

SDD *version 2.0*



Documento di System Design

Gruppo 7

| | | |
|----------------------|----------------------------|-------------------|
| Team Manager: | Roberto Di Russo | 0510201086 |
| Componenti: | Giovanni Maggiolini | 0510201080 |
| | Umberto Annunziata | 0510201230 |
| | Francesco Nicolao | 0510200141 |
| | Claudio Gargiulo | 0510200990 |
| | Vincenzo Iuliano | 0510201428 |

Revision History

| Data | Versione | Descrizione | Autore |
|-------------|-----------------|---|-------------------------------|
| 22/04/08 | SDD 0.1 raw | Introduzione e definizione System Goals | Tutti i componenti del gruppo |
| 23/04/08 | SDD 0.2 raw | Decomposizione del sistema in sottosistemi Hardware/Software Mapping | Tutti i componenti del gruppo |
| 24/04/08 | SDD 0.3 raw | Gestione dei Dati Persistenti Controllo degli accessi e sicurezza Gestione del controllo globale Boundary conditions | Tutti i componenti del gruppo |
| 28/04/08 | SDD 1.0 | Servizi offerti dai sottosistemi Glossario Ulteriore revisione | Tutti i componenti del gruppo |
| 30/04/08 | SDD 1.1 | Inserimento dello schema ER nella Gestione Dati Persistenti Ulteriore revisione | Tutti i componenti del gruppo |
| 03/06/08 | SDD 2.0 | Correzione e revisione finale | Tutti i componenti del gruppo |

SDD

Progetto di un sistema software per la gestione dei posti letto presso una clinica privata

| | |
|---|----|
| 1. Introduzione | 4 |
| 1.1 Scopo del Sistema | 4 |
| 1.2 Design Goals..... | 5 |
| 1.2.1 Performance Criteria..... | 5 |
| 1.2.2 Dependability Criteria | 6 |
| 1.2.3 Maintenance Criteria | 6 |
| 1.2.4 User Criteria..... | 7 |
| 1.3 Definizioni, Acronimi e Abbreviazioni | 7 |
| 1.4 Riferimenti | 7 |
| 2. Architettura del Sistema Attuale | 8 |
| 3. Architettura del Sistema Proposto | 9 |
| 3.1 Overview | 9 |
| 3.2 Subsystem decomposition | 10 |
| 3.3 Hardware/Software mapping | 21 |
| 3.4 Gestione dei Dati Persistenti | 23 |
| 3.5 Controllo degli accessi e sicurezza | 30 |
| 3.5.1 Controllo degli accessi | 30 |
| 3.5.2 Sicurezza | 33 |
| 3.6 Gestione del controllo globale | 34 |
| 3.7 Boundary conditions | 34 |
| 3.7.1 Configuration | 34 |
| 3.7.2 Start up – Shut down use cases. | 36 |
| 3.7.3 Exception handling | 39 |
| 4. Servizi dei Sottosistemi | 40 |
| 5. Glossario..... | 46 |

1. INTRODUZIONE

1.1 Scopo del sistema

L'obiettivo del software proposto è la realizzazione di un sistema di supporto ad una clinica privata, in grado di rendere più agevole la gestione delle degenze (per maggiori dettagli vedi EasyClinique RAD 4.0).

Il software EasyClinique presenta importanti vincoli da rispettare nella fase di progettazione del sistema, ad esempio a livello di prestazioni, di architettura e di portabilità.

Per evitare impatti “traumatici” con l'utente finale, inoltre, il software deve presentare delle interfacce user-friendly: intuitive e dal facile utilizzo.

L'obiettivo del documento di SDD, quindi, è quello di far fronte ai problemi ed ai vincoli del sistema proposto, fornendo un'architettura competitiva e vantaggiosa.

In conclusione, quello che si cerca di fare è ridurre la complessità per rendere il sistema più flessibile ed efficiente.

1.2 Design Goals

Identificare gli obiettivi di design, o design goals, è fondamentale, in quanto su di essi si baseranno le scelte successive, specialmente in fase di implementazione.

Il sistema proposto cerca di arrivare a dei compromessi ragionevoli (trade-offs) tra tutti gli obiettivi di design, dato che alcuni di loro possono essere in contrasto.

Vengono riportati di seguito nel particolare i suddetti obiettivi.

1. Criteri di performance (Performance Criteria)

- *Tempo di risposta:* in seguito ad una richiesta dell'utente al sistema la risposta viene fornita in meno di 20 secondi
- *Throughput:* il sistema non ha grossi vincoli nel numero di task che possono essere portati a termine in un'unità di tempo fissata.
- *Memoria:* la memoria di massa richiesta dal sistema è essenzialmente quella necessaria al mantenimento del database contenente le informazioni riguardanti medici, pazienti, cartelle cliniche, utenti, etc. Per quanto riguarda, invece, la memoria centrale si consiglia, per un'esecuzione ottimale, una capacità di almeno 512 Megabytes.

2. Criteri di affidabilità (Dependability Criteria)

- *Fault tolerance*: in presenza di condizioni di errore, il sistema non può continuare ad operare: verrà mostrata, quindi, una notifica all'utente ed eventualmente un consiglio su come proseguire.
- *Robustezza*: la maggior parte degli input forniti al sistema verranno “filtrati” parzialmente: le operazioni che includono una ricerca si limiteranno a mostrare esito negativo; per quanto riguarda, invece, gli inserimenti non sono previsti controlli semantici sulle stringhe date in input dall'utente (ad esempio, è possibile inserire un nome di un Paziente del tipo “xxklnfkò”)
- *Disponibilità*: il sistema sarà attivo per 24 ore al giorno, tutti i giorni. Non vengono, però, trascurati gli aspetti relativi a situazioni di malfunzionamenti, meglio esplicitate nella sezione “Condizioni limite”.
- *Sicurezza*: il sistema è relativamente sicuro: non è necessaria una connessione ad Internet, ma il database è distribuito sulla rete locale alla clinica. Ogni accesso è regolato da una coppia userID/password che rispetta una politica di permessi.
- *Affidabilità*: in caso di problemi alla connessione con il database le operazioni devono essere disabilitate e l'utente deve essere portato a conoscenza delle limitazioni.

3. Criteri di mantenimento (Maintenance criteria)

- *Adattabilità*: il sistema può essere eventualmente utilizzato, oltre che per lo specifico dominio applicativo, anche per altre applicazioni che gestiscono dati in ambito medico-sanitario.
- *Leggibilità*: la leggibilità del codice sarà garantita da uno stile di formattazione uniforme e da un uso di commenti laddove il codice risulti articolato.
- *Portabilità*: il prodotto software in questione verrà realizzato attraverso una serie di moduli scritti nel linguaggio Java, mentre il database si baserà su un DBMS MySQL. Dunque, è garantito l'adattamento a diverse piattaforme.
- *Estendibilità*: la suddivisione in moduli permetterà di aggiungere nuove funzionalità al sistema per adeguarlo, magari, a nuove esigenze future.

4. *Utente finale* (End User Criteria)

- *Usabilità*: è prevista la stesura di un manuale utente, in modo che quest'ultimo possa contare su una struttura di supporto nell'uso del sistema. Inoltre il sistema si baserà su una estrema intuitività delle interfacce.
- *Utilità*: il sistema risponderà in modo esaustivo a quelli che sono i “desiderata” esplicitati dall'utente in fase di raccolta.

1.3 Definizioni, Acronimi e Abbreviazioni

RAD: Requirements Analysis Document

SDD: System Design Document

DBMS : Database Management System

JDBC : Java Database Connectivity

SQL : Structured Query Language

E-R: Entity-Relationship

1.4 Riferimenti

- RAD version 4.0 di EasyClinique

2. Architettura del Sistema Attuale

L'architettura attualmente proposta non va a sostituire nessuna architettura già esistente.

Verrà utilizzato lo stile architetturale “Three Tiers”, cioè è prevista la scomposizione del sistema in sottosistemi e, per ognuno di essi, una suddivisione in tre livelli (o strati):

- InterfaceLayer, che include la gestione dell'interfaccia utente
- ApplicationLogicLayer, che include la gestione degli oggetti entity e control
- StorageLayer, che effettua la memorizzazione ed il recupero degli oggetti persistenti

Nel seguito viene riportato uno schema generale dello stile “Three Tiers” applicato all'intero sistema



Una migliore descrizione del sistema proposto è fornita al punto 3.

3. Architettura del Sistema Proposto

3.1 Overview

Come già anticipato nel punto 2, l'architettura del sistema proposto prevede una decomposizione in tre livelli logici, conosciuta anche con il nome di architettura "Three Tiers".

Lo scopo della suddivisione in livelli è quello di facilitare l'implementazione e ridurre la complessità generale del sistema .

Ovviamente, la suddivisione proposta è stata definita tenendo conto dei design goals indicati al punto 1.2, dei canoni di basso accoppiamento (LOW COUPLING) tra i sottoinsiemi individuati ed alta coesione (HIGH COHESION) all'interno di ognuno di questi ultimi.

Molta attenzione è stata dedicata anche a quanto già definita nella versione 4.0 del RAD: è facile notare, infatti, come i sottosistemi siano riconducibili agli use cases.

La scelta dell'architettura "Three Tiers" è facilmente giustificabile.

Il nucleo del sistema, infatti, è costituito dal database che memorizza i dati di pazienti, cartelle cliniche, personale medico-sanitario, reparti e stanze, e account.

Il database presenterà all'esterno diverse viste, in base all'attore che accede attraverso l'InterfaceLayer, lasciando all'ApplicationLogicLayer il compito di raccogliere i dati forniti in input dagli utenti e permettere di effettuare operazioni sugli oggetti del sistema.

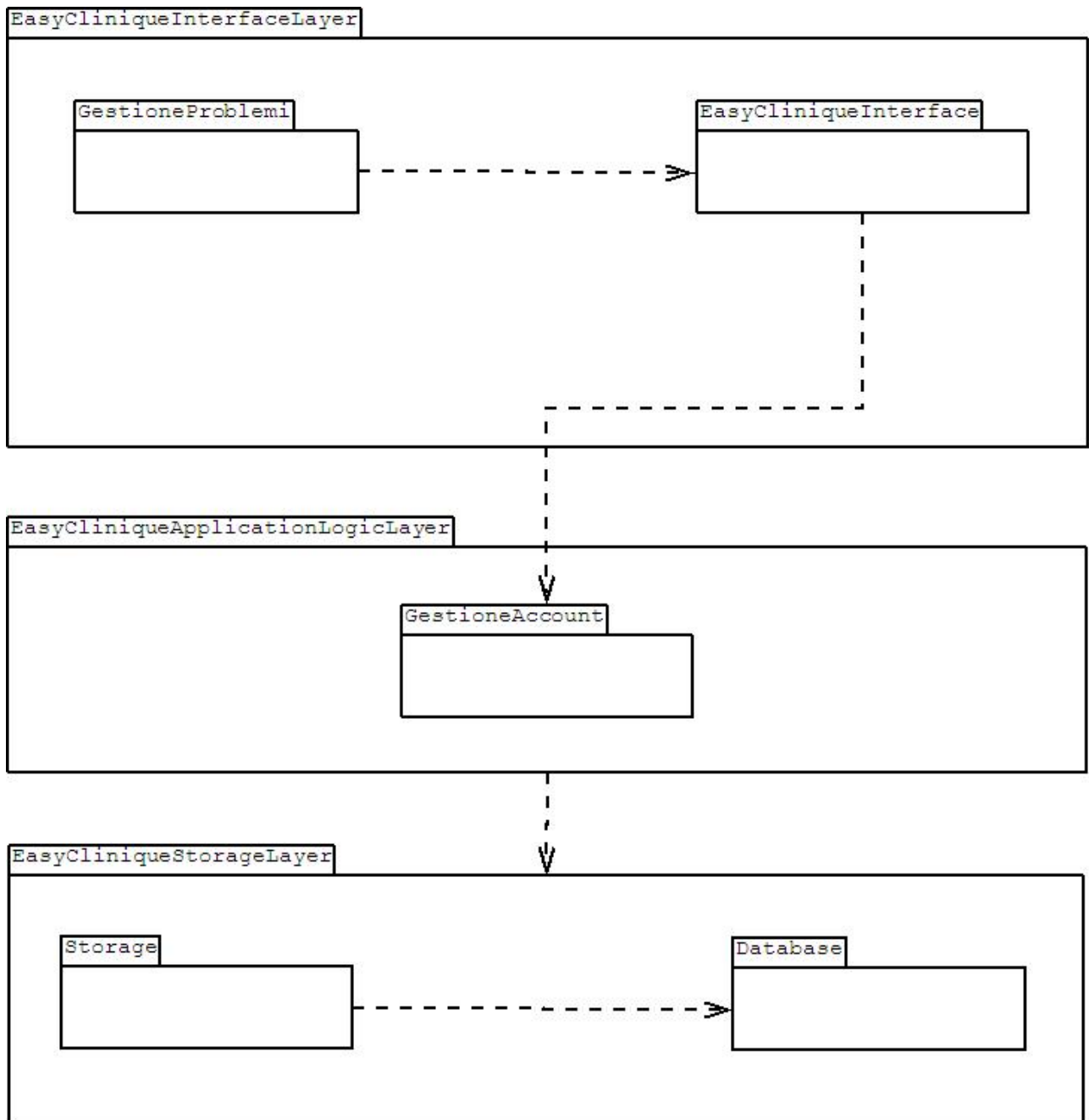
I sottosistemi fondamentali individuati sono quattro:

- SottosistemaAmministratore
- SottosistemaDirigente
- SottosistemaResponsabile
- SottosistemaMedicoParamedico

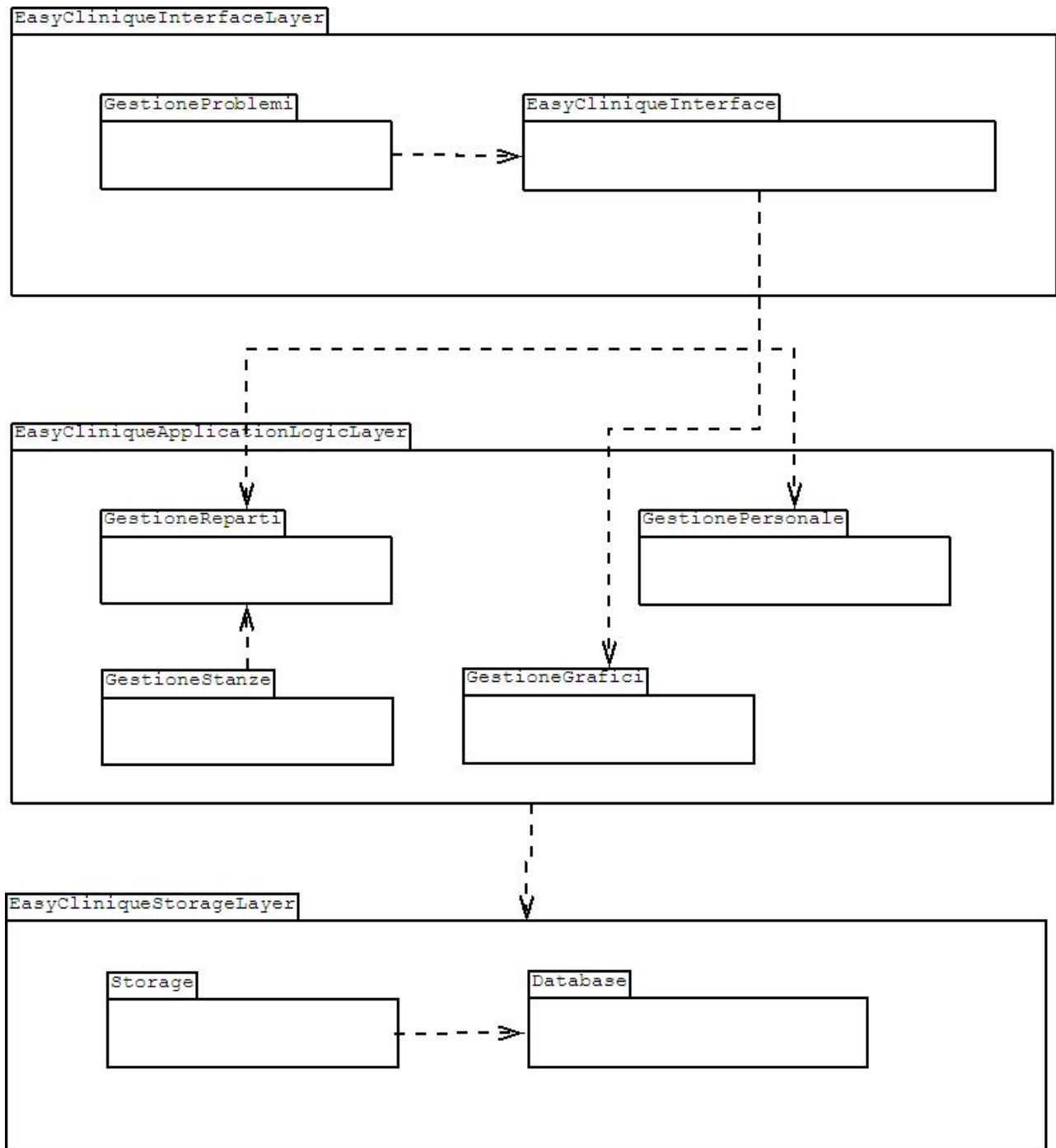
3.2 Decomposizione in sottosistemi

Di seguito vengono riportati graficamente i quattro sottosistemi individuati.

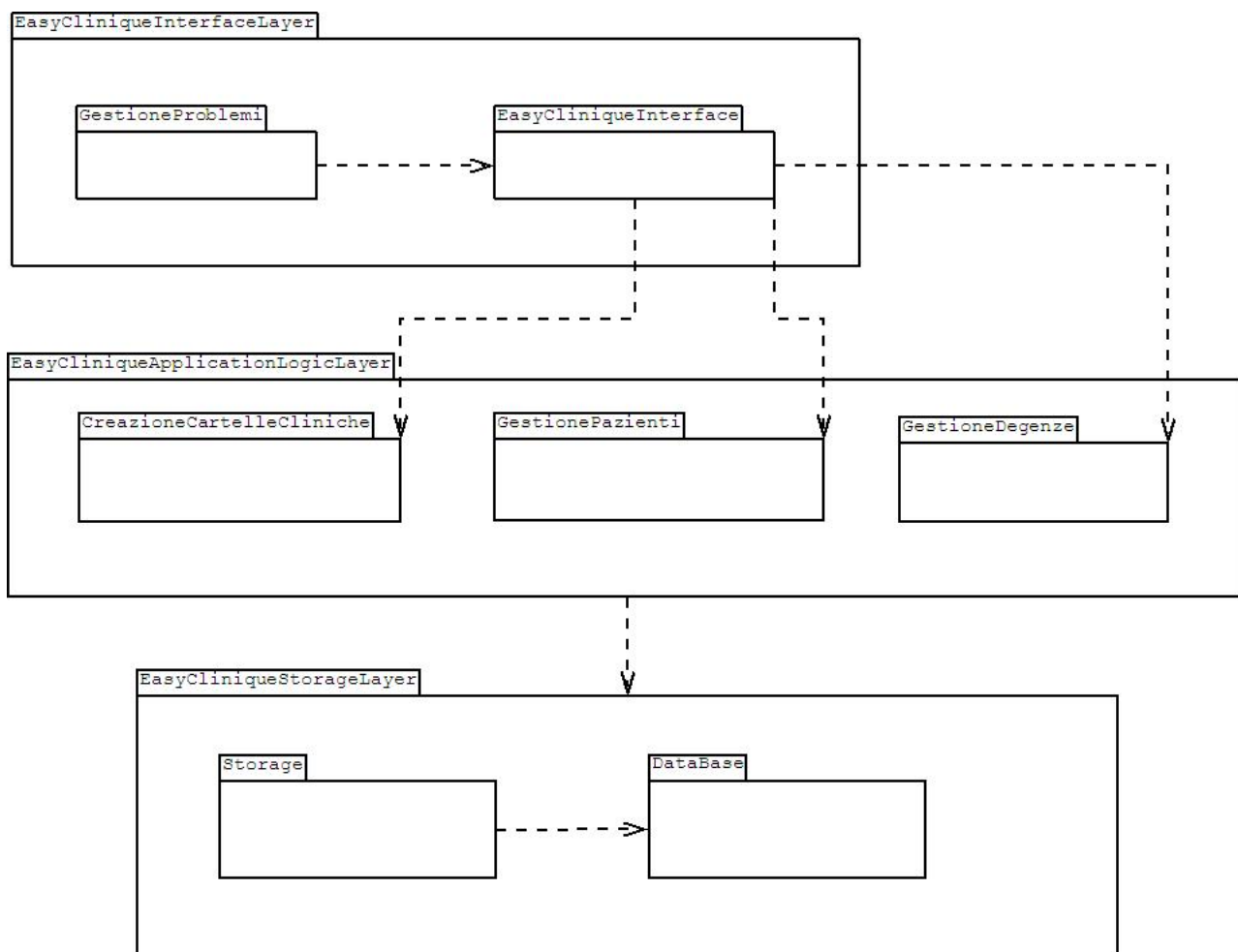
SottosistemaAmministratore



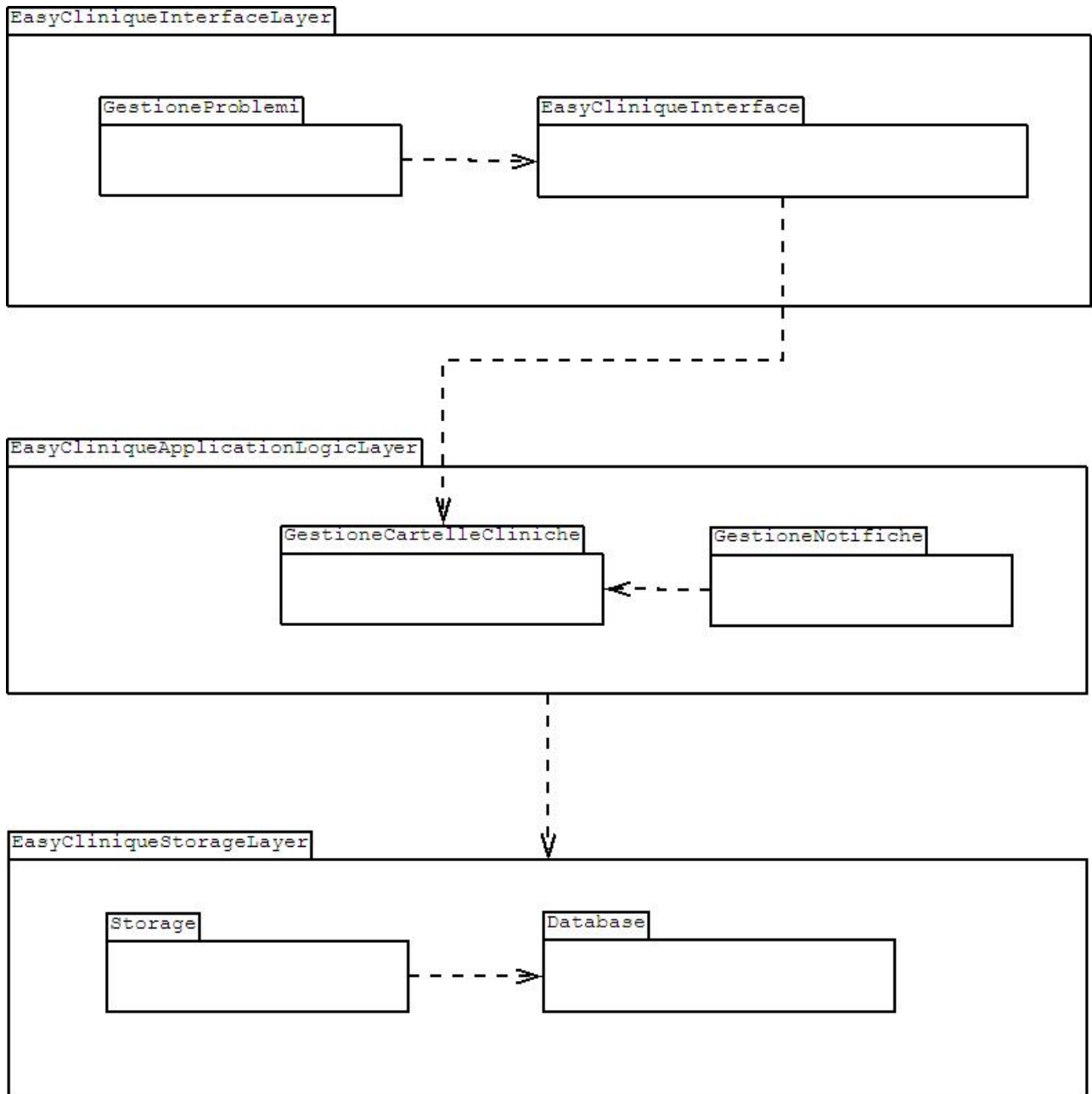
SottosistemaDirigente



SottosistemaResponsabile



SottosistemaMedicoParamedico



Tutti i sottosistemi hanno “in comune” i livelli EasyCliniqueInterface ed EasyCliniqueStorageLayer, di seguito descritti.

EasyCliniqueInterfaceLayer:

| | |
|------------------------------|--|
| EasyCliniqueInterface | Interfaccia grafica con la quale l'utente interagisce e tramite la quale inoltra richieste al sistema e visualizza risultati |
| GestioneProblemi | Modulo che si occupa semplicemente di offrire l'help on-line |

EasyCliniqueStorageLayer:

| | |
|-----------------|--|
| Storage | Modulo che permette di ridurre l'accoppiamento fra il modulo relativo al DataBase e quello dell' EasyCliniqueApplicationLogicLayer. L'interfaccia che fornisce permette di richiamare le operazioni e le interrogazioni direttamente sul database |
| DataBase | Modulo che inoltra le richieste di recupero dati direttamente al database fisico e ne recupera i dati, per poi restituirli all'EasyCliniqueApplicationLogicLayer |

Di seguito viene riportata una breve descrizione, per ogni sottosistema, dell'EasyCliniqueApplicationLogicLayer

SottosistemaAmministratore:

| | |
|------------------------|--|
| GestioneAccount | Modulo che si occupa di gestire le operazioni di registrazione, modifica, cancellazione e ricerca degli account. |
|------------------------|--|

SottosistemaResponsabile:

| | |
|----------------------------------|--|
| CreazioneCartelleCliniche | Modulo che gestisce le operazioni di inserimento delle cartelle cliniche |
| GestionePazienti | Modulo che gestisce le operazioni di inserimento e modifica dei dati anagrafici dei pazienti |
| GestioneDegenze | Modulo che gestisce le operazioni di dimissione dei pazienti, assegnazioni posti letto, e spostamenti di pazienti da un reparto ad un altro. |

SottosistemaDirigente:

| | |
|--------------------------|---|
| GestioneReparti | Modulo che si occupa di gestire le operazioni di inserimento, cancellazione e ricerca dei reparti |
| GestioneStanze | Modulo che si occupa di gestire le operazioni di inserimento, modifica e ricerca delle stanze |
| GestioneGrafici | Modulo che si occupa di gestire la visualizzazione dei grafici statistici |
| GestionePersonale | Modulo che si occupa di gestire le operazioni di inserimento, modifica, cancellazione e ricerca di un membro del personale medico-sanitario |

SottosistemaMedicoParamedico:

| | |
|---------------------------------|--|
| GestioneCartelleCliniche | Modulo che si occupa di gestire le operazioni di modifica, ricerca e stampa delle cartelle cliniche. |
| GestioneNotifiche | Modulo che si occupa di gestire l'inoltro delle notifiche di dimissione di un paziente. |

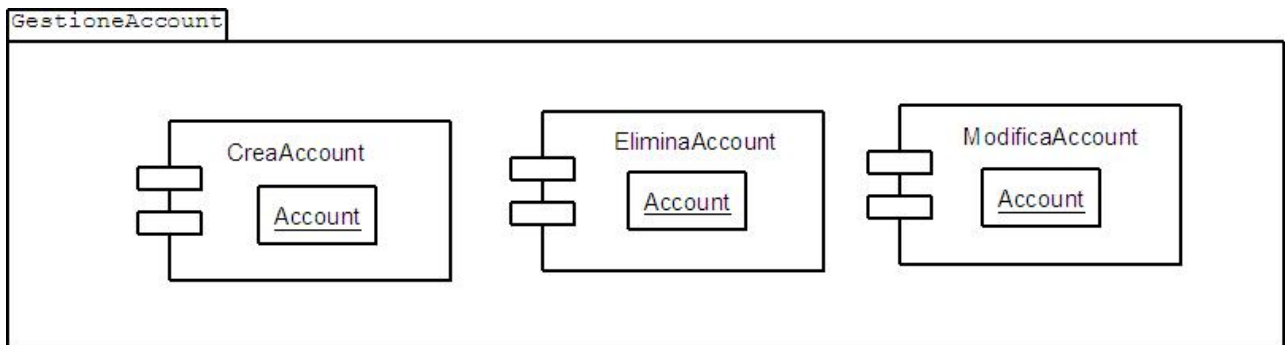
NOTA:

E' stato ritenuto opportuno fondere in un unico sottosistema i due sottosistemi SottosistemaMedico e SottosistemaParamedico, in quanto tra loro molto simili.

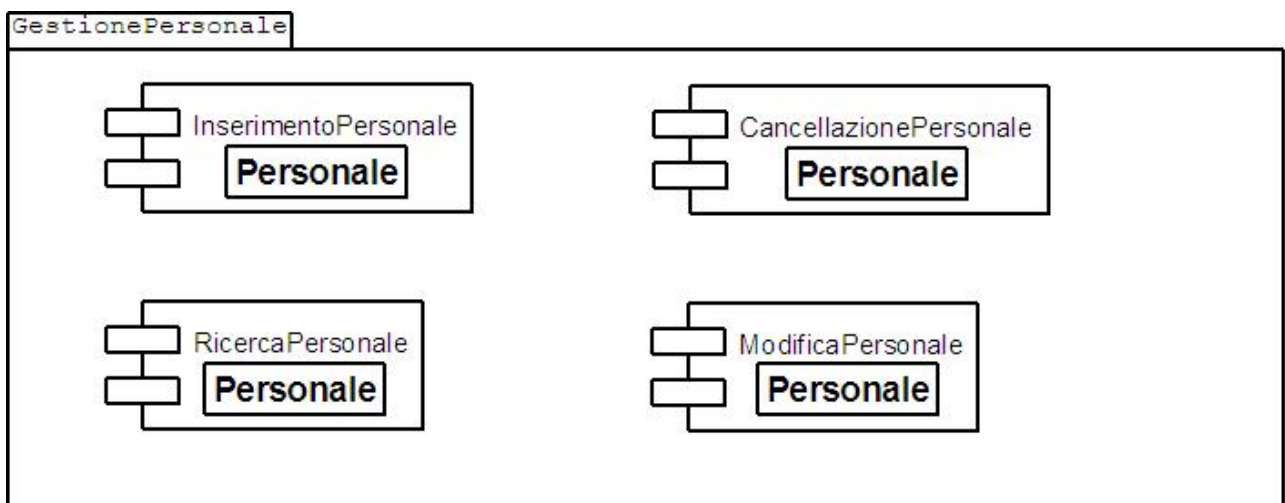
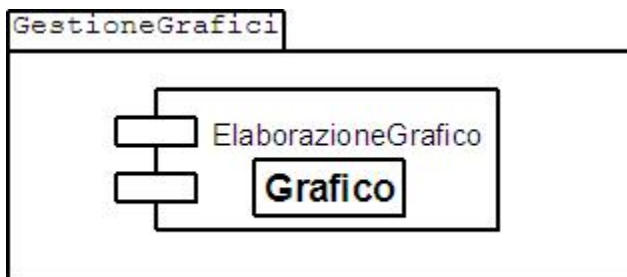
La distinzione tra le funzionalità accessibili da un Medico e quelle accessibili da un Paramedico viene specificata nella sezione "3.5 Controllo accessi e sicurezza"

Di seguito sono specificate, per ogni modulo, le componenti da cui è formato, che rappresentano nel dettaglio le operazioni di cui si occupa, e i dati su cui operano. Tutti moduli specificati fanno parte dell'ApplicationLogicLayer.

SottosistemaAmministratore



SottosistemaDirigente



GestioneReparti



GestioneStanze

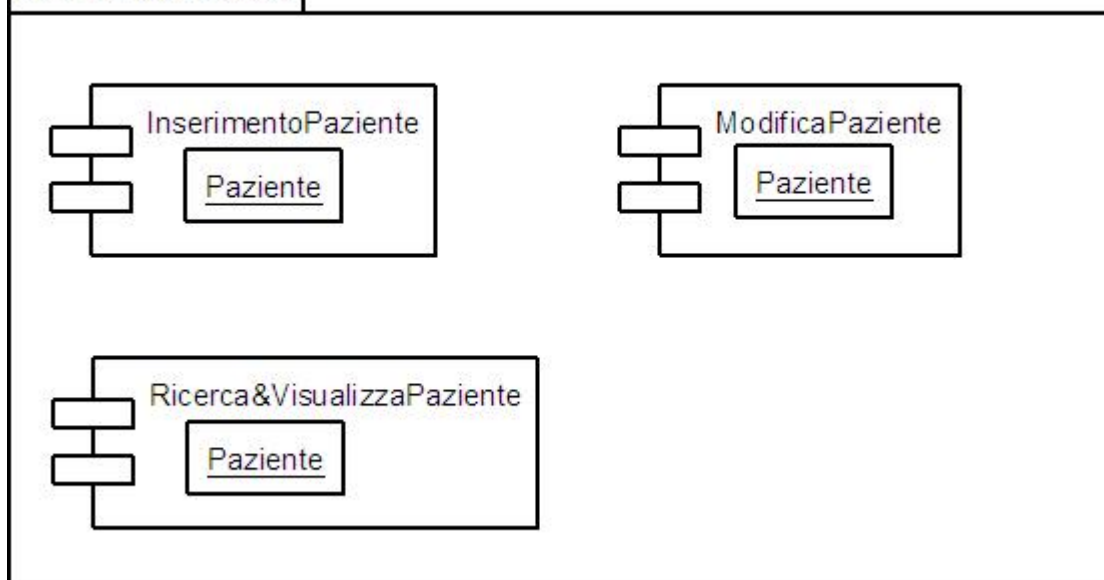


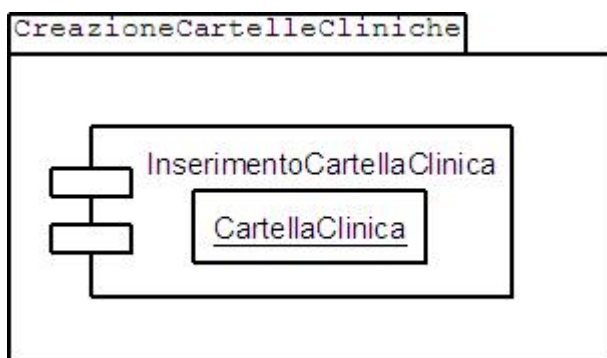
SottosistemaResponsabile

GestioneDegenze

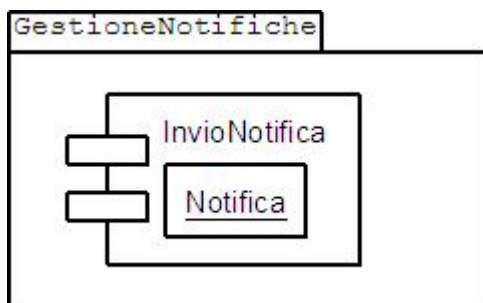
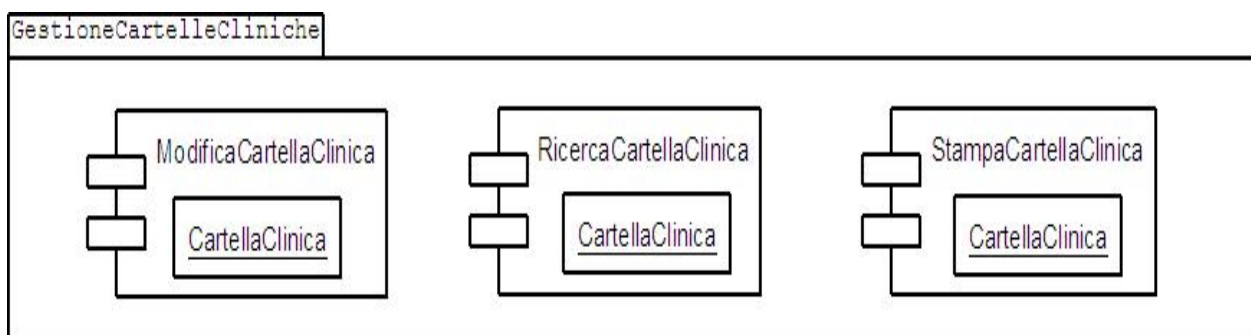


GestionePazienti





SottosistemaMedicoParamedico



3.3 Hardware/Software Mapping

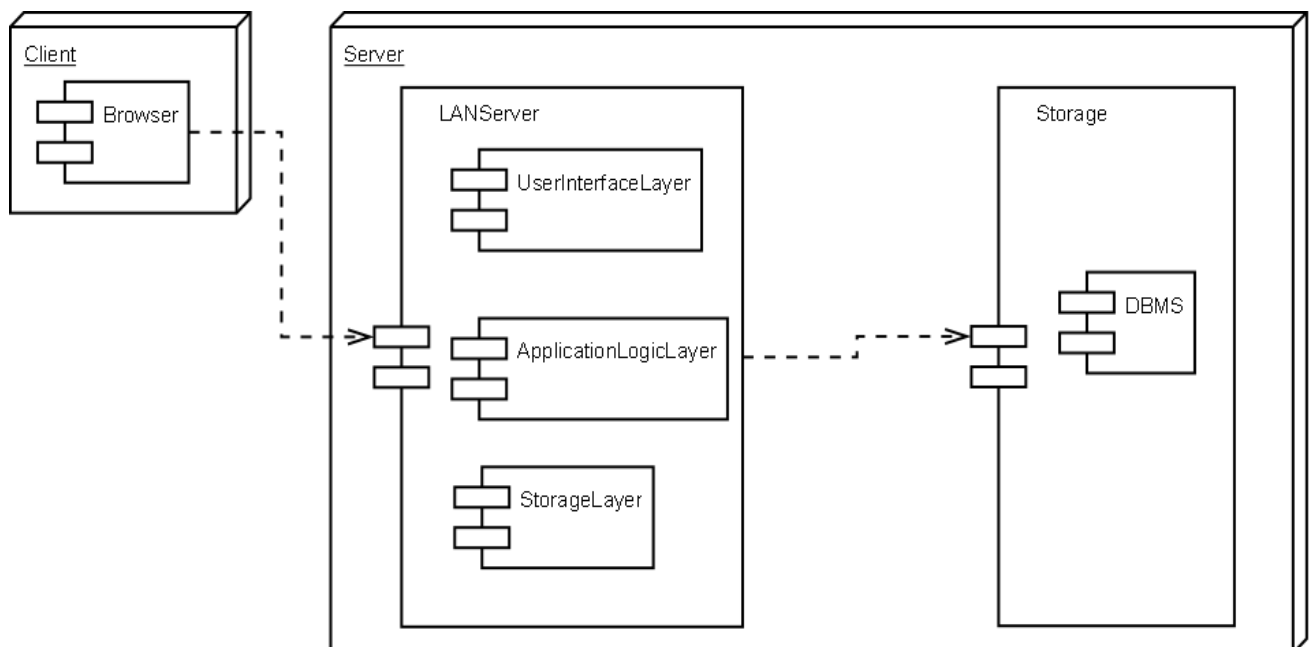
Il sistema software EasyClinique è distribuito su due nodi, Client e Server.

Client è la macchina usata da un qualsiasi utente che funziona semplicemente come un terminale, mentre l'intero sistema è residente su Server.

Ovviamente, Client è una generalizzazione: all'interno della clinica sono a disposizione una serie di macchine, tutte equiparate a cui possono accedere gli utenti registrati.

Inoltre, si suppone che Amministratore, DirigenteSanitario e ResponsabileAccettazione siano dotati nei propri uffici di macchine personali, ma che non vanno oltre il livello Client.

Per facilitare la comprensione, si riporta in seguito lo schema proposto.



UserInterfaceLayer

Lo UserInterfaceLayer è residente su Server ed i suoi contenuti sono visualizzati dall'utente tramite la componente Browser, presente, appunto, sul nodo Client.

ApplicationLogicLayer

L'ApplicationLogicLayer è responsabile della logica di controllo del sistema.

Esso funziona semplicemente come uno strato intermedio tra l'interfaccia utente (UserInterfaceLayer) e il livello sottostante (StorageLayer).

StorageLayer

Lo StorageLayer è responsabile dell'interfacciamento con il database.

Consiste di classi e interfacce scritte in Java che utilizzano la JDBC per comunicare con il DBMS sottostante, ossia MySQL.

DBMS

La gestione dei dati persistenti viene affidata ad un database relazionale, ovvero il DBMS MySQL, nella versione 5.0.

Per interagire con esso viene utilizzata l'interfaccia JDBC di Java.

Si noti che per eseguire il software è necessaria una Java Virtual Machine in grado di supportare Java 5.

3.4 Gestione dei dati persistenti

I dati persistenti rappresentano i dati che devono sopravvivere alla singola esecuzione del sistema. Vengono scelti, quindi, gli oggetti che devono, appunto, persistere quando il sistema viene chiuso.

Per chiusura del sistema si intende sia la normale chiusura (o shutdown) che la chiusura forzata (crash), dovuta, ad esempio, ad errori o malfunzionamenti.

Tuttavia, come specificato in altre sezioni, più che alla chiusura del sistema, i dati devono persistere alla chiusura di una sessione da parte dell'utente, in quanto non è previsto lo spegnimento del sistema stesso.

L'individuazione dei dati persistenti che il sistema deve gestire risulta alquanto semplificata se si va a consultare la documentazione prodotta in fase di analisi dei requisiti.

Il sistema **EasyClinique** presenta i seguenti oggetti da memorizzare in modo persistente:

- DatiPaziente
- Personale
- CartellaClinica
- Stanza
- Reparto
- Account
- Degenza

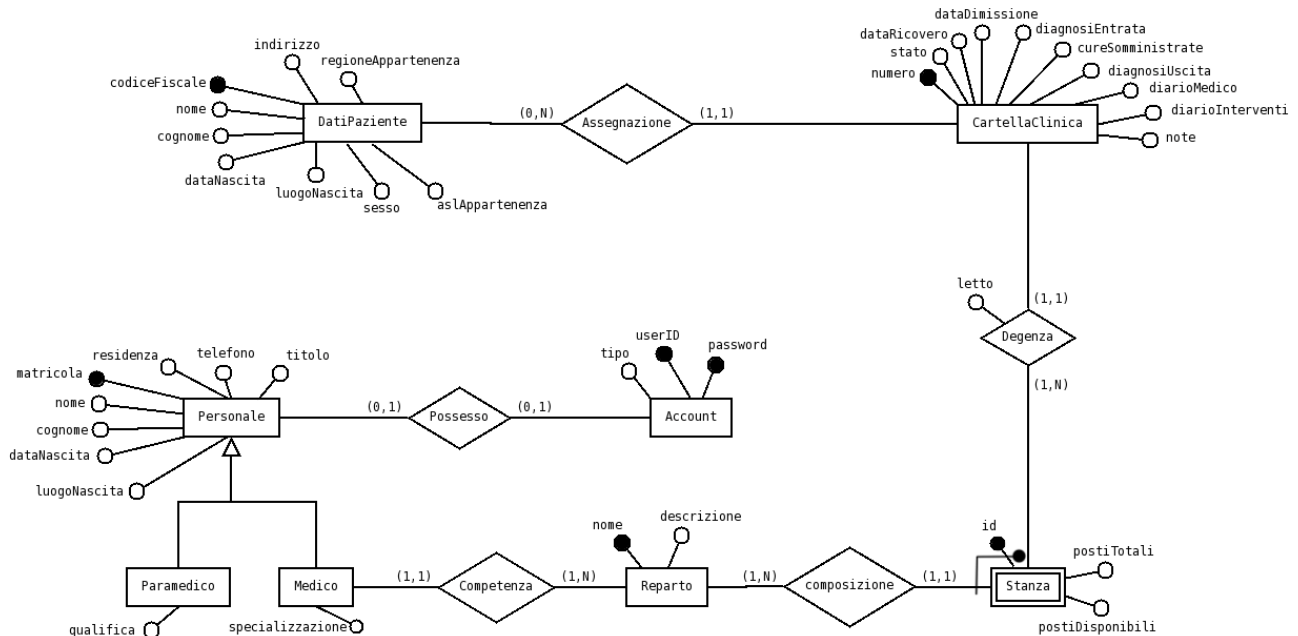
Per assicurare la massima affidabilità e portabilità del sistema in fase di realizzazione, si ricorrerà all'utilizzo di un database relazionale (MySQL) per contenere i dati relativi agli oggetti indicati in precedenza.

La scelta del database relazionale è ulteriormente giustificata dal fatto che sugli oggetti vengono eseguite query complesse.

Questo tipo di database sarà interfacciato al resto del sistema, in maniera del tutto trasparente all'utente, tramite *JDBC*, funzionalità standard fornita dal linguaggio *Java*.

Nel database relazionale i dati vengono memorizzati in tabelle che seguono uno schema predefinito. Quindi ogni oggetto viene memorizzato nella relativa tabella, attraverso un insieme di attributi.

Si riporta di seguito lo schema ER, in modo da evidenziare le entità individuate e le relazioni che intercorrono fra esse.



Per quanto riguarda la relazione Degenza, si preferisce non rispettare i canoni del mapping E-R -> Schema Logico che prevedono, in caso di relazione del tipo 1:N l'accorpamento sul lato 1 degli attributi di relazione e della chiave primaria dell'entità lato N. Per non appesantire la tabella relativa all'entità CartellaClinica, infatti, viene creata una ulteriore tabella Degenza, come se si trattasse di una relazione N:N. Maggiori dettagli sono specificati in seguito.

Si riporta in seguito l'elenco delle tabelle relative agli oggetti con gli attributi corrispondenti.

1. Tabella **DatiPaziente**: contiene i dati anagrafici relativi ai pazienti

- *codiceFiscale*
- *nome*
- *cognome*
- *dataNascita*
- *luogoNascita*
- *sex*
- *indirizzo*
- *asl*
- *regione*

codiceFiscale: chiave primaria, codice fiscale del paziente

nome: nome del paziente

cognome: cognome del paziente

dataNascita: data di nascita del paziente

luogoNascita: luogo di nascita del paziente

sex: il sesso del paziente (M/F)

indirizzo: indirizzo di residenza del paziente

asl: ASL di appartenenza del paziente

regione: regione di appartenenza del paziente (serve per statistiche e servizi interni alla clinica)

2. Tabella **Personale**: contiene i dati relativi ai membri dello staff medico-sanitario

- *matricola*
- *nome*
- *cognome*
- *dataNascita*
- *luogoNascita*
- *residenza*
- *telefono*
- *titolo*

matricola: chiave primaria, identificativo del componente dello staff medico-sanitario

nome: nome del componente

cognome: cognome del componente

dataNascita: data di nascita del componente

luogoNascita: luogo di nascita del componente

residenza: indirizzo e comune di residenza del componente

telefono: numero telefonico del componente

titolo: la qualifica del componente (medico, paramedico, responsabile accettazione)

Per quanto riguarda un Medico vengono aggiunti i seguente attributi:

specializzazione: tipo di specializzazione del medico (ad esempio, ortopedia, oculistica, generale)

reparto: chiave esterna, rappresenta il nome del reparto a cui è stato assegnato il medico, in base alla sua specializzazione

Per quanto riguarda un Paramedico viene aggiunto il seguente attributo:

qualifica: infermiere, portantino, etc

Per un paramedico, non è necessario tenere traccia del Reparto assegnato, in quanto si presume che i paramedici “girino” tra i vari reparti.

Si deduce, dunque, che Medico e Paramedico sono specializzazioni di Personale.

3. Tabella **CartellaClinica**: contiene i dati relativi alle cartelle cliniche associate ai pazienti

- *numero*
- *codiceFiscale*
- *stato*
- *dataRicovero*
- *dataDimissione*
- *diagnosiEntrata*
- *cureSomministrate*
- *diagnosiUscita*
- *diarioMedico*
- *diarioInterventi*
- *note*

numero: chiave primaria, identificativo della cartella clinica (ha la forma numero/anno)

codiceFiscale: chiave esterna, codice fiscale del paziente a cui è associata la cartella clinica

stato: ricoverato/dimesso, necessario in quanto anche le cartelle cliniche vengono conservate del database. Lo stato “Ricoverato” viene impostato alla creazione della cartella clinica, mentre lo stato “Dimesso” viene impostato automaticamente alla dimissione del paziente.

dataRicovero: data di ricovero del paziente associato alla cartella clinica

dataDimissione: data di dimissione del paziente

diagnosiEntrata: diagnosi di entrata del paziente

cureSomministrate: cure somministrate al paziente durante la degenza

diagnosiUscita: diagnosi di dimissione del paziente

diarioMedico: informazioni relative allo specifico ricovero del paziente

diarioInterventi: il diario degli interventi subiti dal paziente nel ricovero attuale

note: eventuali note

4. Tabella **Stanza**

- *nomeReparto*
- *id*
- *postiTotali*
- *postiDisponibili*

nomeReparto: il nome del reparto di cui la stanza fa parte, chiave esterna

id: il numero identificativo della stanza all'interno del reparto, chiave primaria

postiTotali: il numero di posti letto totali della stanza

postiDisponibili: il numero di posti letto liberi della stanza

5. Tabella **Reparto**

- *nome*
- *descrizione*

nome: il nome univoco del reparto, chiave primaria

descrizione: campo di testo contenente una eventuale descrizione del reparto

6. Tabella **Account**

- *tipo*
- *userID*
- *password*
- *matricola*

tipo: il tipo di account (amministratore, dirigente sanitario, responsabile accettazione, medico o paramedico)

userID: la username univoco dell'account, chiave primaria

password: la password dell'account (invisibile), chiave primaria

matricola: matricola del componente del personale proprietario dell'account, chiave esterna

NOTA: diversamente da come definito nel RAD si preferisce utilizzare la dicitura Account invece di DatiUtente.

7. Tabella **Degenza**

- *numeroCartella*
- *idStanza*
- *letto*

numeroCartella: numero della cartella clinica del paziente (per identificare il ricovero), chiave primaria + esterna

idStanza: identificativo della stanza assegnata al cliente, chiave primaria + esterna

letto: letto assegnato al paziente all'interno della stanza

NOTA: la tabella Degenza corrisponde all'Assegnazione di un posto letto ad un paziente ricoverato, identificato nel RAD con la classe Assegnazione.

3.5 Controllo degli accessi e sicurezza

3.5.1 Controllo degli accessi

Il software EasyClinique prevede cinque livelli utente, tutti estensione dell'utente generico, generalizzazione individuata nel RAD 4.0:

1. **Amministratore del sistema (Livello 0):** colui che si occupa della registrazione, modifica e cancellazione degli account che determinano i permessi di accesso ai dati.
2. **Dirigente sanitario (Livello 1):** colui che gestisce i dati relativi al personale ospedaliero ed effettua controlli generali sull'operato della struttura, ad esempio, attraverso la visualizzazione di statistiche e grafici. E', inoltre, l'unico in grado di gestire direttamente i dati relativi ai reparti.
3. **Responsabile accettazione (Livello 2):** colui che gestisce l'assegnazione delle stanze di degenza, basandosi su una lista d'attesa, e che può gestire i dati relativi agli assistiti
4. **Medico (Livello 3):** colui che, oltre a visionare le cartelle cliniche dei pazienti, è in grado di modificarle, inserendo, ad esempio, cure da somministrare, stabilire la dimissione dei pazienti, etc)
5. **Paramedico (infermiere e operatore sanitario) (Livello 4):** colui che visiona le cartelle cliniche dei pazienti (ad esempio per somministrare loro farmaci, accompagnarli in sale per terapie particolari, etc)

Ad ognuno di questi utenti viene assegnata una coppia (userID, password) con la quale può accedere al sistema.

Ogni utente, a seconda del livello di appartenenza, specificato in fase di registrazione, può accedere a diverse funzionalità.

Le politiche di accesso vengono descritte nella *Tabella di Controllo di Accesso*.

Le righe della tabella rappresentano gli attori del sistema, mentre le colonne rappresentano gli oggetti su cui sono regolamentati gli accessi.

Seppur onerosa, la scelta di descrivere le politiche di accesso attraverso la suddetta tabella è effettuata per permettere una estrema facilità di comprensione della suddivisione delle funzionalità.

Tabella di controllo di accesso

Gli attori del sistema sono:

2. Amministratore
3. DirigenteSanitario
4. ResponsabileAccettazione
5. Medico
6. Paramedico

Gli oggetti sono:

- Utente
- Paziente
- Manuale
- Reparto
- Stanza
- CartellaClinica
- Personale
- Degenza

NOTA: lo stereotipo <<create>> all'interno della tabella viene utilizzato per definire la creazione, cioè l'inserimento, di una nuova istanza dell'oggetto interessato

| Oggetto Attore | Utente | Paziente | Cartella Clinica |
|------------------------------|---|---|--|
| Amministratore | <<create>> ModificaUtente CancellaUtente RicercaUtente | RicercaPaziente | |
| Dirigente Sanitario | | RicercaPaziente | |
| Responsabile Accettazione | | <<create>> ModificaPaziente RicercaPaziente | <<create>> RicercaCartellaClinica |
| Medico | | RicercaPaziente | ModificaCartellaClinica StampaCartellaClinica RicercaCartellaClinica |
| Paramedico | | RicercaPaziente | StampaCartellaClinica RicercaCartellaClinica |

| Attore \ Oggetto | Reparto | Stanza | Manuale |
|---------------------------|--|---|------------------|
| Amministratore | | | RicercaArgomento |
| Dirigente Sanitario | <<create>> RicercaReparto CancellazioneReparto | <<create>> RicercaStanza ModificaStanza | RicercaArgomento |
| Responsabile Accettazione | | | RicercaArgomento |
| Medico | | | RicercaArgomento |
| Paramedico | | | RicercaArgomento |

| Attore \ Oggetto | Personale | Degenza |
|---------------------------|--|---|
| Amministratore | | |
| Dirigente Sanitario | <<create>> InserimentoMembroPersonale RicercaMembroPersonale ModificaMembroPersonale StampaMembroPersonale | |
| Responsabile Accettazione | | <<create>> CambioReparto DimissionePaziente |
| Medico | | |
| Paramedico | | |

3.5.2 Sicurezza

Come detto, il software EasyClinique assicura la protezione dei dati tramite un controllo degli accessi al sistema regolato dall'assegnazione ad ogni utente di userID e password.

La coppia userID e password, ovviamente deve essere personale ed univoca.

Nel dettaglio, è preferibile che non ci siano due diversi account con lo stesso userID.

Per utilizzare il sistema, ogni utente deve essere registrato nella lista degli account.

Il sistema, quindi, ad ogni sessione chiede all'utente l'inserimento dei suddetti dati e ne verifica la correttezza. In caso di inserimento dati non corretti, richiede nuovamente la digitazione di userID e password.

Una volta verificati i dati, il sistema controlla il livello di accesso dell'utente e abilita le funzionalità utilizzabili.

EasyClinique fornisce una sicurezza di livello medio, in quanto non è possibile accedervi se non si è in possesso di una coppia (userID - password), ed è difficile “forzarlo” dall'esterno, dato che la comunicazione avviene tramite rete locale.

UserID e password devono rispettare i canoni riportati di seguito.

- userID: deve avere una lunghezza compresa fra un minimo di 4 e un massimo di 11 caratteri. Può contenere caratteri alfanumerici, ma non caratteri speciali, quali ? / *, e segni di punteggiatura.
- password: deve avere una lunghezza compresa fra un minimo di 4 e un massimo di 11 caratteri alfanumerici. Come per userID, non deve contenere caratteri speciali.

3.6 Gestione del controllo globale

Il sistema software è gestito da un “event driven control” che permette di sfruttare appieno le possibilità offerte dall’ambiente di sviluppo Java.

Secondo questo “paradigma”, un ciclo principale attende il verificarsi di eventi.

Al verificarsi di un evento, solitamente generato da un utente attraverso l’interfaccia grafica, il controllo viene trasferito al relativo oggetto che lo gestisce che, a sua volta, manda in esecuzione un sottosistema.

Solitamente, ciascuna operazione è terminata con un messaggio di notifica all'utente che utilizza il Client.

3.7 Boundary Conditions

3.7.1 Configuration

Di seguito vengono definite, per ogni oggetto persistente introdotto nel punto 3.4, le fasi in cui viene creato, distrutto e archiviato nel database.

- **DatiPaziente:** oggetto creato dal ResponsabileAccettazione, come specificato nello use case InserimentoPaziente ed archiviato all’uscita dello stesso. Non è prevista una funzione di rimozione dal sistema.
- **Personale:** oggetto creato dal DirigenteSanitario, come specificato nello use case InserimentoPersonale ed archiviato all’uscita dello stesso. La rimozione dal sistema viene effettuata attraverso lo use case CancellazionePersonale.
- **Account:** oggetto creato dall’Amministratore, come specificato nello use case InserimentoUtente ed archiviato all’uscita dello stesso. La rimozione dal sistema viene effettuata attraverso lo use case CancellazioneUtente.
- **CartellaClinica:** oggetto creato dal ResponsabileAccettazione come specificato nello use case InserimentoCartellaClinica ed archiviato all’uscita dello stesso. Non è prevista una rimozione dal sistema.

- **Reparto:** oggetto creato dal DirigenteSanitario come specificato nello use case InserimentoReparto ed archiviato all'uscita dello stesso. La rimozione dal sistema viene effettuata attraverso lo use case CancellazioneReparto.
- **Degenza:** oggetto creato dal ResponsabileAccettazione come specificato nello use case AssegnazionePostoLetto ed archiviato all'uscita dello stesso. La rimozione dal sistema viene effettuata attraverso lo use case DimissionePaziente.
- **Stanza:** oggetto creato dal DirigenteSanitario come specificato nello use case InserimentoStanza ed archiviato all'uscita dello stesso. Non è prevista una rimozione dal sistema, ma questa è implicita al momento della cancellazione del reparto di appartenenza.

3.7.2 Start up – Shutdown use cases

Start up (primo avvio del sistema)

Il software EasyClinique verrà installato da uno degli sviluppatori sul Server.

Al primo avvio del software vengono creati in automatico un account per l'Amministratore ed uno per il DirigenteSanitario.

L' Amministratore può, così, da subito registrare nuovi account nel database del sistema che determinano i permessi di accesso ai dati, mentre il DirigenteSanitario può inserire nuovi componenti dello staff medico-sanitario a cui “legare” in seguito gli account.

Si deduce, quindi, che per permettere all' Amministratore di inserire nuovi account, il Dirigente deve aver già inserito dei membri dello staff medico-sanitario.

I successivi avvii di EasyClinique mostreranno direttamente la schermata principale, permettendo l'autenticazione dei diversi utenti e, quindi, l'accesso alle varie funzionalità.

Dunque, al primo avvio del sistema si presume che il DirigenteSanitario inserisca nuovi membri dello staff, in particolare i dati relativi al ResponsabileAccettazione.

Start up (avvii del sistema successivi ad imprevisti)

Agli avvii successivi il sistema verifica che non siano avvenuti shut-down improvvisi.

In caso si verificano tali imprevisti, il sistema controlla l'integrità dei dati e ripristina il sistema ad una condizione funzionante.

Start up (nuova sessione di login)

Il software EasyClinique, nel momento in cui riceve una richiesta di apertura di una nuova sessione, chiede all'utente i dati di accesso per poter stabilire quali funzionalità sono, appunto, accessibili.

Lo use case relativo al login è già stato presentato in fase di raccolta ed analisi dei requisiti.

Per completezza, se ne riporta in seguito una copia

| | |
|------------------------|---|
| Nome use case | Login |
| Partecipante | UtenteGenerico |
| Flusso di eventi | <ul style="list-style-type: none"> • L'utente accede al sistema inserendo login e password • Il sistema verifica i dati e visualizza le possibili operazioni che può effettuare l'utente. |
| Condizioni di ingresso | L'utente deve essere registrato |
| Condizioni di uscita | L'utente ha avuto accesso al sistema |
| Eccezioni | Il sistema non riconosce l'utente e mostra un errore. La connessione viene persa durante il riconoscimento dei dati. |
| Requisiti di qualità | |

NOTA: per lo use case diagram vedi EasyClinique RAD 4.0.

Shut Down

Lo Shut Down del sistema, come descritto nella fase di analisi, non avverrà mai, in quanto il sistema resta attivo 24 ore su 24.

Nel caso in cui si verificassero interruzioni impreviste, dovute a malfunzionamenti, cali di tensione ecc., la politica di ripristino intrapresa è strettamente legata a quella di start-up.

Logout

Il logout del sistema EasyClinique potrà essere effettuato da qualsiasi utente che richiede l'uscita dal sistema. Il sistema a quel punto disconnette l'utente e torna alla pagina principale.

Per quanto riguarda il logout richiesto dal cliente, lo use case è del tutto simile a quello definito nell'EasyClinique RAD 4.0.

| | |
|------------------------|---|
| Nome use case | Logout |
| Partecipante | UtenteGenerico |
| Flusso di eventi | <ul style="list-style-type: none"> • L'utente richiede l'uscita dal sistema. • Il sistema disconnette l'utente e mostra la finestra di accesso. |
| Condizioni di ingresso | L'utente deve essere loggato nel sistema. |
| Condizioni di uscita | L'utente è stato disconnesso dal sistema. |
| Eccezioni | Il sistema non riesce a disconnettere l'utente. La connessione viene persa durante l'operazione di logout. |
| Requisiti di qualità | |

3.7.3 Exception handling

OperazioneCompletata

- L'utente chiede al sistema di terminare un'operazione.
- Il sistema sottomette le eventuali modifiche prodotte dall'operazione
- Il sistema mostra la schermata principale: l'operazione risulta eseguita e l'utente viene abilitato dal sistema ad effettuare altre operazioni.

OperazioneInterrotta

- L'utente decide di interrompere l'operazione che aveva avviato.
- Il sistema non sottomette le informazioni eventualmente modificare e mostra la schermata principale.

MalfunzionamentoImprevisto

- Durante l'esecuzione di un'operazione avviene per una causa imprevista, un crash del sistema
- Il sistema annulla le operazioni in corso e notifica l'utente

Al ritorno alla normalità, il sistema agisce come specificato in *Start up (avvii del sistema successivi ad imprevisti)*

4. Servizi dei sottosistemi

Vengono di seguito descritti i servizi forniti da ognuno dei sottosistemi in termini di operazioni.

SottosistemaResponsabile

| | |
|-----------------------------|--|
| Modulo | CreazioneCartelleCliniche |
| Descrizione | Modulo che si occupa della creazione delle cartelle cliniche |
| Servizi offerti | |
| Servizio | Descrizione |
| InserimentoCartella Clinica | Tale servizio permette di inserire nel database le informazioni relative ad una nuova cartella clinica |
| RicercaCartellaClinica | Il servizio permette di ricercare una cartella clinica in base al codice fiscale del paziente o al numero della cartella clinica |

| | |
|-----------------------------|--|
| Modulo | GestionePazienti |
| Descrizione | Modulo che si occupa di gestire le operazioni sui dati dei pazienti |
| Servizi offerti | |
| Servizio | Descrizione |
| InserimentoPaziente | Il servizio permette di inserire nel database i dati di un nuovo paziente |
| ModificaPaziente | Il servizio per modificare i dati di un paziente. Legge il record paziente nel database, lo modifica e salva gli aggiornamenti apportati |
| Ricerca&Visualizza Paziente | Tale servizio ricerca(e visualizza) i dati di un un paziente nel database tramite il suo codice fiscale |

| | |
|------------------------|--|
| Modulo | GestioneDegenze |
| Descrizione | Modulo che si occupa di gestire le operazioni riguardanti le degenze |
| Servizi offerti | |
| Servizio | Descrizione |
| DimissionePaziente | Servizio che gestisce le dimissioni dei pazienti; legge la cartella clinica del paziente, registra la dimissione e libera il posto letto. |
| AssegnazionePostoLetto | Tale servizio permette di assegnare posti letto ai pazienti. Legge il numero della stanza, il numero della cartella clinica del paziente e registra l'assegnazione |
| CambioReparto | Il servizio gestisce il trasferimento di un paziente ricoverato da un reparto ad un altro |

SottosistemaDirigente

| | |
|------------------------|---|
| Modulo | GestioneReparti |
| Descrizione | Modulo che si occupa di gestire i reparti |
| Servizi offerti | |
| Servizio | Descrizione |
| InserimentoReparto | Il servizio permette di inserire nel database le informazioni di un nuovo reparto |
| CancellazioneReparto | Servizio che permette di cancellare un reparto. Ricerca il reparto, visualizzandone i dati, e lo cancella. |
| RicercaReparto | Servizio che permette di ricercare un reparto tramite il suo nome. |

| | |
|------------------------|---|
| Modulo | GestioneStanze |
| Descrizione | Modulo che si occupa della gestione delle stanze |
| Servizi offerti | |
| Servizio | Descrizione |
| InserimentoStanza | Il servizio permette di inserire nel database i dati di una nuova stanza |
| ModificaStanza | Servizio che permette di modificare i dati di una stanza. Ricerca la stanza, la modifica e ne riscrive il contenuto nel database |
| RicercaStanza | Servizio che permette di ricercare una stanza tramite il suo id. |

| | |
|------------------------|--|
| Modulo | GestioneGrafici |
| Descrizione | Modulo che si occupa di gestire i grafici |
| Servizi offerti | |
| Servizio | Descrizione |
| VisualizzaGrafico | Tale servizio permette di visualizzare un grafico statistico |

| | |
|------------------------|--|
| Modulo | GestionePersonale |
| Descrizione | Modulo che si occupa di gestire i dati relativi ai membri del personale medico-sanitario |
| Servizi offerti | |
| Servizio | Descrizione |
| InserimentoPersonale | Il servizio permette di inserire nel database i dati di un |

| | |
|------------------------|---|
| | nuovo membro del personale medico-sanitario |
| ModificaPersonale | Servizio che permette di modificare i dati di un membro del personale medico sanitario; ricerca il membro, lo modifica e ne riscrive il contenuto nel database. |
| CancellazionePersonale | Servizio che permette di cancellare un membro del personale medico sanitario; ricerca il membro e lo cancella dal database |
| RicercaPersonale | Servizio che permette di ricercare un membro del personale |

SottosistemaMedicoParamedico

| | |
|-------------------------|--|
| Modulo | GestioneCartelleCliniche |
| Descrizione | Modulo che si occupa della gestione delle cartelle cliniche. |
| Servizi offerti | |
| Servizio | Descrizione |
| ModificaCartellaClinica | Tale servizio mostra all'utente i dati di una cartella clinica preesistente nel database e permette di modificarne i dati e salvarli nel database. |
| RicercaCartellaClinica | Il servizio permette di ricercare le cartelle cliniche in base al codice fiscale del paziente o al numero di cartella clinica |
| StampaCartellaClinica | Tale servizio permette di stampare, su file o su carta, le informazioni di una cartella clinica. |

| | |
|---------------------------------------|---|
| Modulo | GestioneNotifiche |
| Descrizione | Modulo che si occupa di gestire la notifica delle dimissioni dei pazienti ricoverati. |
| Servizi offerti | |
| Servizio | Descrizione |
| InvioNotifica (NotificaDimissione) | Il servizio permette al medico di notificare al responsabile accettazione la dimissione di un paziente. |

SottosistemaAmministratore

| | |
|------------------------|--|
| Modulo | GestioneAccount |
| Descrizione | Modulo che si occupa della gestione degli account. |
| Servizi offerti | |
| Servizio | Descrizione |
| CreaAccount | Tale servizio permette di inserire nella tabella degli accessi un nuovo utente. |
| ModificaAccount | Tale servizio permette di ricercare un utente e cancellarne l'accesso dalla tabella degli accessi. |
| EliminaAccount | Tale servizio permette di cancellare uno o più utenti dalla tabella degli accessi. |

Inoltre, per tutti gli utenti, sono definiti il seguente modulo.

| | |
|------------------------|---|
| Modulo | VisualizzazionePazienti |
| Descrizione | Modulo che si occupa della ricerca e visualizzazione dei dati di un paziente. |
| Servizi offerti | |
| Servizio | Descrizione |
| RicercaPaziente | Il servizio permette di ricercare e visualizzare un paziente in base al suo codice fiscale. |

5. Glossario

| Terms | Definitions |
|---------------------------------|---|
| EasyClinique | Nome del sistema software sviluppato. |
| UtenteGenerico | Termine che identifica uno qualsiasi degli utenti registrati nel sistema; viene utilizzato nella descrizione di funzionalità accessibili ad ogni tipo di utente (login, logout, consultazione manuale utente, ricerca e visualizzazione dati non medici relativi ai pazienti). |
| Amministratore | Termine che identifica colui che si occupa di registrare, modificare e cancellare gli account degli utenti di altro tipo. |
| DirigenteSanitario | Termine che identifica colui che si occupa di: gestire i dati relativi al personale ospedaliero (ossia medici, paramedici e responsabili dell'accettazione); visualizzare grafici e statistiche sull'operato della struttura; a gestire i dati relativi ai reparti e alle stanze. |
| Dirigente | Termine usato in modo intercambiabile con "Dirigente sanitario" |
| Personale/staff medico | Termine collettivo che identifica medici e paramedici e responsabile sanitario. |
| ResponsabileAccettazione | Termine che identifica colui che gestisce l'assegnazione dei pazienti alle stanze È lui che gestisce i dati non medici relativi ai pazienti. |
| Responsabile | Termine usato in modo intercambiabile con "Responsabile accettazione" |
| Medico | Termine che identifica colui che si occupa della gestione delle cartelle cliniche dei pazienti; è l'unico tipo di utente che può modificarle. |

| | |
|--------------------------------|--|
| Paramedico | Termine collettivo che identifica infermieri e operatori sanitari. Può soltanto visionare le cartelle cliniche dei pazienti e i loro dati. |
| Cartella clinica | Termine con cui si intendono i dati medici relativi ad un paziente. Può essere visionata solo da paramedici e medici, e modificata soltanto da questi ultimi. |
| Dati personale/paziente | Termine con cui si intendono i dati non medici relativi ad un paziente o i dati relativi ad un membro del personale medico. |
| Grafico | Un grafico raffigurante informazioni di interesse sulla struttura. |
| Reparto | Un reparto della struttura, costituito da un insieme di stanze. |
| Stanza | Una stanza facente parte di un reparto e comprendente una serie di posti letto. |
| Posto letto | Posto letto che è incluso in una stanza e può essere assegnato a un paziente. |
| Account | Termine utilizzato per descrivere i dati di un utente registrato nel sistema (userID, password e tipo). I termini userID e userName sono, come di consueto, utilizzati in modo interscambiabile. |
| Report | Documento stampato contenente dati relativi ad una cartella clinica o ad un membro del personale medico. |
| Hardware | Parti fisiche del sistema |
| Software | L'insieme di applicazioni e programmi facenti parte del sistema |

| | |
|---------------------------------|--|
| Design goals | Qualità del software che devono essere ottimizzate |
| Coupling (accoppiamento) | Misura dell'interdipendenza tra due sottosistemi o classi |
| Mapping | Funzione matematica che fa corrispondere un elemento (o un insieme di elementi) di un modello a ogni elemento di un altro modello. Nel caso in questione, il mapping avviene tra hardware e software |
| Boundary conditions | Condizioni particolari che devono essere gestite dal sistema |
| Client | Macchina che funziona da terminale e che inoltra le richieste al Server |
| Server | Macchina sulla quale è memorizzata gran parte del sistema e che fornisce servizi al Client |
| Three-tiers | Tipo di architetture basata su tre strati e utilizzata nella progettazione del sistema |