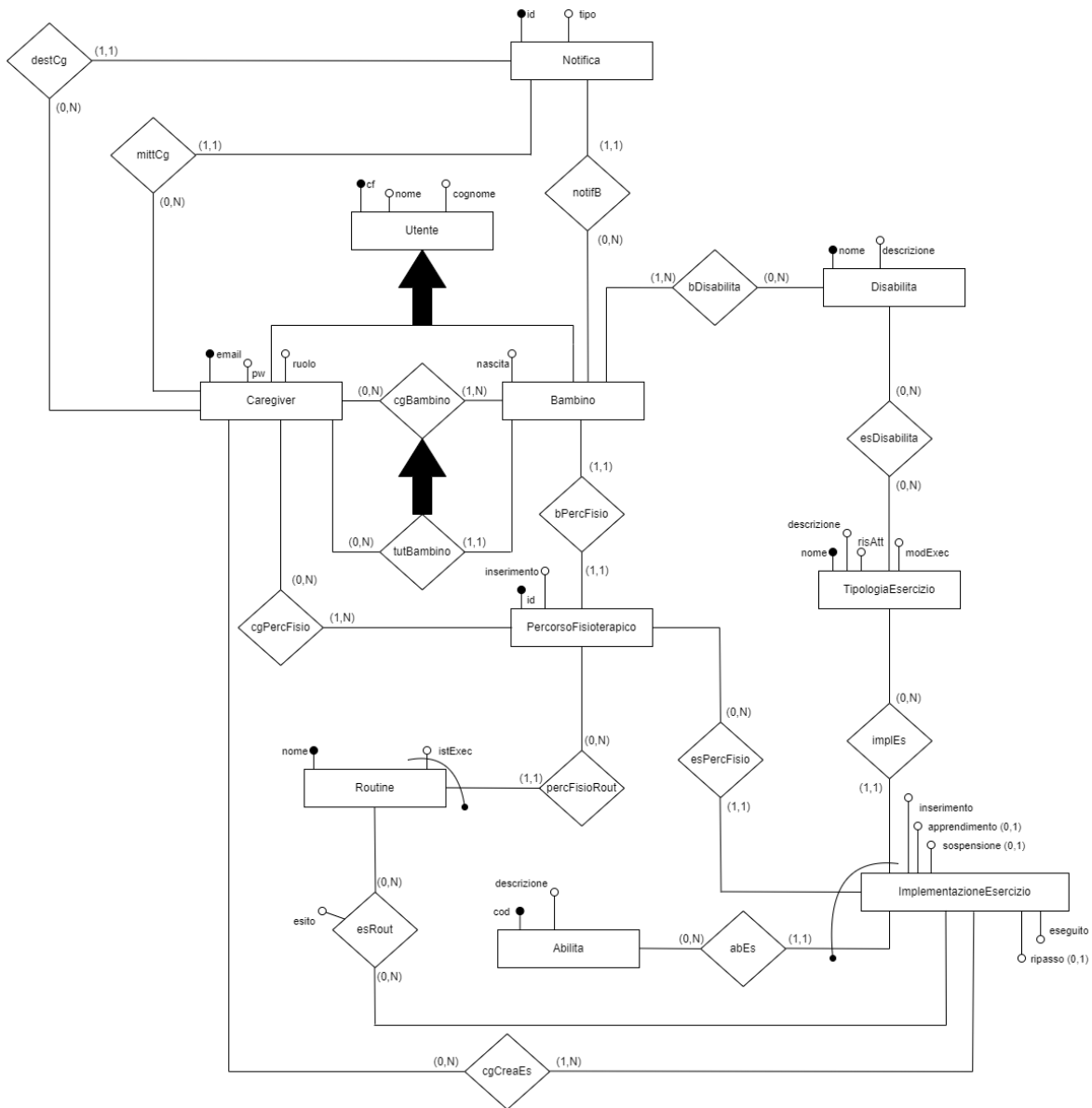


Figura 2: Diagramma ER della base dati

6.2 Specifiche dei dati

- **Entità Utente**

Ogni istanza di questa entità rappresenta un utente del Sistema

attributo	dominio	molteplicità	descrizione
nome	stringa		Il nome dell'utente
cognome	stringa		Il cognome dell'utente
cf	CodiceFiscale		Il codice fiscale dell'utente

Vincoli:

- **[V.Utente.cfUnico]**

Il codice fiscale deve essere univoco per ogni utente

$\forall u_1, u_2, cf_1, cf_2$

$Utente(u_1) \wedge Utente(u_2) \wedge cf(u_1, cf_1) \wedge cf(u_2, cf_2) \rightarrow cf_1 \neq cf_2$

- **Entità Caregiver**

Ogni istanza di questa entità rappresenta un utente del sistema registrato con il ruolo di caregiver

attributo	dominio	molteplicità	descrizione
email	Email		L'email del caregiver
password	Password		La password del caregiver
ruolo	{professionale, nonProfessionale}		Il ruolo ricoperto dal caregiver

Vincoli:

- **[V.Caregiver.emailUnica]**

L'email deve essere univoca per ogni utente

$\forall c_1, c_2, e_1, e_2$

$Caregiver(c_1) \wedge Caregiver(c_2) \wedge email(c_1, e_1) \wedge email(c_2, e_2)$

$\rightarrow e_1 \neq e_2$

- **[V.Caregiver.cgIsAUtente]**

Ogni profilo di un Caregiver deve essere associato ad un utente esistente nel sistema

$\forall c, codf$

$Caregiver(c) \wedge cf(c, codf) \rightarrow$

$\exists u$

$Utente(u) \wedge cf(u, codf)$

- **Entità Bambino**

Ogni istanza di questa entità rappresenta un utente del sistema che è stato registrato con il ruolo di bambino

attributo	dominio	molteplicità	descrizione
nascita	data		La data di nascita del bambino
eta	intero > 0		L'età del bambino espressa in anni

Vincoli:

- [V.Bambino.bIsAUtente]

Ogni profilo di un Bambino deve essere associato ad un utente esistente nel sistema

$\forall b, \text{codf}$

$\text{Bambino}(b) \wedge \text{cf}(b, \text{codf}) \rightarrow$

$\exists u$

$\text{Utente}(u) \wedge \text{cf}(u, \text{codf})$

- **Entità Notifica**

Ogni istanza di questa entità rappresenta una notifica che può rappresentare o una richiesta effettuata da un caregiver non professionale per collegarsi ad un bambino o una conferma dell'avvenuta connessione.

La richiesta viene poi inviata al caregiver tutore del bambino che può decidere se accettarla o cancellarla: in caso di accettazione viene inviata una notifica di conferma al caregiver che aveva effettuato la richiesta

attributo	dominio	molteplicità	descrizione
id	intero > 0		Identificativo della richiesta
tipo	{richiesta, conferma}		La tipologia della notifica

Vincoli:

- [V.Notifica.idUnico]

L'identificativo di ogni notifica deve essere univoco nel sistema

$\forall n_1, n_2, id_1, id_2$

$\text{Notifica}(n_1) \wedge \text{Notifica}(n_2) \wedge \text{id}(r_1, id_1) \wedge \text{id}(r_2, id_2)$

$\rightarrow id_1 \neq id_2$

- **Entità Disabilita**

Ogni istanza di questa entità rappresenta una forma di disabilità registrata nel Sistema

attributo	dominio	molteplicità	descrizione
nome	stringa		Il nome della disabilità
descrizione	stringa		La descrizione della disabilità

Vincoli:

- **[V.Disabilita.nomeUnico]**
 Il nome di ogni disabilità deve essere univoco nel sistema
 $\forall d_1, d_2, n_1, n_2$
 $\text{Disabilita}(d_1) \wedge \text{Disabilita}(d_2) \wedge \text{nome}(d_1, n_1) \wedge \text{nome}(d_2, n_2)$
 $\rightarrow n_1 \neq n_2$

- **Entità PercorsoFisioterapico**

Ogni istanza di questa entità rappresenta un percorso terapeutico assegnato ad un bambino

attributo	dominio	molteplicità	descrizione
id	intero		L'identificativo del percorso fisioterapico
inserimento	data		La data di inserimento del percorso fisioterapico nel sistema

Vincoli:

- **[V.PercorsoFisioterapico.idUnico]**
 L'id di ogni percorso fisioterapico deve essere univoco nel sistema
 $\forall pf_1, pf_2, id_1, id_2$
 $\text{PercorsoFisioterapico}(pf_1) \wedge \text{PercorsoFisioterapico}(pf_2) \wedge \text{id}(pf_1, id_1)$
 $\wedge \text{id}(pf_2, id_2) \rightarrow id_1 \neq id_2$

- **Entità TipologiaEsercizio**

Ogni istanza di questa entità rappresenta una tipologia di esercizio da far eseguire ad un paziente.

All'interno della stessa tipologia possono ricadere più esercizi specifici creati su misura delle esigenze del paziente.

Un esempio di tipologia di esercizi è l'accoppiamento di immagini che può poi essere caratterizzato in modo differente a seconda dell'immagine ritraente l'oggetto del mondo che si vuole insegnare al bambino.

attributo	dominio	molteplicità	descrizione
nome	stringa		Il nome dell'esercizio
descrizione	stringa		La descrizione delle azioni svolte dal caregiver per la preparazione dell'esecuzione dell'esercizio da parte del bambino
modExec	stringa		La descrizione della modalità di esecuzione dell'esercizio da parte del bambino
risAtt	stringa		La descrizione delle competenze apprese dal paziente a seguito dell'apprendimento dell'esercizio

Vincoli:

- **[V.TipologiaEsercizio.nomeUnico]**

Il nome della tipologia di esercizio deve essere univoco nel sistema

$\forall te_1, te_2, n_1, n_2$

$TipologiaEsercizio(te_1) \wedge TipologiaEsercizio(te_2) \wedge nome(te_1, n_1) \wedge nome(te_2, n_2) \rightarrow n_1 \neq n_2$

- **Entità ImplementazioneEsercizio**

Ogni istanza di questa entità rappresenta una specifica implementazione per un esercizio da inserire in un piano terapeutico di un paziente.

Ogni istanza quindi rappresenta uno specifico esercizio di una specifica tipologia che viene caratterizzato per un dato paziente.

Ad esempio, data la tipologia "accoppiamento di immagini" implementazioni diverse per l'esercizio possono essere l'accoppiamento pensato per insegnare al bambino l'oggetto del mondo "cane", quello per insegnargli l'oggetto del mondo "gatto" e così via.

Si noti che in alcuni casi gli esercizi presuppongono una risposta corretta da parte del paziente che cambia a seconda del paziente stesso (si tratta degli esercizi che ricadono nella tipologia "domande sociali" dove vengono effettivamente chieste al bambino informazioni sulla sua vita).

attributo	dominio	molteplicità	descrizione
inserimento	data		La data di inserimento dell'implementazione dell'esercizio nel piano terapeutico
apprendimento	data	(0,1)	La data di apprendimento dell'abilità insegnata dall'esercizio
sospensione	data	(0,1)	La data di sospensione dell'esercizio dal piano terapeutico del paziente
eseguito	booleano		Indica se l'esercizio è stato eseguito almeno una volta
respAcc	stringa	(0,1)	La risposta corretta da fornire in caso di domande
ripasso	data	(0,1)	La data di ripasso dell'abilità insegnata dall'esercizio

Vincoli:

– [V.ImplementazioneEsercizio.esStessaTipStessaAbUnicoInPercorsoFisio]

Questo vincolo comporta tre casistiche diverse qui di seguito riportate:

- * All'interno di uno stesso percorso terapeutico è ammessa una sola implementazione di esercizio per la stessa tipologia di esercizio e la stessa abilità;
- * All'interno di uno stesso percorso terapeutico due esercizi che riguardano la stessa abilità devono essere di tipologia diversa;
- * Due esercizi riguardanti la stessa abilità e della stessa tipologia devono far parte di percorsi terapeutici diversi

Qui di seguito viene riportata in logica del primo ordine solo la prima delle tre casistiche in quanto le altre due seguono una sintassi simile.

$\forall ie_1, ie_2, ab_1, ab_2, c_1, c_2, te, pf$

$ImplementazioneEsercizio(ie_1) \wedge ImplementazioneEsercizio(ie_2)$
 $\wedge Abilita(ab_1) \wedge Abilita(ab_2) \wedge cod(ab_1, c_1) \wedge cod(ab_2, c_2)$
 $\wedge abEs(ie_1, ab_1) \wedge abEs(ie_2, ab_2) \wedge TipologiaEsercizio(te)$
 $\wedge implEs(te, ie_1) \wedge implEs(te, ie_2)$
 $\wedge esPercFisio(pf, ie_1) \wedge esPercFisio(pf, ie_2) \rightarrow c_1 \neq c_2$

– [V.ImplementazioneEsercizio.apprendimentoLegale]

La data di apprendimento di un esercizio deve essere successiva o uguale alla data di inserimento dell'esercizio stesso nel piano terapeutico

$\forall i, ins, app$

$ImplementazioneEsercizio(i) \wedge inserimento(i, ins) \wedge apprendimento(i, app) \rightarrow app \geq ins$

– [V.ImplementazioneEsercizio.sospensioneLegale]

La data di sospensione di un esercizio deve essere successiva o uguale alla data di inserimento dell'esercizio stesso nel piano terapeutico

$\forall i, ins, sos$

$ImplementazioneEsercizio(i) \wedge inserimento(i, ins) \wedge sospensione(i, sos) \rightarrow sos \geq ins$

– [V.ImplementazioneEsercizio.esitoLegale]

Si può inserire la data di apprendimento o di sospensione per l'esercizio solamente se questo è stato eseguito almeno una volta dal bambino

$\forall i, app, sos$

$(ImplementazioneEsercizio(i) \wedge apprendimento(i, app)) \vee (ImplementazioneEsercizio(i) \wedge sospensione(i, sos)) \rightarrow eseguito(i, True)$

- [V.ImplementazioneEsercizio.appresoOSospeso]
 Un esercizio può essere considerato in modo mutualmente esclusivo
 solamente appreso o sospeso a seguito di almeno una sua esecuzione
 $\forall i$
 $\text{ImplementazioneEsercizio}(i) \wedge \text{eseguito}(i, \text{True}) \rightarrow$
 $\exists \text{app}$
 $\text{apprendimento}(i, \text{app})$
 \vee
 $\exists \text{sos}$
 $\text{sospensione}(i, \text{sos})$
- [V.ImplementazioneEsercizio.ripassoLegale]
 La data di ripasso di un esercizio deve essere successiva alla sua data
 di apprendimento
 $\forall i, \text{app}, \text{rip}$
 $\text{ImplementazioneEsercizio}(i) \wedge \text{apprendimento}(i, \text{app}) \wedge \text{ripasso}(i,$
 $\text{rip}) \rightarrow \text{rip} > \text{app}$

- **Entità Abilita**

Ogni istanza di questa entità rappresenta una abilità che può essere insegnata ad un paziente tramite esercizi.

Si noti che per abilità si intendono tutti i concetti del mondo che vengono generalmente appresi dai bambini nel corso della loro crescita.

Esempi di abilità possono essere "cane", "gatto", "gioia", "tristezza", "ordine", "attesa" e così via.

attributo	dominio	molteplicità	descrizione
cod	intero		Il codice identificativo della abilità nella base dati
descrizione	stringa		Descrizione dell'abilità

Vincoli:

- [V.Abilita.codUnico]
 Il codice identificativo di una abilità deve essere univoco nel sistema
 $\forall ab_1, ab_2, c_1, c_2$
 $\text{Abilita}(ab_1) \wedge \text{Abilita}(ab_2) \wedge \text{cod}(ab_1, c_1) \wedge \text{cod}(ab_2, c_2) \rightarrow c_1 \neq c_2$

- **Entità Routine**

Ogni istanza di questa entità rappresenta una routine di esercizi (presenti nel piano terapeutico) da far eseguire ad un paziente.

Per routine di esercizi si intende quindi il sottogruppo di esercizi assegnati ad un paziente (e quindi presenti nel suo piano terapeutico) che possono essere eseguiti dal bambino in un dato momento di utilizzo dell'applicazione di supporto ai caregiver.

attributo	dominio	molteplicità	descrizione
nome	NomeRoutine		Il nome della routine
istExec	dataOra		L'istante di inizio di esecuzione della routine

Vincoli:

- **[V.Routine.nomeUnico]**

Il nome identificativo di una routine deve essere univoco nel sistema

$\forall r_1, r_2, n_1, n_2$

$Routine(r_1) \wedge Routine(r_2) \wedge n(r_1, n_1) \wedge n(r_2, n_2) \rightarrow n_1 \neq n_2$

- **[V.Routine.istExecUnicoPerPercorso]**

Può essere avviata solo una routine di esecuzione di esercizi (inerente allo stesso percorso terapeutico) per volta

$\forall r_1, r_2, iste_1, iste_2, pf$

$Routine(r_1) \wedge Routine(r_2)$

$\wedge istExec(r_1, iste_1) \wedge istExec(r_2, iste_2) \wedge PercorsoFisioterapico(pf)$

$\wedge percFisioRout(r_1, pf) \wedge percFisioRout(r_2, pf) \rightarrow iste_1 \neq iste_2$

- **Relationship cgBambino**

Ogni istanza di questa relationship lega un caregiver con i bambini di cui si occupa

Attributi: nessuno

Vincoli: nessuno

- **Relationship tutBambino**

Ogni istanza di questa relationship lega un caregiver con i bambini di cui è tutore

Attributi: nessuno

Vincoli: nessuno

- **Relationship mittCg**

Ogni istanza di questa relationship lega una notifica con il caregiver che l'ha inviata

Attributi: nessuno

Vincoli: nessuno

- **Relationship notifB**
 Ogni istanza di questa relationship lega una notifica con il bambino per cui è stata inviata la notifica
Attributi: nessuno
Vincoli: nessuno
- **Relationship destCg**
 Ogni istanza di questa relationship lega una notifica con il caregiver a cui è destinata
Attributi: nessuno
Vincoli: nessuno
- **Relationship bDisabilita**
 Ogni istanza di questa relationship lega un bambino con le disabilità (almeno una) che gli sono state diagnosticate
Attributi: nessuno
Vincoli: nessuno
- **Relationship esDisabilita**
 Ogni istanza di questa relationship lega un esercizio con le disabilità per cui è indicato
Attributi: nessuno
Vincoli: nessuno
- **Relationship bPercFisio**
 Ogni istanza di questa relationship lega un bambino con il percorso fisioterapico creato per lui
Attributi: nessuno
Vincoli: nessuno
- **Relationship cgPercFisio**
 Ogni istanza di questa relationship lega un percorso fisioterapico con i caregiver (almeno uno) che lo hanno creato
Attributi: nessuno
Vincoli:
 - [V.cgPercFisio.cgSoloProf]
 I percorsi terapeutici possono essere creati solo da caregiver il cui ruolo è professionale
 $\forall \text{cg, pf}$
 $\text{Caregiver}(\text{cg}) \wedge \text{PercorsoFisioterapico}(\text{pf}) \wedge \text{cgPercFisio}(\text{cg, pf}) \rightarrow \text{ruolo}(\text{cg, professionale})$
- **Relationship esPercFisio**
 Ogni istanza di questa relationship lega un percorso fisioterapico con le implementazioni di esercizi presenti al suo interno
Attributi: nessuno
Vincoli: nessuno

- **Relationship cgCreaEs**

Ogni istanza di questa relationship lega un caregiver con le implementazioni di esercizio che ha creato

Attributi: nessuno

Vincoli: nessuno

- **Relationship implEs**

Ogni istanza di questa relationship lega una implementazione di esercizio con la tipologia di esercizio a cui appartiene

Attributi: nessuno

Vincoli: nessuno

- **Relationship percFisioRout**

Ogni istanza di questa relationship lega una routine con il percorso fisioterapico da cui sta prendendo gli esercizi

Attributi: nessuno

Vincoli: nessuno

- **Relationship esRout**

Ogni istanza di questa relationship lega una routine con gli esercizi che ne fanno parte.

Si noti che la stessa implementazione di esercizio può essere presente in più routine in quanto lo stesso esercizio assegnato al paziente può essere svolto più volte dal paziente stesso durante diverse sedute di terapia.

L'esecuzione dell'esercizio da parte del paziente viene quindi salvata sotto forma dell'esito che è legato alla coppia formata dall'implementazione dell'esercizio con la routine in quanto rappresenta se l'esercizio è stato correttamente eseguito dal paziente oppure no in una data sessione di svolgimento di esercizi.

L'esito può quindi poi essere consultato da parte dei caregiver professionisti per decidere se considerare concluso il periodo di tempo di apprendimento dell'esercizio e per constatare se questo deve essere ritenuto appreso oppure sospeso

attributo	dominio	molteplicità	descrizione
esito	stringa		Esito dell'esecuzione del particolare esercizio inserito in una specifica routine

Vincoli:

- [V.esRout.esLegali]

Un esercizio può essere inserito in una routine inerente ad un percorso terapeutico solamente se è presente nel percorso terapeutico stesso

$\forall r, es, pf$

$Routine(r) \wedge ImplementazioneEsercizio(es) \wedge PercorsoFisioterapico(pf) \wedge esRout(r, es) \wedge percFisioRout(r, pf) \rightarrow esPercFisio(pf, es)$

- **Dominio Email**

Il dominio è una stringa che rispetta gli standard di formattazione di un indirizzo email

- **Dominio Password**

Il dominio è una stringa che rispetta gli standard di formattazione di una password

- **Dominio CodiceFiscale**

Il dominio è una stringa che rispetta gli standard di formattazione di un codice fiscale

- **Dominio NomeRoutine**

Il dominio è una stringa con il seguente formato:

ID_percorso_fisioterapico-istExec_routine

6.3 Ristrutturazione del Diagramma ER e delle Specifiche dei Dati

6.3.1 Diagramma ER Ristrutturato

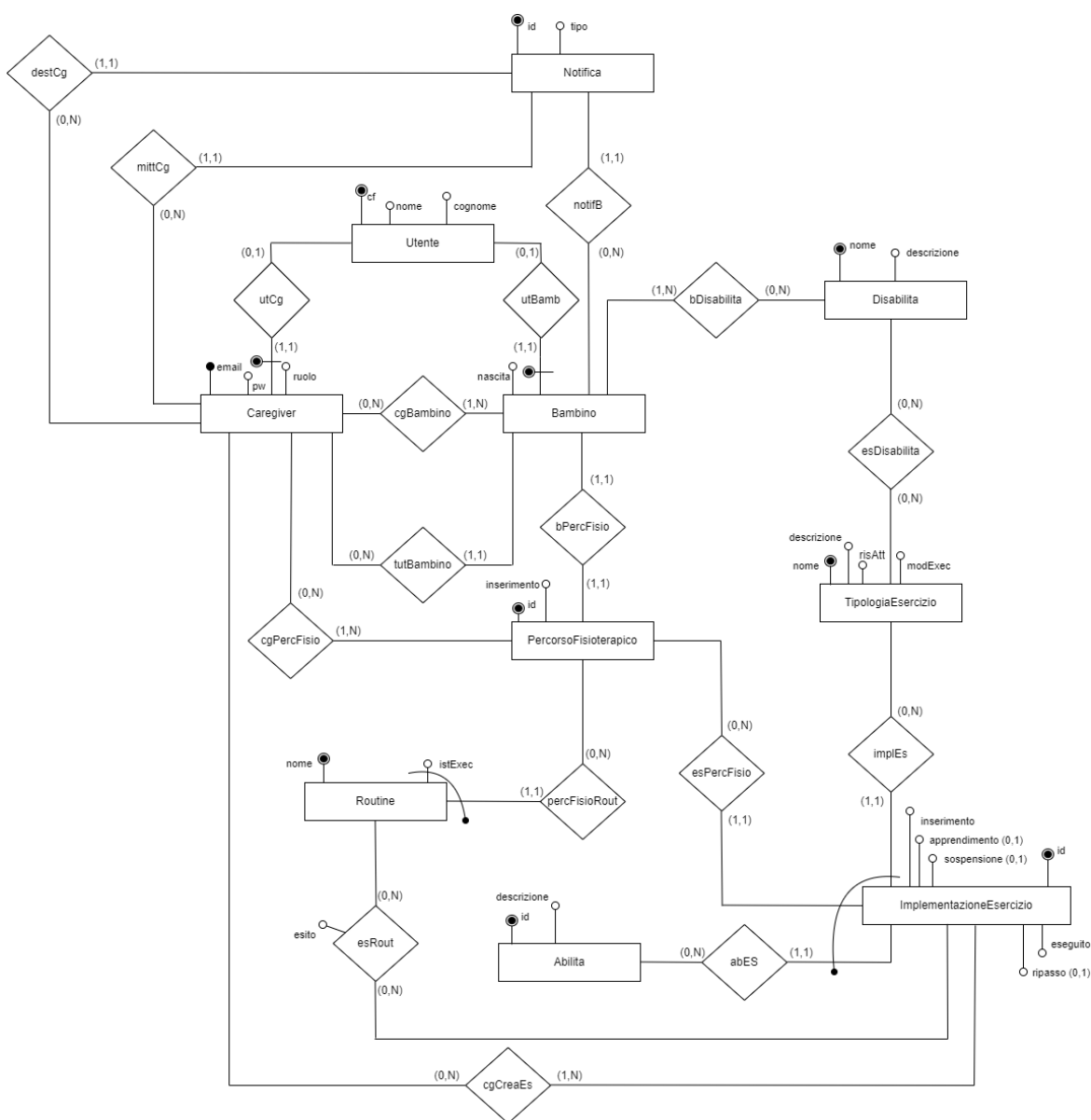


Figura 3: Diagramma ER della base dati ristrutturato

6.3.2 Specifiche dei Dati Ristrutturate

- **Entità Utente**

Ogni istanza di questa entità rappresenta un utente del Sistema

attributo	dominio	molteplicità	descrizione
nome	StringS		Il nome dell'utente
cognome	StringS		Il cognome dell'utente
cf	CodiceFiscale		Il codice fiscale dell'utente

Vincoli:

- **[V.Utente.cfUnico]**
Il codice fiscale deve essere univoco per ogni utente
 $\forall u_1, u_2, cf_1, cf_2$
 $Utente(u_1) \wedge Utente(u_2) \wedge cf(u_1, cf_1) \wedge cf(u_2, cf_2) \rightarrow cf_1 \neq cf_2$
- **[V.Utente.completa]**
Ogni utente del sistema deve essere o un caregiver o un bambino
 $\forall ut$
 $Utente(ut) \rightarrow \exists cg, b$
 $(Caregiver(cg) \wedge utCg(ut, cg)) \vee (Bambino(b) \wedge utBmab(ut, b))$
- **[V.Utente.disgiunta]**
Un utente del sistema che è registrato come caregiver non può essere registrato come bambino
 $\forall ut, cg$
 $Utente(ut) \wedge Caregiver(cg) \wedge utCg(ut, cg) \rightarrow \nexists b$
 $Bambino(b) \wedge utBamb(ut, b)$

- **Entità Caregiver**

Ogni istanza di questa entità rappresenta un utente del sistema registrato con il ruolo di caregiver

attributo	dominio	molteplicità	descrizione
email	Email		L'email del caregiver
password	Password		La password del caregiver
ruolo	Ruolo		Il ruolo ricoperto dal caregiver

Vincoli:

- **[V.Caregiver.emailUnica]**
L'email deve essere univoca per ogni utente
 $\forall c_1, c_2, e_1, e_2$
 $Caregiver(c_1) \wedge Caregiver(c_2) \wedge email(c_1, e_1) \wedge email(c_2, e_2)$
 $\rightarrow e_1 \neq e_2$

– [V.Caregiver.cgIsAUtente]

Ogni profilo di un Caregiver deve essere associato ad un utente esistente nel sistema

$\forall c, \text{codf}$

$\text{Caregiver}(c) \wedge \text{cf}(c, \text{codf}) \rightarrow$

$\exists u$

$\text{Utente}(u) \wedge \text{cf}(u, \text{codf})$

• **Entità Bambino**

Ogni istanza di questa entità rappresenta un utente del sistema che è stato registrato con il ruolo di bambino

attributo	dominio	molteplicità	descrizione
nascita	date		La data di nascita del bambino
eta	IntegerGZ		L'età del bambino espressa in anni

Vincoli:

– [V.Bambino.bIsAUtente]

Ogni profilo di un Bambino deve essere associato ad un utente esistente nel sistema

$\forall b, \text{codf}$

$\text{Bambino}(b) \wedge \text{cf}(b, \text{codf}) \rightarrow$

$\exists u$

$\text{Utente}(u) \wedge \text{cf}(u, \text{codf})$

• **Entità Notifica**

Ogni istanza di questa entità rappresenta una notifica che può rappresentare o una richiesta effettuata da un caregiver non professionale per collegarsi ad un bambino o una conferma dell'avvenuta connessione.

La richiesta viene poi inviata al caregiver tutore del bambino che può decidere se accettarla o cancellarla: in caso di accettazione viene inviata una notifica di conferma al caregiver che aveva effettuato la richiesta

attributo	dominio	molteplicità	descrizione
id	IntegerGZ		Identificativo della richiesta
tipo	TipoNotifica		La tipologia della notifica

Vincoli:

– [V.Notifica.idUnico]

L'identificativo di ogni notifica deve essere univoco nel sistema

$\forall n_1, n_2, id_1, id_2$

$\text{Notifica}(n_1) \wedge \text{Notifica}(n_2) \wedge id(n_1, id_1) \wedge id(n_2, id_2)$

$\rightarrow id_1 \neq id_2$

- **Entità Disabilita**

Ogni istanza di questa entità rappresenta una forma di disabilità registrata nel Sistema

attributo	dominio	molteplicità	descrizione
nome	StringS		Il nome della disabilità
descrizione	StringL		La descrizione della disabilità

Vincoli:

- [V.Disabilita.nomeUnico]
Il nome di ogni disabilità deve essere univoco nel sistema
 $\forall d_1, d_2, n_1, n_2$
 $\text{Disabilita}(d_1) \wedge \text{Disabilita}(d_2) \wedge \text{nome}(d_1, n_1) \wedge \text{nome}(d_2, n_2)$
 $\rightarrow n_1 \neq n_2$

- **Entità PercorsoFisioterapico**

Ogni istanza di questa entità rappresenta un percorso terapeutico assegnato ad un bambino

attributo	dominio	molteplicità	descrizione
id	IntegerGZ		L'identificativo del percorso fisioterapico
inserimento	date		La data di inserimento del percorso fisioterapico nel sistema

Vincoli:

- [V.PercorsoFisioterapico.idUnico]
L'id di ogni percorso fisioterapico deve essere univoco nel sistema
 $\forall pf_1, pf_2, id_1, id_2$
 $\text{PercorsoFisioterapico}(pf_1) \wedge \text{PercorsoFisioterapico}(pf_2) \wedge \text{id}(pf_1, id_1)$
 $\wedge \text{id}(pf_2, id_2) \rightarrow id_1 \neq id_2$

- **Entità TipologiaEsercizio**

Ogni istanza di questa entità rappresenta una tipologia di esercizio da far eseguire ad un paziente.

All'interno della stessa tipologia possono ricadere più esercizi specifici creati su misura delle esigenze del paziente.

Un esempio di tipologia di esercizi è l'accoppiamento di immagini che può poi essere caratterizzato in modo differente a seconda dell'immagine ritraente l'oggetto del mondo che si vuole insegnare al bambino.

attributo	dominio	molteplicità	descrizione
nome	StringS		Il nome dell'esercizio
descrizione	StringL		La descrizione delle azioni svolte dal caregiver per la preparazione dell'esecuzione dell'esercizio da parte del bambino
modExec	StringL		La descrizione della modalità di esecuzione dell'esercizio da parte del bambino
risAtt	StringL		La descrizione delle competenze apprese dal paziente a seguito dell'apprendimento dell'esercizio

Vincoli:

- [V.TipologiaEsercizio.nomeUnico]

Il nome della tipologia di esercizio deve essere univoco nel sistema

$\forall te_1, te_2, n_1, n_2$

$TipologiaEsercizio(te_1) \wedge TipologiaEsercizio(te_2) \wedge nome(te_1, n_1) \wedge$

$nome(te_2, n_2) \rightarrow n_1 \neq n_2$

- **Entità ImplementazioneEsercizio**

Ogni istanza di questa entità rappresenta una specifica implementazione per un esercizio da inserire in un piano terapeutico di un paziente.

Ogni istanza quindi rappresenta uno specifico esercizio di una specifica tipologia che viene caratterizzato per un dato paziente.

Ad esempio, data la tipologia "accoppiamento di immagini" implementazioni diverse per l'esercizio possono essere l'accoppiamento pensato per insegnare al bambino l'oggetto del mondo "cane", quello per insegnargli l'oggetto del mondo "gatto" e così via.

Si noti che in alcuni casi gli esercizi presuppongono una risposta corretta da parte del paziente che cambia a seconda del paziente stesso (si tratta degli esercizi che ricadono nella tipologia "domande sociali" dove vengono effettivamente chieste al bambino informazioni sulla sua vita).

attributo	dominio	molteplicità	descrizione
id	IntegerGZ		L'identificativo dell'esercizio nel Sistema
inserimento	date		La data di inserimento dell'implementazione dell'esercizio nel piano terapeutico
apprendimento	date	(0,1)	La data di apprendimento dell'abilità insegnata dall'esercizio
sospensione	date	(0,1)	La data di sospensione dell'esercizio dal piano terapeutico del paziente
eseguito	boolean		Indica se l'esercizio è stato eseguito almeno una volta
respAcc	StringS	(0,1)	La risposta accettata per l'esecuzione dell'esercizio
ripasso	date	(0,1)	La data di ripasso dell'abilità insegnata dall'esercizio

Vincoli:

- **[V.ImplementazioneEsercizio.idUnico]**

L'identificativo di una implementazione di esercizio deve essere univoco nel sistema

$\forall ie_1, ie_2, id_1, id_2$

$ImplementazioneEsercizio(ie_1) \wedge ImplementazioneEsercizio(ie_2) \wedge id(ie_1, id_1) \wedge id(ie_2, id_2) \rightarrow id_1 \neq id_2$

– **[V.ImplementazioneEsercizio.esStessaTipStessaAbUnicoInPercorsoFisio]**

Questo vincolo comporta tre casistiche diverse qui di seguito riportate:

- * All'interno di uno stesso percorso terapeutico è ammessa una sola implementazione di esercizio per la stessa tipologia di esercizio e la stessa abilità;
- * All'interno di uno stesso percorso terapeutico due esercizi che riguardano la stessa abilità devono essere di tipologia diversa;
- * Due esercizi riguardanti la stessa abilità e della stessa tipologia devono far parte di percorsi terapeutici diversi

Qui di seguito viene riportata in logica del primo ordine solo la prima delle tre casistiche in quanto le altre due seguono una sintassi simile.

$$\begin{aligned} & \forall ie_1, ie_2, ab_1, ab_2, c_1, c_2, te, pf \\ & \text{ImplementazioneEsercizio}(ie_1) \wedge \text{ImplementazioneEsercizio}(ie_2) \\ & \wedge \text{Abilita}(ab_1) \wedge \text{Abilita}(ab_2) \wedge \text{cod}(ab_1, c_1) \wedge \text{cod}(ab_2, c_2) \\ & \wedge \text{abEs}(ie_1, ab_1) \wedge \text{abEs}(ie_2, ab_2) \wedge \text{TipologiaEsercizio}(te) \\ & \wedge \text{implEs}(te, ie_1) \wedge \text{implEs}(te, ie_2) \\ & \wedge \text{esPercFisio}(pf, ie_1) \wedge \text{esPercFisio}(pf, ie_2) \rightarrow c_1 \neq c_2 \end{aligned}$$

– **[V.ImplementazioneEsercizio.apprendimentoLegale]**

La data di apprendimento di un esercizio deve essere successiva o uguale alla data di inserimento dell'esercizio stesso nel piano terapeutico

$$\begin{aligned} & \forall i, ins, app \\ & \text{ImplementazioneEsercizio}(i) \wedge \text{inserimento}(i, ins) \wedge \text{apprendimento}(i, app) \rightarrow app \geq ins \end{aligned}$$

– **[V.ImplementazioneEsercizio.sospensioneLegale]**

La data di sospensione di un esercizio deve essere successiva o uguale alla data di inserimento dell'esercizio stesso nel piano terapeutico

$$\begin{aligned} & \forall i, ins, sos \\ & \text{ImplementazioneEsercizio}(i) \wedge \text{inserimento}(i, ins) \wedge \text{sospensione}(i, sos) \rightarrow sos \geq ins \end{aligned}$$

– **[V.ImplementazioneEsercizio.esitoLegale]**

Si può inserire la data di apprendimento o di sospensione per l'esercizio solamente se questo è stato eseguito almeno una volta dal bambino

$$\begin{aligned} & \forall i, app, sos \\ & (\text{ImplementazioneEsercizio}(i) \wedge \text{apprendimento}(i, app)) \vee (\text{ImplementazioneEsercizio}(i) \wedge \text{sospensione}(i, sos)) \rightarrow \text{eseguito}(i, \text{True}) \end{aligned}$$

- [V.ImplementazioneEsercizio.appresoOSospeso]
 Un esercizio può essere considerato in modo mutualmente esclusivo
 solamente appreso o sospeso a seguito di almeno una sua esecuzione
 $\forall i$
 $\text{ImplementazioneEsercizio}(i) \wedge \text{eseguito}(i, \text{True}) \rightarrow$
 $\exists \text{app}$
 $\text{apprendimento}(i, \text{app})$
 \vee
 $\exists \text{sos}$
 $\text{sospensione}(i, \text{sos})$
- [V.ImplementazioneEsercizio.ripassoLegale]
 La data di ripasso di un esercizio deve essere successiva alla sua data
 di apprendimento
 $\forall i, \text{app}, \text{rip}$
 $\text{ImplementazioneEsercizio}(i) \wedge \text{apprendimento}(i, \text{app}) \wedge \text{ripasso}(i,$
 $\text{rip}) \rightarrow \text{rip} > \text{app}$

- **Entità Abilita**

Ogni istanza di questa entità rappresenta una abilità che può essere insegnata ad un paziente tramite esercizi.

Si noti che per abilità si intendono tutti i concetti del mondo che vengono generalmente appresi dai bambini nel corso della loro crescita.

Esempi di abilità possono essere "cane", "gatto", "gioia", "tristezza", "ordine", "attesa" e così via.

attributo	dominio	molteplicità	descrizione
cod	IntegerGZ		Il codice identificativo della abilità nella base dati
descrizione	StringM		Descrizione dell'abilità

Vincoli:

- [V.Abilita.codUnico]
 Il codice identificativo di una abilità deve essere univoco nel sistema
 $\forall ab_1, ab_2, c_1, c_2$
 $\text{Abilita}(ab_1) \wedge \text{Abilita}(ab_2) \wedge \text{cod}(ab_1, c_1) \wedge \text{cod}(ab_2, c_2) \rightarrow c_1 \neq c_2$

- **Entità Routine**

Ogni istanza di questa entità rappresenta una routine di esercizi (presenti nel piano terapeutico) da far eseguire ad un paziente.

Per routine di esercizi si intende quindi il sottogruppo di esercizi assegnati ad un paziente (e quindi presenti nel suo piano terapeutico) che possono essere eseguiti dal bambino in un dato momento di utilizzo dell'applicazione di supporto ai caregiver.

attributo	dominio	molteplicità	descrizione
nome	NomeRoutine		Il nome della routine
istExec	timestamp		L'istante di inizio di esecuzione della routine

Vincoli:

- **[V.Routine.nomeUnico]**

Il nome identificativo di una routine deve essere univoco nel sistema

$\forall r_1, r_2, n_1, n_2$

$Routine(r_1) \wedge Routine(r_2) \wedge n(r_1, n_1) \wedge n(r_2, n_2) \rightarrow n_1 \neq n_2$

- **[V.Routine.istExecUnicoPerPercorso]**

Può essere avviata solo una routine di esecuzione di esercizi (inerente allo stesso percorso terapeutico) per volta

$\forall r_1, r_2, iste_1, iste_2, pf$

$Routine(r_1) \wedge Routine(r_2)$

$\wedge istExec(r_1, iste_1) \wedge istExec(r_2, iste_2) \wedge PercorsoFisioterapico(pf)$

$\wedge percFisioRout(r_1, pf) \wedge percFisioRout(r_2, pf) \rightarrow iste_1 \neq iste_2$

- **Relationship utCg**

Ogni istanza di questa relationship lega un utente con il suo ruolo di caregiver nel sistema

Attributi: nessuno

Vincoli: nessuno

- **Relationship utBamb**

Ogni istanza di questa relationship lega un utente con il suo ruolo di bambino nel sistema

Attributi: nessuno

Vincoli: nessuno

- **Relationship cgBambino**

Ogni istanza di questa relationship lega un caregiver con i bambini di cui si occupa

Attributi: nessuno

Vincoli: nessuno

- **Relationship tutBambino**
 Ogni istanza di questa relationship lega un caregiver con i bambini di cui è tutore
Attributi: nessuno
Vincoli:
 - [V.tutBambino.tutoreIsACaregiver]
 Il tutore di un bambino deve essere anche suo caregiver
 $\forall \text{ tut, b}$
 $\text{Caregiver}(\text{tut}) \wedge \text{Bambino}(\text{b}) \wedge \text{tutBambino}(\text{tut, b}) \rightarrow \text{cgBambino}(\text{tut, b})$
- **Relationship mittCg**
 Ogni istanza di questa relationship lega una notifica con il caregiver che l'ha inviata
Attributi: nessuno
Vincoli: nessuno
- **Relationship notifB**
 Ogni istanza di questa relationship lega una notifica con il bambino per cui è stata inviata la notifica
Attributi: nessuno
Vincoli: nessuno
- **Relationship destCg**
 Ogni istanza di questa relationship lega una notifica con il caregiver a cui è destinata
Attributi: nessuno
Vincoli: nessuno
- **Relationship bDisabilita**
 Ogni istanza di questa relationship lega un bambino con le disabilità (almeno una) che gli sono state diagnosticate
Attributi: nessuno
Vincoli: nessuno
- **Relationship esDisabilita**
 Ogni istanza di questa relationship lega un esercizio con le disabilità per cui è indicato
Attributi: nessuno
Vincoli: nessuno
- **Relationship bPercFisio**
 Ogni istanza di questa relationship lega un bambino con il percorso fisioterapico creato per lui
Attributi: nessuno
Vincoli: nessuno

- **Relationship cgPercFisio**

Ogni istanza di questa relationship lega un percorso fisioterapico con i caregiver (almeno uno) che lo hanno creato

Attributi: nessuno

Vincoli:

- [V.cgPercFisio.cgSoloProf]

I percorsi terapeutici possono essere creati solo da caregiver il cui ruolo è professionale

$\forall \text{ cg, pf}$

$\text{Caregiver}(\text{cg}) \wedge \text{PercorsoFisioterapico}(\text{pf}) \wedge \text{cgPercFisio}(\text{cg, pf}) \rightarrow \text{ruolo}(\text{cg, professionale})$

- **Relationship esPercFisio**

Ogni istanza di questa relationship lega un percorso fisioterapico con le implementazioni di esercizi presenti al suo interno

Attributi: nessuno

Vincoli: nessuno

- **Relationship cgCreaEs**

Ogni istanza di questa relationship lega un caregiver con le implementazioni di esercizio che ha creato

Attributi: nessuno

Vincoli: nessuno

- **Relationship implEs**

Ogni istanza di questa relationship lega una implementazione di esercizio con la tipologia di esercizio a cui appartiene

Attributi: nessuno

Vincoli: nessuno

- **Relationship percFisioRout**

Ogni istanza di questa relationship lega una routine con il percorso fisioterapico da cui sta prendendo gli esercizi

Attributi: nessuno

Vincoli: nessuno

- **Relationship esRout**

Ogni istanza di questa relationship lega una routine con gli esercizi che ne fanno parte.

Si noti che la stessa implementazione di esercizio può essere presente in più routine in quanto lo stesso esercizio assegnato al paziente può essere svolto più volte dal paziente stesso durante diverse sedute di terapia.

L'esecuzione dell'esercizio da parte del paziente viene quindi salvata sotto forma dell'esito che è legato alla coppia formata dall'implementazione dell'esercizio con la routine in quanto rappresenta se l'esercizio è stato correttamente eseguito dal paziente oppure no in una data sessione di svolgimento di esercizi.

L'esito può quindi poi essere consultato da parte dei caregiver professionisti per decidere se considerare concluso il periodo di tempo di apprendimento dell'esercizio e per constatare se questo deve essere ritenuto appreso oppure sospeso

attributo	dominio	molteplicità	descrizione
esito	StringL		Esito dell'esecuzione del particolare esercizio inserito in una specifica routine

Vincoli:

– [V.esRout.esLegali]

Un esercizio può essere inserito in una routine inerente ad un percorso terapeutico solamente se è presente nel percorso terapeutico stesso

$\forall r, es, pf$

$Routine(r) \wedge ImplementazioneEsercizio(es) \wedge PercorsoFisioterapico(pf) \wedge esRout(r, es) \wedge percFisioRout(r, pf) \rightarrow esPercFisio(pf, es)$

• Dominio Email

Il dominio è una stringa che rispetta gli standard di formattazione di un indirizzo email, ossia che rispetta la seguente espressione regolare: ¹:

$$\begin{aligned} & ^{[a-zA-Z0-9.!#\$\%&'*\+\backslash/=?\^_ \{\} \sim -]+@[a-zA-Z0-9] \\ & (?:[a-zA-Z0-9-]\{0,61\}[a-zA-Z0-9])?(?:\.[a-zA-Z0-9] \\ & (?:[a-zA-Z0-9-]\{0,61\}[a-zA-Z0-9])?)*\$ \end{aligned}$$

• Dominio Password

Il dominio è una stringa di al più 255 caratteri che rappresenta una password cifrata. La password inserita prima della cifratura dovrà rispettare gli standard di formattazione di una password, ossia dovrà rispettare la seguente espressione regolare: ²:

$$\begin{aligned} & ^{(?=.*\backslash d)(?=. *[a-z])(?=. *[A-Z])(?=. *[^a-zA-Z0-9]) \\ & (?!.*\backslash s).\{8,16\}\$ \end{aligned}$$

• Dominio CodiceFiscale

Il dominio è una stringa che rispetta gli standard di formattazione di un codice fiscale, ossia che rispetta la seguente espressione regolare: ³

$$[A-Z]\{6\}[\backslash d]\{2\}[A-Z]\{1\}[\backslash d]\{2\}[A-Z0-9]\{4\}[A-Z]\{1\}$$

¹L'espressione regolare per la validazione dell'indirizzo e-mail è stata presa da: <https://html.spec.whatwg.org/multipage/input.html> nella sezione "4.10.5.1.5 Email state (type=email)"

²L'espressione regolare per la validazione della password, secondo le regole descritte nel Glossario, è stata scritta partendo dalle regole qui presenti: <https://www.ocpssoft.org/tutorials/regular-expressions/password-regular-expression/>

³L'espressione regolare per la validazione dei codici fiscali è stata presa da: <https://userapps.support.sap.com/sap/support/knowledge/en/2413962>

- **Dominio IntegerGZ**

Il dominio è dato dal sottoinsieme del dominio integer formato dai valori > 0

- **Dominio Ruolo**

Il dominio è un dominio enumerativo contenente i seguenti valori:

- professionale
- non professionale

- **Dominio TipoNotifica**

Il dominio è un dominio enumerativo contenente i seguenti valori:

- richiesta
- conferma

- **Dominio NomeRoutine**

Il dominio è una stringa con il seguente formato:

ID_percorso_fisioterapico-istExec_routine

- **Domino StringS**

Il dominio è dato dalle stringhe di al più 500 caratteri

- **Dominio StringM**

Il dominio è dato dalle stringhe di al più 1000 caratteri

- **Dominio StringL**

Il dominio è dato dalle stringhe di al più 2000 caratteri

6.4 Tabelle della Base Dati

In questa sezione viene riportato lo schema relazionale della base dati e i relativi vincoli prodotti a partire dallo schema ER ristrutturato.

Si noti che gli **attributi di chiave primaria** nelle tabelle sono **sottolineati** e che gli **attributi i cui valori possono essere NULL** sono **contrassegnati con ***

Relazione Utente derivante da entità	
Attributi	Domini
<u>cf</u>	CodiceFiscale
nome	StringS
cognome	StringS

Vincoli (foreign key, inclusione, altra chiave, di ennuola, di dominio): nessuno

Relazione Caregiver derivante da entità	
Attributi	Domini
<u>utente</u>	CodiceFiscale
email	Email
password	Password
ruolo	Ruolo

Vincoli (foreign key, inclusione, altra chiave, di ennuola, di dominio):

fk: utente refers Utente(cf)

unique: email

La relazione accorpa le relazioni che implementano la relationship:

utCg

Relazione Bambino derivante da entità	
Attributi	Domini
<u>utente</u>	CodiceFiscale
nascita	date
eta	IntegerGZ
percFisio	IntegerGZ
tutore	CodiceFiscale

Vincoli (foreign key, inclusione, altra chiave, di ennuola, di dominio):

fk: utente refers Utente(id)

fk: percFisio refers PercorsoFisioterpico(id)

fk: tutore refers Caregiver(utente)

inclusione: (utente) \subseteq cgBambino(bambino)

inclusione: (utente) \subseteq bDisabilita(bambino)

La relazione accorpa le relazioni che implementano la relationship:

utBamb, bPercFisio, tutBambino

Relazione Notifica derivante da entità	
Attributi	Domini
<u>id</u>	IntegerGZ
cfMittente	CodiceFiscale
cfB	CodiceFiscale
cfDest	CodiceFiscale
tipo	TipoNotifica

Vincoli (foreign key, inclusione, altra chiave, di ennuola, di dominio):

fk: cfMittente refers Caregiver(utente)

fk: cfB refers Bambino(utente)

fk: cfDest refers Caregiver(utente)

La relazione accorpa le relazioni che implementano la relationship:

mittCg, notifB, destCg

Relazione Disabilita derivante da entità	
Attributi	Domini
<u>nome</u>	StringS
descrizione	StringL

Vincoli (foreign key, inclusione, altra chiave, di ennuola, di dominio): nessuno

Relazione PercorsoFisioterapico derivante da entità	
Attributi	Domini
<u>id</u>	IntegerGZ
inserimento	date

Vincoli (foreign key, inclusione, altra chiave, di ennuola, di dominio):

inclusione: (id) \subseteq Bambino(percFisio)

inclusione: (id) \subseteq cgPercFisio(percFisio)

Relazione ImplementazioneEsercizio derivante da entità	
Attributi	Domini
<u>id</u>	IntegerGZ
inserimento	date
apprendimento*	date
sospensione*	date
eseguito	boolean
respAcc*	StringM
ripasso*	date
tipologia	StringS
percFisio	IntegerGZ
abilita	IntegerGZ

Vincoli (foreign key, inclusione, altra chiave, di ennupla, di dominio):

fk: tipologia refers TipologiaEsercizio(nome)

fk: percFisio refers PercorsoFisioterapico(id)

fk: abilita refers Abilita(cod)

notNull: tipologia not NULL

notNull: percFisio not NULL

notNull: abilita not NULL

unique: (tipologia, percFisio, abilita)

ennupla: apprendimento > inserimento

ennupla: sospensione > inserimento

ennupla: apprendimento \vee inserimento

ennupla: terminato = True \rightarrow apprendimento \vee inserimento

ennupla: ripasso > apprendimento

inclusione: (id) \subseteq cgCreaEs(implEs)

La relazione accorpa le relazioni che implementano la relationship:

implEs, esAb, esPercFisio

Relazione Abilita derivante da entità	
Attributi	Domini
<u>cod</u>	IntegerGZ
descrizione	StringM

Vincoli (foreign key, inclusione, altra chiave, di ennupla, di dominio): nessuno

Relazione TipologiaEsercizio derivante da entità	
Attributi	Domini
<u>nome</u>	StringS
descrizione	StringL
modExec	StringL
risAtt	StringL

Vincoli (foreign key, inclusione, altra chiave, di ennupla, di dominio): nessuno

Relazione Routine derivante da entità	
Attributi	Domini
<u>nome</u>	NomeRoutine
istExec	timestamp
percFisio	IntegerGZ

Vincoli (foreign key, inclusione, altra chiave, di ennupla, di dominio):

fk: percFisio refers PercorsoFisioterapico(id)

unique: (istExec, percFisio)

La relazione accorpa le relazioni che implementano la relationship:

percFisioRout

Relazione cgBambino derivante da relationship	
Attributi	Domini
<u>caregiver</u>	CodiceFiscale
<u>bambino</u>	CodiceFiscale

Vincoli (foreign key, inclusione, altra chiave, di ennupla, di dominio):

fk: caregiver refers Caregiver(utente)

fk: bambino refers Bambino(utente)

Relazione bDisabilita derivante da relationship	
Attributi	Domini
<u>bambino</u>	CodiceFiscale
<u>disabilita</u>	StringS

Vincoli (foreign key, inclusione, altra chiave, di ennupla, di dominio):

fk: bambino refers Bambino(utente)

fk: disabilita refers Disabilita(nome)

Relazione esDisabilita derivante da relationship	
Attributi	Domini
<u>disabilita</u>	StringS
<u>tipo</u>	StringS

Vincoli (foreign key, inclusione, altra chiave, di ennupla, di dominio):

fk: disabilita refers Disabilita(nome)

fk: tipo refers TipologiaEsercizio(nome)

Relazione cgPercFisio derivante da relationship	
Attributi	Domini
caregiver	CodiceFiscale
<u>percFisio</u>	IntegerGZ

Vincoli (foreign key, inclusione, altra chiave, di ennuola, di dominio):

fk: cg refers Caregiver(utente)

fk: percFisio refers PercorsoFisioterapico(id)

Relazione esRout derivante da relationship	
Attributi	Domini
<u>esImpl</u>	IntegerGZ
<u>routine</u>	NomeRoutine
esito	StringL

Vincoli (foreign key, inclusione, altra chiave, di ennuola, di dominio):

fk: esImpl refers ImplementazioneEsercizio(id)

fk: routine refers Routine(nome)

Relazione cgCreaEs derivante da relationship	
Attributi	Domini
<u>caregiver</u>	CodiceFiscale
<u>esImpl</u>	IntegerGZ

Vincoli (foreign key, inclusione, altra chiave, di ennuola, di dominio):

fk: caregiver refers Caregiver(utente)

fk: esImpl refers ImplementazioneEsercizio(id)

6.4.1 Ulteriori Vincoli Esterni

Vengono ora riportati i trigger per i vincoli esterni nella base dati:

- **Trigger per [V.Utente.disgiunta]**

Se un utente è un Caregiver non può essere un Bambino

Operazioni: inserimento o modifica in Caregiver o Bambino

Istante di invocazione: prima dell'operazione intercettata

Funzione:

1. Sia

```
isError = FALSE
```

2. Sia

```
ne
```

l'ennupla che si sta inserendo oppure l'ennupla risultato della modifica;

3. Se si sta inserendo o modificando una ennupla in Caregiver:

```
isError := exists(select * from
                  Bambino b
                  where b.utente = ne.utente);
```

4. Altrimenti (inserimento/modifica di una ennupla in Bambino):

```
isError := exists(select * from
                  Caregiver cg
                  where cg.utente = ne.utente);
```

5. Se

```
isError = TRUE
```

blocca l'operazione;

6. Altrimenti permetti l'operazione.

- **Trigger per [V.Utente.completa]**

Ogni utente deve essere o un Caregiver o un Bambino

Operazioni: inserimento o modifica in Caregiver o Bambino

Istante di invocazione: prima dell'operazione intercettata

Funzione:

1. Sia

```
isOk = FALSE
```

2. Sia

```
ne
```

l'ennupla che si sta inserendo oppure l'ennupla risultato della modifica;

3.

```
isOk := exists(select * from
  Bambino b
  where b.utente = ne.utente)
or
exists(select * from
  Caregiver cg
  where cg.utente = ne.utente);
```

4. Se

```
isOk = FALSE
```

blocca l'operazione;

5. Altrimenti permetti l'operazione.

- **Trigger per [V.cgPercFisio.cgSoloProf]**

I percorsi fisioterapici per i pazienti possono essere inseriti solo dai caregiver il cui ruolo è "professionale"

Operazioni: inserimento in PercorsoFisioterapico da parte di un dato Caregiver

Istante di invocazione: prima dell'operazione intercettata

Funzione:

1. Sia

```
isOk = FALSE
```

2. Sia

```
ne
```

l'ennupla che si sta inserendo;

3.

```
isOk := 'professionale' == select ruolo from  
Caregiver cg, cgPercFisio cgpf  
where  
cgpf.caregiver = cg.utente and  
cgpf.percFisio = ne.id;
```

4. Se

```
isOk = FALSE
```

blocca l'operazione;

5. Altrimenti permetti l'operazione.

- **Trigger per [V.esRout.esLegali]**

Affinché un esercizio possa essere inserito in una routine di un percorso fisioterapico questo deve essere presente anche nel dato percorso fisioterapico **Operazioni:** inserimento in esRout

Istante di invocazione: prima dell'operazione intercettata

Funzione:

1. Sia

```
isOk = FALSE
```

2. Sia

```
ne
```

l'ennupla che si sta inserendo;

```
3.      isOk := exists(select * from
                        ImplementazioneEsercizio ie,
                        PercorsoFisioterapico pf,
                        Routine r
                        where
                        esRout.esImpl = ie.id and
                        esRout.routine = r.nome and
                        r.percFisio = pf.id and
                        ie.percFisio = pf.id);
```

4. Se

```
isOk = FALSE
```

blocca l'operazione;

5. Altrimenti permetti l'operazione.