

SDD

System

Design

Document

ChemioPlan

|  |  |
| --- | --- |
| Riferimento |  |
| Versione | 0.7 |
| Data | 2/12/2022 |
| Destinatari | Prof.ssa F. Ferrucci, Prof. F. Palomba |
| Presentato da | Clericuzio Alessandro  Contardo Vittorio  di Pippa Francesco Pio  Lo Conte Christian  Matteis Francesco |
| Approvato da |  |

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Data | Versione | Cambiamenti | Autori |
| 29/11/2022 | 0.1 | Obiettivi del sistema, design goals, definizioni, riferimenti, architettura del sistema proposto, panoramica, condizioni limite, controllo globale del sistema | [Gruppo] |
| 30/11/2022 | 0.2 | Decomposizione in sottosistemi, mapping, gestione dati persistenti | [Gruppo] |
| 30/11/2022 | 0.3 | Revisione | [Gruppo] |
| 1/12/2022 | 0.4 | Gestione dati persistenti, controllo degli accessi e sicurezza | [Gruppo] |
| 2/12/2022 | 0.5 | Architettura dei sistemi simili, servizi dei sottosistemi | [Gruppo] |
| 2/12/2022 | 0.6 | Revisione | [Gruppo] |
| 29/12/2022 | 0.7 | Revisione e modifiche | [Gruppo] |

Sommario

[1. Introduzione 4](#_Toc123222528)

[1.1 Obiettivi del Sistema 4](#_Toc123222529)

[1.2 Design Goals e Trade-off 4](#_Toc123222530)

[1.3 Definizioni, acronimi e abbreviazioni 5](#_Toc123222531)

[1.4 Riferimenti 5](#_Toc123222532)

[1.5 Panoramica 6](#_Toc123222533)

[2.Architettura dei Sistemi simili 6](#_Toc123222534)

[3.Architettura del Sistema proposto 7](#_Toc123222535)

[3.1 Panoramica 7](#_Toc123222536)

[3.2 Decomposizione in sottosistemi 7](#_Toc123222537)

[3.2.1 Deployment Diagram 9](#_Toc123222538)

[3.3 Mapping Hardware/Software 10](#_Toc123222539)

[3.4 Gestione dati persistenti 10](#_Toc123222540)

[3.5 Controllo degli accessi e sicurezza 10](#_Toc123222541)

[3.6 Controllo Globale del Sistema 12](#_Toc123222542)

[3.7 Condizioni limite 12](#_Toc123222543)

[12](#_Toc123222544)

[3.7.1 Start-up 12](#_Toc123222545)

[3.7.2 Shut-down 14](#_Toc123222546)

[3.7.3 Fallimenti 15](#_Toc123222547)

[4. Servizi dei sottosistemi 16](#_Toc123222548)

[5. Glossario 20](#_Toc123222549)

# 

# 1. Introduzione

## 1.1 Obiettivi del Sistema

L’ obiettivo del sistema è quello di fornire uno strumento di supporto all’ospedale universitario Federico II di Napoli per gestire al meglio le sedute di somministrazione chemioterapiche. Per questo motivo nasce ChemioPlan!

Il sistema fornirà un forte aiuto a dottori e farmacisti dell’ospedale, andando ad evitare gli sprechi di medicinali e migliorando la fase di scheduling delle prenotazioni di somministrazioni per i pazienti.

## 1.2 Design Goals e Trade-off

Nella seguente tabella andiamo ad illustrare gli obiettivi di design per il sistema e le relative priorità (1 = priorità alta, 2 = priorità media, 3 = priorità bassa). Per ogni obiettivo viene riportato anche il requisito non funzionale di origine.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Priorità** | **ID** | **Descrizione** | **Categoria** | **Origine** |
| 1 | DG\_1 | *Modificabilità*: Il Sistema deve garantire una buona modularità, in modo da poter modificare facilmente le funzionalità esistenti | Manutenibilità | RQNF\_15 |
| 2 | DG\_2 | *Estensibilità*: Il sistema prodotto deve offrire una buona estendibilità, in modo da poter aggiungere facilmente nuove funzionalità al sistema | Manutenibilità | RQNF\_15 |
| 3 | DG\_3 | *Tempi di risposta*: Il sistema dovrà garantire che la quantità dei farmaci venga aggiornata in al più 2 secondi, in modo da evitare il crearsi di ordini di prodotti che non sono disponibili. | Performance | RQNF\_1 |
| 3 | DG\_4 | *Throughput:* Il sistema dovrà:  1) Creare l’ordine.  2) Aggiornare le quantità del prodotto, in al più 3 secondi. | Performance | RQNF\_1 |
| 2 | DG\_5 | *Portabilità*: il sistema deve garantire il corretto funzionamento su ogni piattaforma web | Maintenance | RQNF\_6 |
| 3 | DG\_6 | *Utilità:* il sistema deve supportare il lavoro dell’utente fornendo ogni tipo di funzionalità che esso necessità, in base al ruolo che ricopre | End user | RQNF\_7 |
| 2 | DG\_7 | *Usabilità*: Il sistema deve fornire una rapida navigazione attraverso l’uso di un menu accessibile da tutte le aree del sistema | End user | RQNF\_12 |
| 3 | DG\_8 | *Usabilità*: Il sistema deve essere intuitivo da utilizzare anche senza consultare manuali o documentazioni | End user | RQNF\_12 |
| 2 | DG\_ 9 | *Robustezza*: il sistema sarà in grado di garantire la correttezza dei dati inseriti in input notificando all’utente dei messaggi di errore. | Dependability | RQNF\_9 |
| 1 | DG\_10 | *Safety*: il sistema dovrà garantire che i dati riguardo i pazienti verranno salvati in modo corretto. | Dependability | RQNF\_9 |

*Safety vs Tempo di risposta*

Siccome il sistema gestisce dati piuttosto sensibili, preferiamo garantire che i dati siano salvati correttamente a scapito dei tempi di risposta.

*Robustezza vs Throughput*

Siccome i dati gestisti sono sensibili, preferiamo un maggior controllo dei dati di input a scapito dei tempi di risposta del throughput.

*Usabilità (DG\_8) vs Utilità*

Siccome il sistema deve essere eseguito su vari dispositivi, preferiamo che il sistema sia più facile da utilizzare piuttosto che introdurre più funzionalità.

## 1.3 Definizioni, acronimi e abbreviazioni

* DG: Design goal
* RQNF: Requisito non funzionale

## 1.4 Riferimenti

* Requisiti non funzionali: Sezione 3.3 del RAD

## 1.5 Panoramica

Nel documento andremo ad analizzare le architetture di sistemi simili, la decomposizione in sottosistemi del sistema proposto andando a definire la strategia di deploy e le condizioni limite. Saranno definiti i servizi esposti da ciascun sottosistema.

# 2.Architettura dei Sistemi simili

Nel nostro caso non è presente un’architettura software già esistente, abbiamo quindi deciso di analizzare delle architetture di sistemi simili.

Il primo software per la gestione della pianificazione del personale medico è “Petal scheduling for physicians”. Esso è composto da un unico modulo supportato dall'intelligenza artificiale per un efficiente scheduling dei turni di lavoro di medici ed infermieri.

Il Secondo software “Comarch medNote” è un sistema per la gestione della documentazione medica, anch'esso è composto da un unico modulo per la gestione delle informazioni sullo stato di salute dei pazienti.

Nei software analizzati si evince che sia stata utilizzata un’architettura three–tier, dato che questi sistemi si basano su un database per la memorizzazione e la gestione dei pazienti e della documentazione medica che viene interrogato da applicazioni desktop/web.

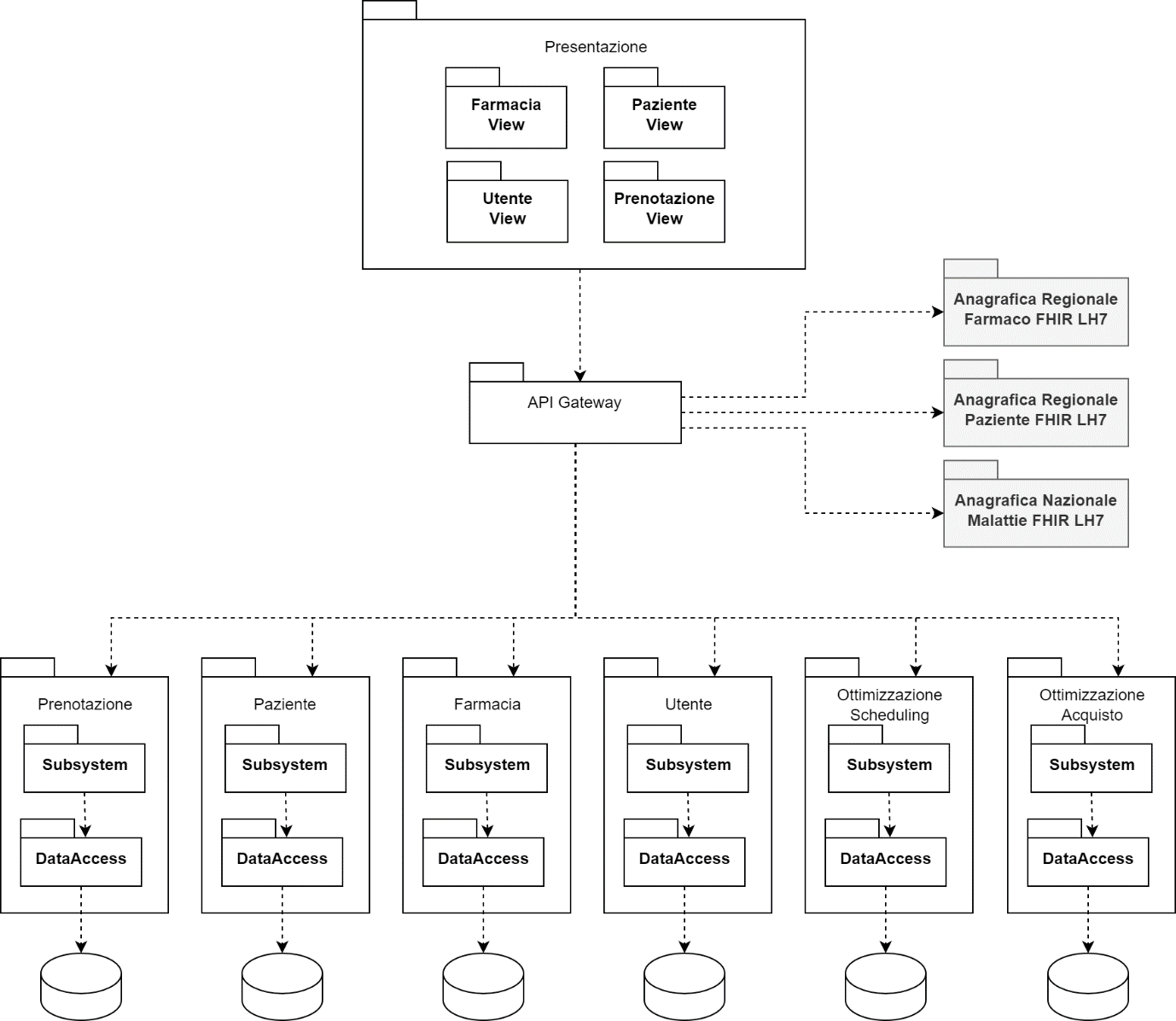
Nessuno dei software analizzati comprende quindi un processo di scheduling ottimale per le somministrazioni chemioterapiche e una gestione efficiente dei farmaci.

# 3.Architettura del Sistema proposto

## 3.1 Panoramica

ChemioPlan è una web application che fornisce diverse funzionalità tra cui: lo scheduling delle prenotazioni delle sedute chemioterapiche e la gestione efficiente delle scorte farmaceutiche.

## 3.2 Decomposizione in sottosistemi



Il sistema è diviso in 4 livelli logici

* Presentazione: che ha il compito di mostrare all'utente le informazioni del sistema
* API Gateway: livello intermediario tra client e microservizi
* Microservizi: che si occupano dei servizi di business e di accedere e modificare i dati
* Database: rende persistente i dati

Il livello Presentazione è composto da 4 sottosistemi

* Utenza: mostra all'utente i servizi di autenticazione e accesso
* Farmacia: mostra le informazioni relative al farmaco e al magazzino
* Paziente: mostra le informazioni dei pazienti e delle relative terapie
* Prenotazione: mostra le informazioni delle prenotazioni

L’API Gateway ha il compito di chiamare i microservizi, comunicare ed eseguire chiamate a server esterni.

Il sistema utilizza 3 diversi sistemi esterni:

* Anagrafica Regionale Farmaco: fornisce informazioni relativi ai farmaci utilizzati;
* Anagrafica Regionale Paziente: fornisce informazioni relativi ai pazienti dell’ospedale;
* Anagrafica Nazionale Malattia: fornisce informazioni relativi alle malattie dei pazienti.

L’API Gateway comunica con questi 3 servizi attraverso lo standard FHIR HL7.

Il sistema è composto da 6 microservizi che si occupano delle logiche di business e di accedere e apportare modifiche ai propri DB

* Utenza: modella l'autenticazione e l'accesso dell'utente e i relativi accessi al DB
* Farmacia: modella il magazzino e la gestione dei farmaci e relativi accessi a DB
* Paziente: modella il paziente e l'accesso al DB
* Prenotazione: modella le prenotazioni e l'accesso al DB
* Scheduling: si occupa di ottimizzazione le prenotazioni
* Ottimizzazione Acquisto: si occupa di ottimizzazione gli acquisti dei medicinali

Il sistema ha una struttura a 4 livelli usata per disaccoppiare il livello di presentazione dai relativi microservizi.

Abbiamo scelto questa struttura in modo da rendere facile l’aggiunta e la modifica (in linea con i design goal DG\_ 1 e DG\_ 2) sia di parti dell'interfaccia utente che dell’implementazione dei servizi di business con i microservizi. Abbiamo aggiunto un API Gateway per ridurre l'accoppiamento tra UI e microservizi.

### 

### 3.2.1 Deployment Diagram

Chemio Plan si compone di 3 componenti: una web app, un container di business ed i container dei microservizi con i relativi DB.

È stato scelto MongoDB come DataBase per una memorizzazione ed interrogazione facile ed efficiente dei dati persistenti divisi in ogni microservizio.

L’architettura prevede un dialogo tra un applicazione web, i livelli di business e i mircoservizi attraverso un protocollo di rete e la comunicazione tra i microservizi e il DB MongoDB attraverso il MongoDB Driver Sync per Java e PyMongo per Python (se necessario).

## 3.3 Mapping Hardware/Software

Essendo un sistema basato su microservizi avremmo bisogno di diversi componenti per farlo funzionare:

* Web server, che conterrà il layer di presentazione e il layer di business con i control
* Dei server per i microservizi implementati
* Un Database per ogni microservizio

Il sistema avrà bisogno di diverse macchine con un browser in grado di connettersi al web server.

Il sistema inoltre avrà bisogno di una macchina che consenta di far funzionare ogni microservizio.

## 3.4 Gestione dati persistenti

Per la gestione dei dati persistenti, ChemioPlan utilizza diversi database di tipo non relazionale tramite MongoDB, in quanto il sistema è suddiviso in microservizi che hanno un proprio Database.

Utilizziamo MongoDB perché offre maggiore sicurezza, affidabilità ed efficienza, inoltre è anche molto flessibile e ci consente di modificare la struttura o lo schema dei dati rispettando il design goal sulla modificabilità.

Altri dati che riguardano informazioni generali dei farmaci e dei pazienti, vengono recuperati da un servizio esterno che segue lo standard [FHIR-HL7](https://hl7.org/fhir/).

## 3.5 Controllo degli accessi e sicurezza

L’accesso al sistema è consentito agli utenti che possiedono uno username e una password, in modo da evitare accessi fraudolenti al sistema di utenti non autorizzati assicurando l’integrità dei dati sensibili.

Nelle prime versioni del sistema non sono previsti metodi per il recupero o la modifica della password.

Il salvataggio delle password avverrà con cifratura, utilizzando l’algoritmo di hashing SHA-1.

Questo processo di autenticazione garantisce un primo livello di sicurezza e un controllo sulla tipologia dell’utente assicurerà che ogni utente si interfacci solamente con le proprie mansioni di competenza.

Le operazioni che ogni utente potrà effettuare sulla web application sono definite nella seguente tabella:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Oggetti  Attore | Utente | Farmacia | Prenotazioni | Pazienti |
| Dottore | 1.Login  2.Logout |  | 1.Inserisce una nuova prenotazione  2.Modifica prenotazione  3.Visualizza tutte le prenotazioni | 1.Visualizza scheda paziente  2.Modifica trattamento paziente  3.Inserisce una nuova scheda paziente |
| Responsabile  Farmacia | 1.Login  2.Logout | 1.Visualizza tutti i farmaci in magazzino  2.Inserisce lotto farmaci già registrato  3.Inserisce un nuovo lotto farmaco  4.Ordina un farmaco  5.Visualizza dettagli farmaco  6.Modifica del farmaco |  |  |
| Amministratore | 1.Login  2.Logout  3.Registrazione utenti |  |  |  |

## 3.6 Controllo Globale del Sistema

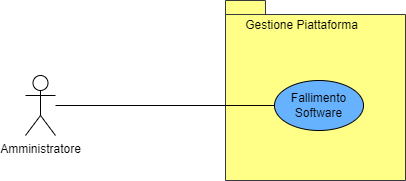
Il sistema del controllo del flusso globale adottando da ChemioPlan è di tipo Event-Driven Control; quindi, il sistema attende un qualsiasi evento esterno determinato dall’utente e agisce di conseguenza. Tutte le interfacce grafiche consentiranno di navigare tra le funzionalità del sistema.

I server su cui sono installati i DBMS per i microservizi consentiranno a tutti i Client l’accesso ai dati persistenti.

## 

## 3.7 Condizioni limite

## 



### 3.7.1 Start-up

|  |  |  |  |
| --- | --- | --- | --- |
| Identificativo  UC\_SU\_1 | Start-up del sistema | Data | 30/11/2022 |
| Vers. | 0.1 |
| Autore | Alessandro Clericuzio  Vittorio Contardo |
| Descrizione | Definisce la funzionalità di avvio del sistema. | | |
| Attore principale | Amministratore della piattaforma.  Avvia il sistema. | | |
| Attori secondari | N/A | | |
| Entry Condition | L’amministratore accede al dispositivo su cui è installato il sistema. | | |
| Exit Condition  On success | Il sistema si avvia. | | |
| Exit condition  On failure | Il sistema non si avvia. | | |
| Rilevanza/User priority | Elevata | | |
| Frequenza stimata | 1/giorno | | |
| Extension point | N/A | | |
| Generalization of | N/A | | |
| Flusso di eventi principale/ Main scenario | | | |
| 1 | Amministratore | Avvia il servizio MongoDB. | |
| 2 | Amministratore | Avvia i Server relativi ai microservizi. | |
| 3 | Amministratore | Avvia il Web Server. | |
| 4 | Sistema | Comunica all’amministratore che la fase di start-up è avvenuta con successo. | |
| 5 | Amministratore | Accede ai Server | |
| Scenario/ Flusso di eventi di errore: Non è possibile avviare il servizio MongoDB | | | |
| 1.1 | Sistema | Mostra un messaggio relativo alla causa dell’errore. | |
| Scenario/ Flusso di errore: Non è possibile avviare i Server | | | |
| 2 .1 | Sistema | Mostra un messaggio relativo alla causa dell’errore. | |
| Scenario/ Flusso di errore: Non è possibile avviare il Web Server | | | |
| 2 .1 | Sistema | Mostra un messaggio relativo alla causa dell’errore. | |

### 

### 

### 3.7.2 Shut-down

|  |  |  |  |
| --- | --- | --- | --- |
| Identificativo  UC\_SD\_1 | Terminazione del sistema | Data | 30/11/2022 |
| Vers. | 0.1 |
| Autore | Alessandro Clericuzio  Vittorio Contardo |
| Descrizione | Definisce la funzionalità di terminazione del sistema. | | |
| Attore principale | Amministratore della piattaforma.  Termina il sistema. | | |
| Attori secondari | N/A | | |
| Entry Condition | L’amministratore ha accesso al dispositivo su cui è avviato il sistema. | | |
| Exit condition  On success | Il sistema termina. | | |
| Exit condition  On failure | Il sistema non termina. | | |
| Rilevanza/User priority | Elevata | | |
| Frequenza stimata | 1/giorno | | |
| Extension point | N/A | | |
| Generalization of | N/A | | |
| Flusso di eventi principale/ Main scenario | | | |
| 1 | Amministratore | Chiede la terminazione del Web Server. | |
| 2 | Amministratore | Chiede la terminazione dei Server relativi ai microservizi. | |
| 3 | Amministratore | Chiede la terminazione del servizio MongoDB. | |
| 4 | Sistema | Termina il Web Server. | |
| 5 | Sistema | Termina i Server. | |
| 6 | Sistema | Termina il servizio MongoDB. | |
| 7 | Sistema | Comunica la corretta terminazione. | |
| Scenario/ Flusso di eventi di errore: Non è possibile terminare il Web Server | | | |
| 4.1 | Sistema | Mostra un messaggio relativo alla causa dell’errore. | |
| Scenario/ Flusso di errore: Non è possibile terminare i Server | | | |
| 5.1 | Sistema | Mostra un messaggio relativo alla causa dell’errore. | |
| Scenario/ Flusso di errore: Non è possibile terminare il servizio MongoDB | | | |
| 6.1 | Sistema | Mostra un messaggio relativo alla causa dell’errore. | |

### 3.7.3 Fallimenti

ChemioPlan può incorrere a fallimenti riguardanti sia Hardware che Software:

1. Fallimenti Hardware:
   1. Spegnimento improvviso della macchina su cui sono avviati i Server: il sistema non prevede nessuna strategia di backup o ripristino dei dati.
2. Fallimenti Software:
   1. Crash improvviso di un Server relativo ad un microservizio: l’unico processo che può essere eseguita è il riavvio del Server.

# 

# 4. Servizi dei sottosistemi

|  |  |
| --- | --- |
| Utente | |
| Servizio | Descrizione |
| Log-in | Permettere agli utenti di Accedere al sistema |
| Log-out | Permette all’utente di uscire dal Sistema |
| Sing-Up | Permettere la registrazione degli utenti da parte dell’amministratore |

|  |  |
| --- | --- |
| Data Access (Utente) | |
| Servizio | Descrizione |
| Visualizza Utente | Restituisce tutti i dati dell’utente selezionato |
| Inserisci Utente | Inserisce l’utente nel DB |
| Modifica Utente | Aggiorna i dati nel DB dell’utente selezionato |
| Elimina Utente | Elimina l’utente selezionata dal DB |

|  |  |
| --- | --- |
| Paziente | |
| Servizio | Descrizione |
| Aggiunta | Aggiungere un nuovo paziente |
| Cambio Terapia | Cambiare medicina e terapia a un paziente |
| Visualizza Paziente | Visualizza la Scheda di un paziente |

|  |  |
| --- | --- |
| Data Access (Paziente) | |
| Servizio | Descrizione |
| Caricamento Scheda Paziente | Restituisce tutti i dati del paziente con lo stesso codice fiscale dato dall’utente |
| Inserimento Scheda paziente | Inserisce la scheda paziente nel DB |
| Aggiorna dati Scheda Paziente | Aggiorna I dati nel DB della Scheda paziente selezionata |
| Elimina Scheda Paziente | Elimina la Scheda paziente selezionata dal DB |
| Visualizza Schede Paziente | Restituisce tutte le Schede presenti nel DB |

|  |  |
| --- | --- |
| Data Access (Malattia) | |
| Servizio | Descrizione |
| Visualizza Malattia | Restituisce i dati della malattia selezionata |

|  |  |
| --- | --- |
| Farmacia | |
| Servizio | Descrizione |
| Visualizza farmaci in magazzino | Visualizza tutti i farmaci presenti in magazzino |
| Visualizza farmaco | Visualizza il singolo farmaco |
| Inserimento lotto farmaco già registrato | inserisce un lotto di farmaci già registrato |
| Inserimento nuovo lotto farmaco | Inserisce un nuovo lotto di farmaci non registrato |
| Ordine farmaco | Ordina un nuovo lotto di farmaci |
| Modifica farmaco | Modifica i dati relativi ad un farmaco |

|  |  |
| --- | --- |
| Data Access (Farmaco) | |
| Servizio | Descrizione |
| Inserimento Scheda Farmaco | Inserisce la scheda farmaco nel DB |
| Modifica Scheda Farmaco | Aggiorna i dati passati dal chiamante nel DB |
| Elimina Scheda Farmaco | Elimina la scheda farmaco selezionato dal DB |
| Visualizza Scheda Farmaco | Restituisce i dati della scheda farmaco selezionata |

|  |  |
| --- | --- |
| Prenotazione | |
| Servizio | Descrizione |
| Nuova Prenotazione | Crea una nuova prenotazione |
| Elimina Prenotazione | Elimina la prenotazione selezionata |
| Visualizza Prenotazione | Stampa a video tutti i dati della prenotazione |
| Modifica Prenotazione | Modifica prenotazione |

|  |  |
| --- | --- |
| Data Access (Prenotazione) | |
| Servizio | Descrizione |
| Inserimento Prenotazione | Inserisce la prenotazione le DB |
| Modifica Prenotazione | Aggiorna i dati passati dal chiamante nel DB |
| Elimina Prenotazione | Elimina la prenotazione selezionata dal DB |
| Visualizza Prenotazione | Restituisce i dati della prenotazione selezionata |

|  |  |
| --- | --- |
| Ottimizzazione Scheduling | |
| Servizio | Descrizione |
| Suggerimento data | Suggerisce una data per una nuova prenotazione |

|  |  |
| --- | --- |
| Data Access (Ottimizzazione Scheduling) | |
| Servizio | Descrizione |
| Visualizza prenotazione | Restituisce i dati presenti nel DB di una determinata prenotazione |
| Inserisci prenotazione | Inserisce una nuova prenotazione nel DB |
| Modifica prenotazione | Modifica i dati nel DB di una specifica prenotazione |
| Elimina prenotazione | Elimina dal DB una specifica prenotazione |

|  |  |
| --- | --- |
| Ottimizzazione Acquisto | |
| Servizio | Descrizione |
| Suggerimento quantità | Suggerisce una quantità ottimale da ordinare di un determinato farmaco |

|  |  |
| --- | --- |
| Data Access (Ottimizzazione Acquisto) | |
| Servizio | Descrizione |
| Visualizza ordine | Restituisce i dati presenti nel DB di un determinato ordine |
| Inserisci ordine | Inserisce un nuovo ordine nel DB |
| Modifica ordine | Modifica i dati nel DB di uno specifico ordine |
| Elimina ordine | Elimina dal DB uno specifico ordine |

# 

# 5. Glossario

*DB*: DataBase

*DBMS*: sistema software per la gestione dei dati persistenti

*MongoDB*: specifico DBMS

*SHA-1*: tecnica di criptazione

*DG*: Design Goals