



Laurea Triennale in Informatica-Università di Salerno Corso di
Ingegneria del Software- Prof C. Gravino

System Design Document

MediCare

Riferimento	
Versione	1.1
Data	02/02/2024
Destinatario	C. Gravino
Presentato da	Andrea Gisolfi, Giacomo Favale, Giovanni Nigro, Antonio Merola
Approvato da	



Laurea Triennale in Informatica-Università di Salerno Corso di
Ingegneria del Software- Prof C. Gravino

Team Composition

Ruolo	Nome	Acronimo	Contatti
Team Member	Andrea Gisolfi	AG	a.gisolfi4@studenti.unisa.it
Team Member	Giacomo Favale	GF	g.favale1@studenti.unisa.it
Team Member	Giovanni Nigro	GN	g.nigro32@studenti.unisa.it
Team Member	Antonio Merola	AM	a.merola29@studenti.unisa.it



Sommario

1. Introduzione.....	5
1.1 Scopo del sistema	5
1.2 Design Goals.....	5
1.3 Definizioni, acronimi e abbreviazioni	8
1.4 Riferimenti.....	9
1.5 Organizzazione	9
2. Architettura del sistema corrente	9
3. Architettura del sistema proposto	10
3.1 Panoramica	10
3.2 Decomposizione in sottosistemi.....	10
3.3 Mapping hardware/software	12
3.4 Gestione dei dati persistenti	13
3.5 Controllo degli accessi e sicurezza	16
3.6 Controllo globale del software.....	17
3.7 Condizioni limite	17
4. Servizio dei sottosistemi	21
5. Glossario.....	23



Revision History

Data	Versione	Descrizione	Autori
18/11/2023	0.1	Prima stesura	AG, GF, GN, AM
26/11/2023	0.2	Definiti Design Goals e Design trade-off	AG, GF, GN, AM
26/11/2023	0.2	Definita la decomposizione in sottosistemi	GN,GF
26/11/2023	0.2	Aggiunto diagramma di decomposizione	AG, AM
27/11/2023	0.3	Aggiunta matrice degli accessi	AG, AM
02/12/2023	0.4	Aggiunto diagramma architetturale	GF, GN
02/12/2023	0.4	Definito mapping H/S	GF, GN
02/12/2023	0.4	Definita la gestione dei dati persistenti	AG, AM
02/12/2023	0.4	Identificati i boundary condition	AG, GF, GN, AM
02/12/2023	0.4	Stesura paragrafo 3.1 relativo alla panoramica del sistema	AG, GF, GN, AM
17/12/2023	1.0	Revisione	AG, GF, GN, AM
28/01/2023	1.1	Revisione Finale	AG, GF, GN, AM



1. Introduzione

1.1 Scopo del sistema

Lo star bene e in buona salute sono da sempre alcune tra le top priority della popolazione. Proprio in questi ultimi anni, con l'emergenza COVID-19, abbiamo visto come una cattiva gestione degli ambienti pubblici e della sanità generale possa portare al totale caos. Proprio questa è una tra le prerogative di MediCare, che si pone come obiettivo quello di fornire un sistema efficiente e intuitivo che sia in grado di fornire utilità sia ai più giovani che ai più anziani. Anche avvicinare alla sanità è uno dei punti chiave di Medicare, obiettivo che tenta di raggiungere fornendo una interfaccia semplice ed intuitiva per tutte le età.

1.2 Design Goals

In questo paragrafo vengono esposti i Design Goals, i quali descrivono le qualità su cui il sistema deve essere concentrato (quelle che devono essere ottimizzate) e si stabiliscono le loro priorità. Ogni importante decisione di design deve essere fatta seguendo lo stesso insieme di criteri; sviluppando in maniera corretta i Design Goal sarà possibile definire un percorso di sviluppo solido e dettagliato.

Nella tabella che segue, i design goals sono descritti in base ai seguenti campi, ognuno corrispondente a una colonna:

- **Rank:** viene indicata la priorità rispetto agli altri design goal;
- **ID Design Goal:** è un identificativo univoco del design goal;
- **Descrizione:** fornisce una breve descrizione delle caratteristiche;
- **Categoria:** categoria a cui i design goals appartengono;
- **RNF Origine:** il requisito non funzionale che ha generato il design goal;

È importante osservare che ogni design goal è stato organizzato secondo:

- **Performance:** indica il livello di prestazioni del sistema software, in termini di spazio e velocità;
- **Dependability:** relativa all'affidabilità del sistema (quantifichiamo lo sforzo che deve essere speso per rendere minimi gli eventuali crash di sistema, e le loro conseguenze - gestione degli errori, resistenza ai crash);
- **Cost:** requisiti sui costi per sviluppare il sistema, metterlo in



funzione e amministrarlo;

- **Maintenance:** indica quanto sforzo è necessario per effettuare modifiche al sistema dopo la prima release;
- **End user:** inerente ai canoni dell'interazione uomo-macchina (quindi qualità che sono effettivamente desiderabili dal POV dell'utente) che non sono stati ricoperti dai criteri di dependability e performance.

Rank	ID Design Goal	Descrizione	Categoria	RNF di origine
3	DG_1 Tempo di risposta della mappa	Il sistema deve fornire una rappresentazione della zona sulla mappa entro 4 secondi.	Performance	RNF_P1
7	DG_2 Robustezza	Il sistema deve essere in grado di supportare un grande numero di utenti senza rallentamenti e senza crash.	Performance	RNF_A1
2	DG_3 Tempo di risposta del sistema	Il sistema deve essere interattivo e impiegare non più di 6 secondi per rispondere agli input dell'utente.	Performance	RNF_P3
9	DG_4 Reperibilità	Il sistema deve essere disponibile nell'orario compreso tra le 2:00 fino alle 00:00. Le due ore non garantite possono essere dedicate alla manutenzione.	Dependability	RNF_P2
1	DG_5 Sicurezza	Il sistema deve garantire la sicurezza seguendo le norme del GDPR.	Dependability	RNF_L1
10	DG_6 Affidabilità	Il sistema deve tornare ad essere disponibile entro 3 ore dall'identificazione di un guasto.	Dependability	RNF_A2
12	DG_7 Accessibilità	Il sistema deve essere compatibile e in grado di funzionare su qualsiasi sistema operativo del campo smartphones.	Maintenance	RNF_U2



6	DG_8 Manutenibilità	Il sistema, previa l'utilizzo di una architettura precisa, la documentazione e i commenti, risulta essere chiaro e comprensibile.	Maintenance	RNF_S1
8	DG_9 Estendibilità	Il sistema risulta essere strutturato in maniera tale da consentire una facile estensione delle sue funzionalità	Maintenance	RNF_S2
5	DG_10 Costo di Upgrade	Eventuali aggiornamenti ed implementazioni del sistema non superano il costo di upgrade; inoltre, si effettua la verifica al fine di assicurarsi che il costo stesso dell'upgrade valga i benefici ottenuti.	Cost	RNF_S3
4	DG_11 Costi di sviluppo	Il sistema riesce ad implementare tutte le funzionalità senza superare il budget impostato	Cost	RNF_S4
11	DG_12 Usabilità	Il sistema deve risultare comprensibile e di facile uso per gli utenti.	End User	RNF_U1

Design Trade-off

Usabilità vs Accessibilità: Si predilige un sistema che sia facile da utilizzare e intuitivo piuttosto che avere un sistema distribuito su vari sistemi operativi ma che risulta di difficile utilizzo.

Tempo di rilascio vs Robustezza: Si desidera dare priorità al tempo di rilascio in quanto, messo in funzione il sistema, si può analizzare l'affluenza degli utenti ed eventualmente introdurre altre macchine fisiche, rendendo l'intero sistema più robusto.

Tempo di rilascio vs Sicurezza: Per avere un sistema che sia effettivamente funzionante e non metta a rischio gli utenti, risulta fondamentale dare priorità



alla sicurezza stessa del sistema.

Manutenibilità vs Estendibilità: Si vuole dare priorità alla struttura del codice, facendo sì che il esso sia chiaro e preciso. Inoltre, avere chiarezza nel codice porta anche ad una più semplice comprensione ed estendibilità delle funzionalità.

Costi di sviluppo vs Tempo di rilascio: Si preferisce riuscire a rimanere nel budget impostato, rinviando l'eventuale pubblicazione del sistema.

Trade off	
Usabilità	Accessibilità
Tempo di rilascio	Robustezza
Sicurezza	Tempo di rilascio
Manutenibilità	Estendibilità
Costi di sviluppo	Tempo di rilascio

1.3 Definizioni, acronimi e abbreviazioni

Di seguito è riportata una lista di acronimi ed eventuali abbreviazioni utilizzati nella documentazione:

- **SDD:** System Design Document
- **RAD:** Requirement Analysis Document
- **Dati persistenti:** Dati che persistono anche dopo che è terminata l'esecuzione del programma che li ha creati
- **Design Goal:** Qualità sulle quali il sistema in questione deve essere concentrato
- **Sottosistema:** Sottoinsieme dei servizi del “dominio applicativo” del sistema, costituito dai servizi legati da un qualche tipo di relazione funzionale
- **End User:** Colui o colei che acquista il prodotto e lo usa
- **Requisito non funzionale (RNF):** Anche detti requisiti di qualità, sono dei vincoli e delle proprietà che possono essere usati per giudicare il funzionamento del sistema



1.4 Riferimenti

Sono state prese in considerazione le seguenti fonti per la realizzazione della presente documentazione:

1. Dispense presentate nel corso di Ingegneria del Software, tenuto dal prof. Carmine Gravino, fornite mediante Piattaforma E-learning del Corso di Laurea in Informatica dell'Università degli Studi di Salerno;
2. Libro di testo "Object Oriented Software Engineering Using UML Patterns and Java Prentice Hall 2010 Bernd Bruegge Allen H.Dutoit"
3. Libro di testo "C. GHEZZI, D. MANDRIOLI, M. JAZAYERI, INGEGNERIA DEL SOFTWARE – FONDAMENTI E PRINCIPI, PRENTICE HALL, 2004"

1.5 Organizzazione

Il seguente documento è stato strutturato in modo tale da fornire in maniera chiara una breve introduzione, illustrandone poi le problematiche e presentando il sistema. È suddiviso in:

- **Introduzione:** viene descritto in maniera generale lo scopo e gli obiettivi di design che si intende raggiungere.
- **Architettura software corrente:** viene descritto lo stato attuale di una architettura presente cui MediCare si ispira.
- **Architettura software proposta:** viene descritto come il sistema sarà diviso e definito in eventuali sottosistemi, con i relativi mapping hardware/software e la gestione dei dati persistenti (databases).
- **Servizi dei sottosistemi:** viene riportata una descrizione dei servizi di ogni sottosistema, presentando la struttura di ciascuno di essi e le relative boundary conditions riguardanti l'intero sistema.
- **Glossario:** sezione dedicata alla spiegazione di eventuali termini tecnici utilizzati nel documento.

2. Architettura del sistema corrente

L'architettura di una delle applicazioni più simili a MediCare è "Campania in Salute", dove è possibile prenotare visite mediche in Campania; essa è un software per piattaforme Android/iOS/iPadOS.

Questo sistema, a differenza di MediCare, non consente né una ricerca



basata su località degli ospedali disponibili. Inoltre, il sistema tenuto in considerazione non consente di ricevere una diagnosi preliminare e una gravità in base alla diagnosi stessa.

3. Architettura del sistema proposto

3.1 Panoramica

Il sistema MediCare da noi proposto è basato sull'architettura Three-Tier, utilizzando come linguaggio di programmazione Java, ed il framework JavaFX; per il modulo di intelligenza artificiale è stato usato invece il linguaggio Python. Questo modello architetturale è stato scelto perché ha il vantaggio di garantire una separazione logica delle componenti software, in modo da aumentarne la manutenibilità, la scalabilità ed il riuso del codice. Come linguaggi e framework adottati per lo sviluppo del sistema troviamo:

- **Java** per lo sviluppo del back-end;
- **Python** per lo sviluppo del modulo di intelligenza artificiale;
- **JavaFX** per lo sviluppo dell'interfaccia grafica;
- **JDBC** per il collegamento al database;
- **MySQL** per l'interazione e la gestione del database.

3.2 Decomposizione in sottosistemi

Sono stati individuati i seguenti sottosistemi:

- **Registrazione:** si occupa di definire e gestire la registrazione dell'ospite al sistema.
- **Login:** si occupa di definire e gestire le funzionalità di accesso dell'utente al sistema.
- **Area Personale:** si occupa di definire e fornire le funzionalità di disconnessione dell'account, visualizzazione delle informazioni personali inserite nella fase di registrazione, modifica delle informazioni e l'eventuale possibilità di cancellazione dell'account.
- **Cerca Ospedali:** è responsabile di fornire le funzionalità di ricerca di ospedali vicini ad una certa località, e di ospedali vicini alla posizione dell'utente.
- **Prenotazione:** si occupa di gestire ogni funzionalità collegata alle prenotazioni; gestisce quindi la funzionalità di modifica prenotazione, creazione prenotazione, visualizza prenotazione e cancella prenotazione.

- **Chatbot:** è responsabile della gestione dell'interazione tra il chatbot e l'utente; fornisce le funzioni di diagnosi preliminare in base ai sintomi del paziente e di una valutazione della gravità della stessa.
- **Storage:** è responsabile della gestione di tutti i dati persistenti con un database.
- **Storage Connection:** si occupa dell'interazione tra i vari sottosistemi ed il sottosistema di Storage.

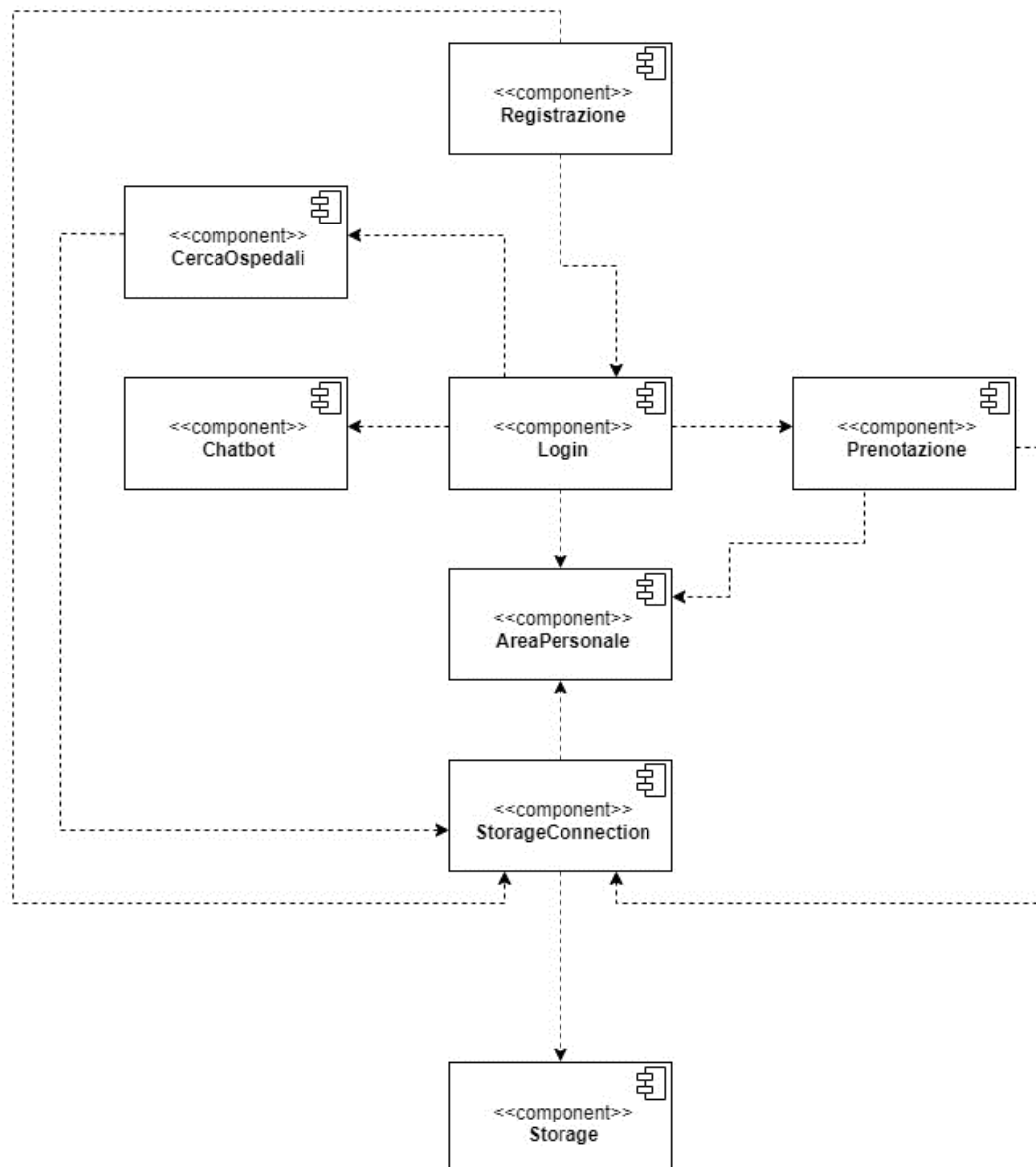
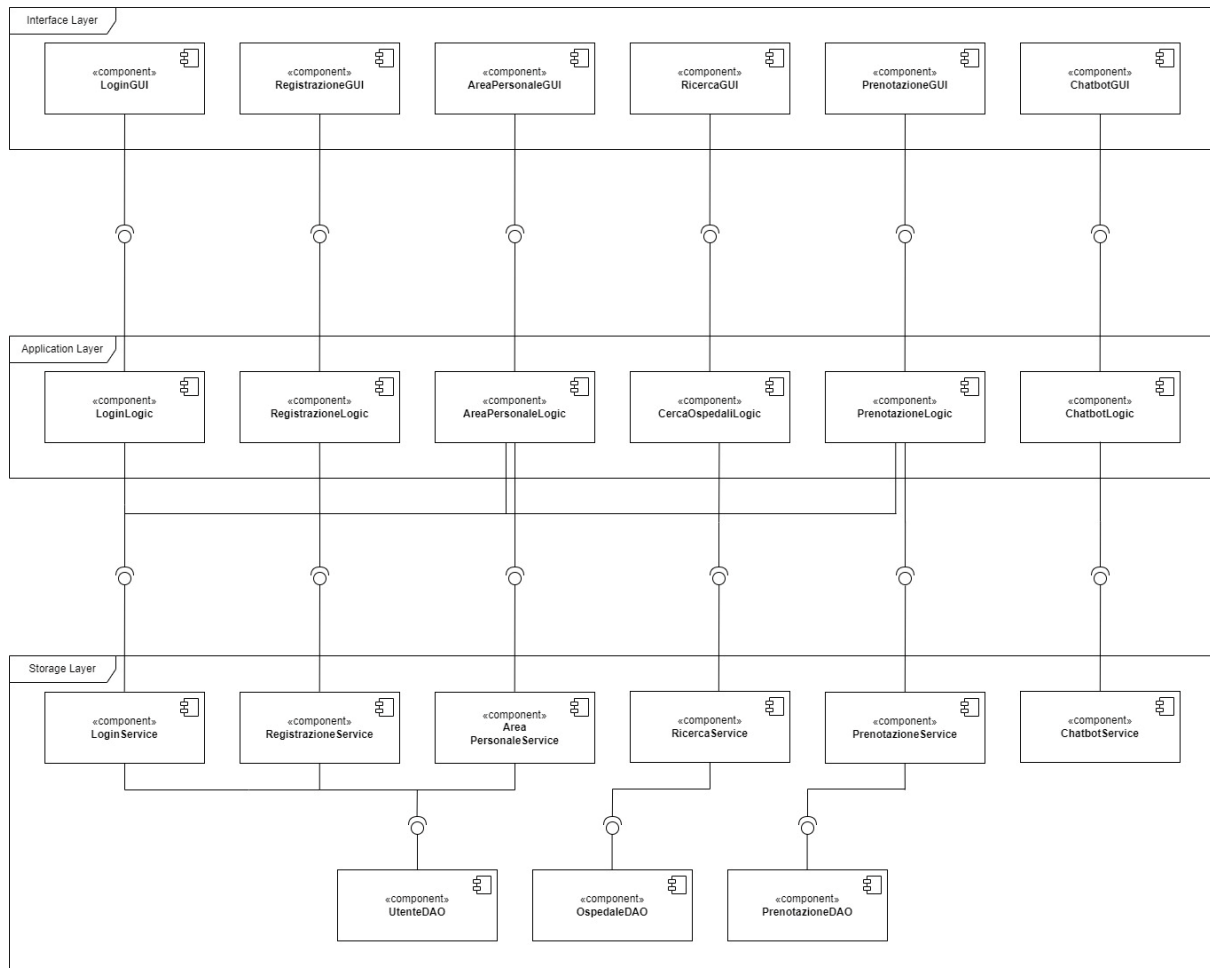


Diagramma architetturale



Il diagramma architetturale presentato mostra la suddivisione three-tier. In particolare, la suddivisione presenta:

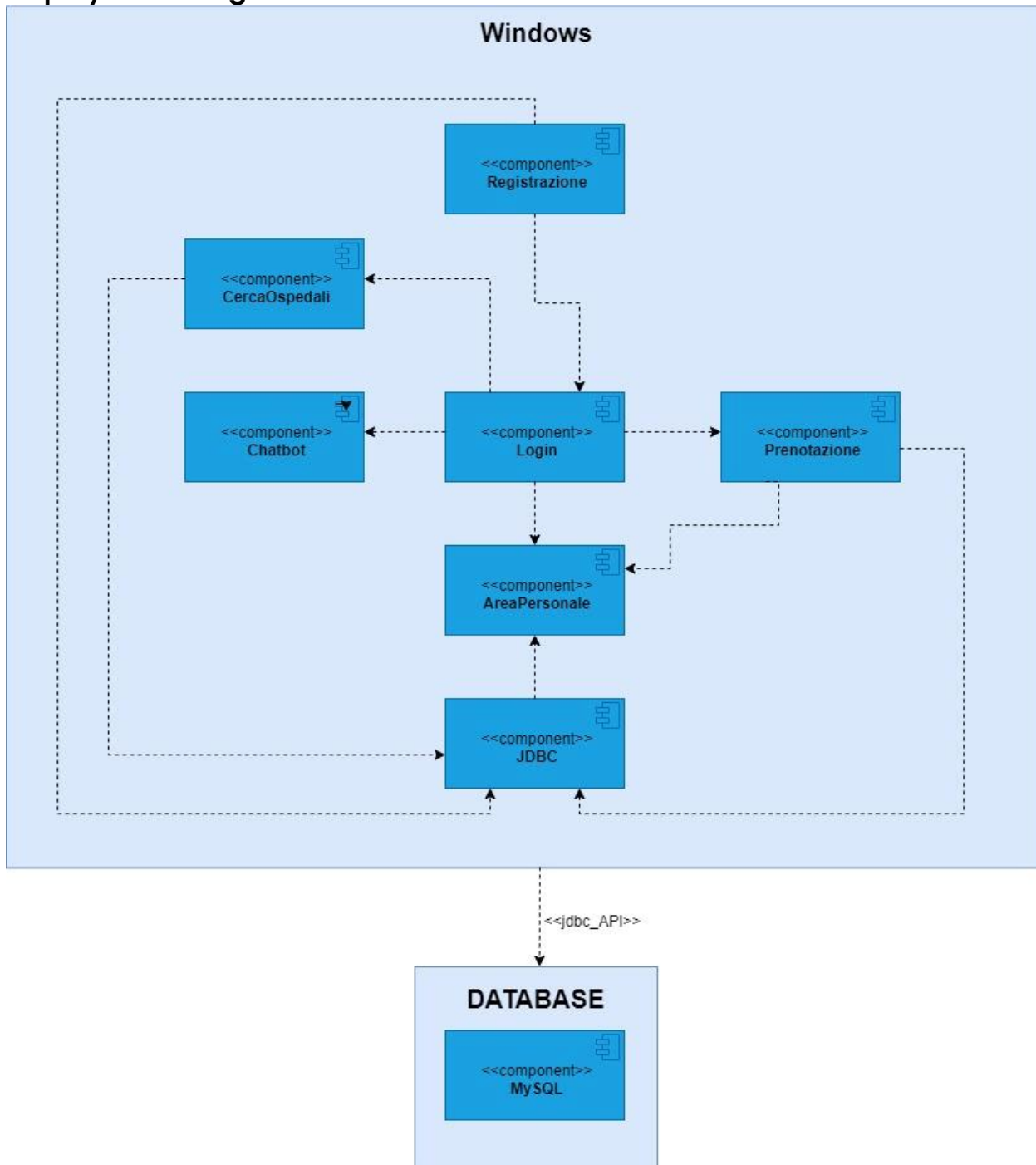
- **Interface Layer:** layer che si occupa della presentazione e interazione diretta con l'utente;
- **Application Layer:** layer applicativo che si occupa della gestione della logica del sistema
- **Storage Layer:** layer che si occupa della gestione dei dati persistenti oltre all'interfaccia tra il sistema e il database

3.3 Mapping hardware/software

La piattaforma che verrà sviluppata sarà una app desktop mirata agli utenti dei dispositivi fissi e portatili con sistema operativo Windows, aventi una connessione ad Internet. I dati sono memorizzati su un database con cui

l'applicazione interagisce mediante API. Di seguito è riportato l'UML
Deployment Diagram che descrive il mapping hardware/software:

Deployment Diagram:

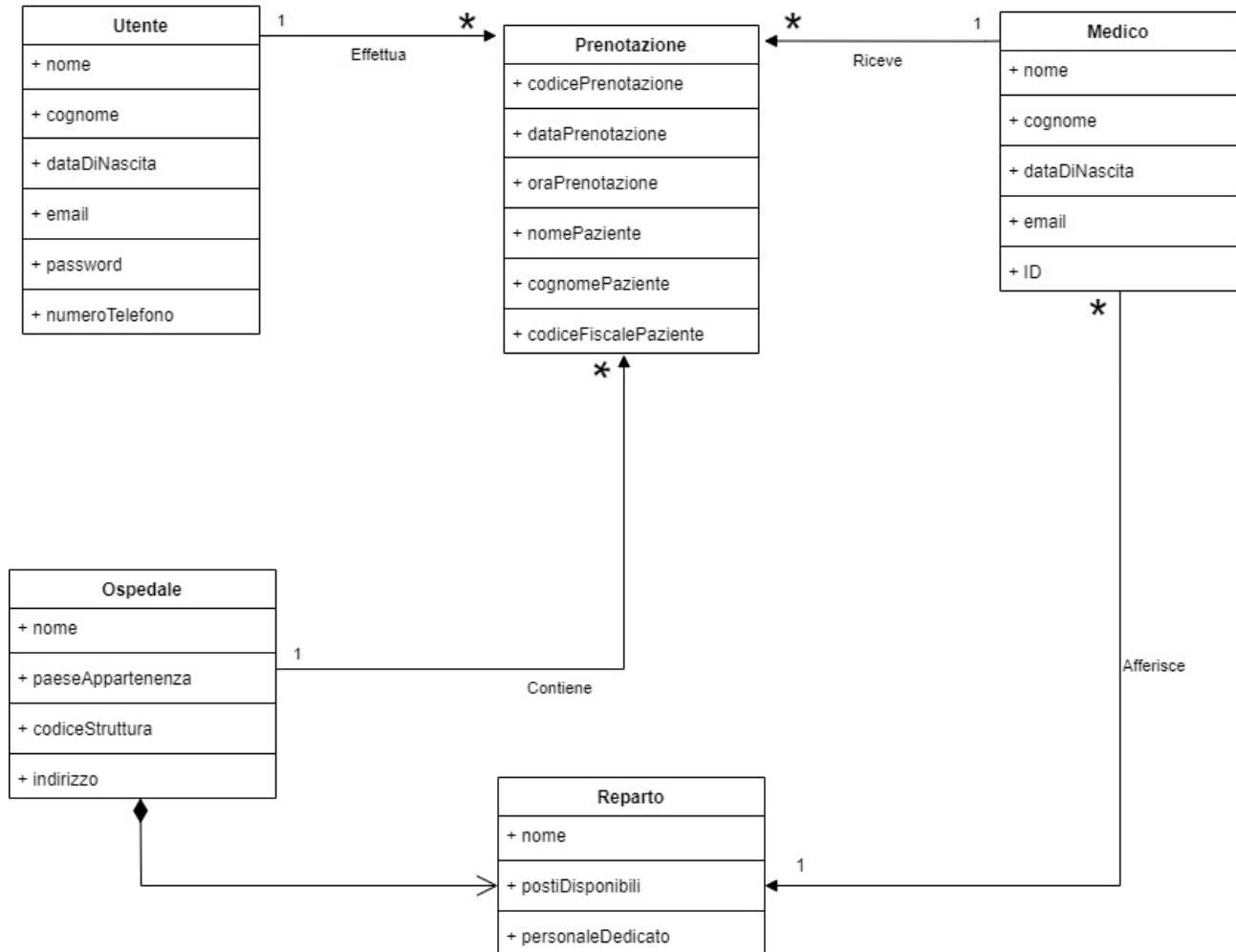


3.4 Gestione dei dati persistenti

Per la gestione dei dati persistenti, il team di MediCare ha deciso di utilizzare il DBMS messo a disposizione dalla **Oracle Corporation**, ovvero **MySQL**, un

sistema di gestione di database relazionali.

Schema concettuale:



Schema relazionale:

Utente(email, nome, cognome, dataDiNascita, password, numeroDiTelefono);

Ospedale(codiceStruttura, nome, paeseAppartenenza, indirizzo);

Medico(ID, email, nome, cognome, dataDiNascita, nomeReparto, codiceStruttura);

Prenotazione(codicePrenotazione, dataPrenotazione, oraPrenotazione, nomePaziente, cognomePaziente, codiceFiscalePaziente, emailUtente, IDMedico, codiceStruttura)

Reparto(nome, codiceStruttura, postiDisponibili, personaleDedicato)



Dizionario dei dati:

Nome entità		Utente	
Descrizione		Contiene i dati relativi all'utente registrato al sistema.	
Nome campo	Tipo	Vincolo di chiave	Altri vincoli
Email	Varchar(100)	PRIMARY KEY	NOT NULL
Nome	Varchar(20)		NOT NULL
Cognome	Varchar(20)		NOT NULL
DataDiNascita	Date		NOT NULL
Password	Varchar(50)		NOT NULL
NumeroDiTelefono	Varchar(10)		NOT NULL

Nome entità		Ospedale	
Descrizione		Contiene i dati relativi ad un ospedale.	
Nome campo	Tipo	Vincolo di chiave	Altri vincoli
CodiceStruttura	Int	PRIMARY KEY	NOT NULL
Nome	Varchar(100)		NOT NULL
PaeseAppartenenza	Varchar(50)		NOT NULL
Indirizzo	Varchar(100)		NOT NULL

Nome entità		Prenotazione	
Descrizione		Contiene i dati relativi a una prenotazione effettuata da un utente.	
Nome campo	Tipo	Vincolo di chiave	Altri vincoli
CodicePrenotazione	Int	PRIMARY KEY	NOT NULL
DataPrenotazione	Date	PRIMARY KEY	NOT NULL
OraPrenotazione	Varchar(10)	PRIMARY KEY	NOT NULL
nomePaziente	Varchar(20)		NOT NULL
cognomePaziente	Varchar(20)		NOT NULL
codiceFiscalePaziente	Varchar(16)		NOT NULL
EmailUtente	Varchar(100)	FOREIGN KEY	NOT NULL
IDMedico	Int	PRIMARY KEY, FOREIGN KEY	NOT NULL
CodiceStruttura	Int	FOREIGN KEY	NOT NULL



Nome entità		Medico	
Descrizione		Contiene i dati relativi ad un medico.	
Nome campo	Tipo	Vincolo di chiave	Altri vincoli
ID	Int	PRIMARY KEY	NOT NULL
Nome	Varchar(20)		NOT NULL
Cognome	Varchar(20)		NOT NULL
Email	Varchar(100)		NOT NULL
DataDiNascita	Date		NOT NULL
NomeReparto	Varchar(40)	FOREIGN KEY	NOT NULL
CodiceStruttura	Int	FOREIGN KEY	NOT NULL

Nome entità		Reparto	
Descrizione		Contiene i dati relativi ad un reparto ospedaliero a cui un medico afferisce.	
Nome campo	Tipo	Vincolo di chiave	Altri vincoli
Nome	Varchar(40)	PRIMARY KEY	NOT NULL
CodiceStruttura	Int	PRIMARY KEY, FOREIGN KEY	NOT NULL
PostiDisponibili	Int		NOT NULL
PersonaleDedicato	Int		NOT NULL

3.5 Controllo degli accessi e sicurezza

In questa sezione viene riportata la matrice degli accessi, in modo tale da tracciare degli attori che possono accedere ad i vari servizi offerti dal sistema.

	Attori	Utente	Ospite
Oggetti			
Registrazione			RegistrazioneUtente
Login		LoginUtente	
Area Personale		Logout VisualizzaDatiPersonali ModificaDatiPersonali EliminaAccount	



Prenotazione	NuovaPrenotazione VisualizzaPrenotazione ModificaPrenotazione EliminaPrenotazione	
Chat-Bot	NuovaDiagnosi GravitàDiagnosi	
Cerca Ospedali	RicercaTramiteLocalità RicercaTramiteMappa	

3.6 Controllo globale del software

Il sistema MediCare rappresenta un ambiente interattivo dove ogni funzionalità viene avviata mediante un comando, che viene fornito dall'utente tramite un'interfaccia grafica. Quando l'utente desidera accedere ed utilizzare una specifica funzionalità, può farlo tramite la GUI, che seleziona il controllo corrispondente. Questa azione genera un evento gestito dal relativo handler, il quale dirige il flusso degli eventi al sottosistema responsabile della logica di controllo e di gestione il quale, successivamente, si collega ai servizi per la logica applicativa. Il controllo di flusso usato in MediCare risulta essere di tipo procedure-driven, essendo MediCare stessa una app lanciata dall'Utente.

3.7 Condizioni limite

Vengono presentate e riportate le varie boundary conditions relative all'avvio del sistema, allo spegnimento del sistema, al fallimento del sistema ed agli errori di accesso a dati persistenti.

Avvio del Sistema

Identificativo:	UC_LC1 - Avvio del sistema	Data:	02/12/2023
		Vers.:	0.1
		Autore:	GN
Descrizione	Lo UC consente l'avvio del sistema		
Attore Principale	Amministratore		
Attori secondari	NA		
Entry Condition	L'amministratore effettua l'accesso al Server		
Exit condition On success	Il sistema viene avviato correttamente		



Exit condition On failure		Il sistema non viene avviato correttamente
FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO		
1	Amministratore:	L'amministratore avvia il sistema mediante l'apposita sequenza di operazione.
2	Sistema:	Il sistema risponde alle operazioni, rendendo disponibili le sue funzionalità e i suoi servizi.
I Scenario/Flusso di eventi ERRORE: le componenti del sistema non si avviano correttamente		
2.e1	Sistema:	Il sistema avvisa l'amministratore dell'errato avvio.
2.e2	Amministratore:	L'amministratore corregge le problematiche sorte
2.e.3	Amministratore:	L'amministratore ritenta l'avvio del sistema.

Spegnimento del sistema

Identificativo:	UC_LC2 - Spegnimento del sistema	Data:	02/12/2023
		Vers.:	0.1
		Autore:	AG
Descrizione	Lo UC consente lo spegnimento del sistema		
Attore Principale	Amministratore		
Attori secondari	NA		
Entry Condition	L'amministratore ha effettuato il login AND Il sistema è stato avviato AND Il sistema funziona correttamente AND Il sistema non è già spento		
Exit condition On success	Il sistema viene spento correttamente		
Exit condition On failure	Il sistema non viene spento correttamente		
FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO			
1	Amministratore:	L'amministratore invia il comando di spegnimento al sistema.	
2	Sistema:	Il sistema riceve il comando, esegue le eventuali operazioni di salvataggio, e procede allo spegnimento.	
I Scenario/Flusso di eventi ERRORE: il sistema non riesce a eseguire lo spegnimento			
2.e1	Sistema:	Il sistema non riesce ad effettuare lo spegnimento.	
2.e2	Amministratore:	L'amministratore corregge le problematiche sorte.	
2.e.3	Amministratore:	L'amministratore ritenta lo spegnimento del sistema.	



Fallimento del sistema

Identificativo:		UC_LC3 - Fallimento del sistema	Data:	02/12/2023
			Vers.:	0.1
			Autore:	GF
Descrizione		Lo UC consente di individuare eventuali fallimenti determinando il comportamento da seguire.		
Attore Principale		Amministratore		
Attori secondari		NA		
Entry Condition		Il sistema è in funzione AND Il sistema, improvvisamente, arresta le sue funzioni.		
Exit condition On success		Il sistema viene riavviato post correzione della problematica		
Exit condition On failure		Il sistema non viene riavviato correttamente.		
FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO				
1	Amministratore:	L'amministratore verifica il crash del sistema e analizza la problematica		
2	Amministratore:	L'amministratore risolve la problematica e procede al riavvio del sistema		
I Scenario/Flusso di eventi ERRORE: il sistema non riesce a eseguire il riavvio				
2.e1	Amministratore:	L'amministratore riscontra il crash e prova a riavviare il sistema.		
2.e2	Sistema:	Il sistema non riesce ad avviarsi e comunica l'impossibilità, procedendo allo spegnimento.		

Errori di accesso ai dati persistenti

Identificativo:	UC_LC4 - Errori di accessi ai dati persistenti	Data:	02/12/2023
		Vers.:	0.1
		Autore:	AM
Descrizione	Lo UC definisce il comportamento da applicare dal sistema quando o vi sono problemi all'integrità dei dati persistenti oppure il sistema non riesce ad accedere alla base dati.		
Attore Principale	Amministratore		
Attori secondari	NA		
Entry Condition	Il sistema è in funzione AND Il sistema non riesce ad accedere ai dati persistenti OR I dati persistenti sono corrotti		
Exit condition On success	Il sistema riesce ad accedere ai dati persistenti		
Exit condition On failure	Il sistema non riesce ad accedere ai dati persistenti		
FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO			



1	Sistema:	Il sistema comunica all'operatore che ci sono problemi nella lettura o nell'accesso ai dati persistenti.
2	Sistema:	Il sistema termina il funzionamento con un errore.
3	Amministratore:	L'amministratore analizza la problematica, la risolve e riavvia il sistema
4	Sistema:	Il sistema riprende la sua corretta funzione.
I Scenario/Flusso di eventi ERRORE: Il sistema, al riavvio, non riesce ad accedere ai dati		
4.e1	Amministratore:	L'amministratore si occupa di effettuare un controllo approfondito delle problematiche
4.e2	Amministratore:	L'amministratore trova le problematiche relative all'integrità sia dei dati che della base dati, le risolve, ed esegue il riavvio



4. Servizio dei sottosistemi

Qui di seguito vengono riportati i servizi di ogni sottosistema precedentemente individuati ed elencati.

Sottosistema Registrazione

Servizio	Descrizione	Interfaccia
Registrazione Utente	Questa funzionalità rende possibile la registrazione dell'ospite alla piattaforma rendendolo, di conseguenza, Utente.	Registrazione Service

Sottosistema Login

Servizio	Descrizione	Interfaccia
Login Utente	Questa funzionalità permette all'Utente di effettuare il Login alla piattaforma.	Login Service

Sottosistema Area Personale

Servizio	Descrizione	Interfaccia
Visualizzazione dei dati personali	Questa funzionalità permette all'Utente di accedere alla propria pagina personale e visualizzare i propri dati personali.	Area Personale Service
Modifica dei dati personali	Questa funzionalità permette all'Utente di modificare i dati personali, precedentemente inseriti nel sistema in fase di registrazione.	Area Personale Service
Eliminazione dell'account	Questa funzionalità permette all'Utente di cancellare il proprio Account, creato in fase di registrazione.	Area Personale Service
Logout dell'Utente	Questa funzionalità permette all'Utente di effettuare la disconnessione del proprio Account dalla piattaforma.	Area Personale Service



Sottosistema Cerca Ospedali

Servizio	Descrizione	Interfaccia
Ricerca tramite posizione	Questa funzionalità permette di ricercare gli ospedali vicini alla posizione dell'Utente.	Ricerca Service
Ricerca tramite località	Questa funzionalità permette di ricercare gli ospedali vicini ad una certa località fornita dall'Utente.	Ricerca Service

Sottosistema Prenotazioni

Servizio	Descrizione	Interfaccia
Nuova prenotazione	Questa funzionalità permette di creare una nuova prenotazione.	Prenotazioni Service
Modifica prenotazione	Questa funzionalità consente di modificare una prenotazione già esistente.	Prenotazioni Service
Cancella prenotazione	Questa funzionalità consente di eliminare una prenotazione già esistente.	Prenotazioni Service
Visualizza prenotazione	Questa funzionalità consente di visualizzare una prenotazione appena creata o precedentemente creata.	Prenotazioni Service

Sottosistema Chatbot

Servizio	Descrizione	Interfaccia
Ricezione nuova diagnosi	Questa funzionalità consente di ricevere una nuova diagnosi, dopo aver inserito una serie di sintomi.	Chatbot Service
Visione gravità della diagnosi	Questa funzionalità permette di visionare la gravità della diagnosi appena ricevuta.	Chatbot Service



5. Glossario

Termine	Definizione
POV	Acronimo che sta per Point-of-View (punto di vista dell'utente del software)
Prenotazione	Atto in cui una persona si impegna ad occupare una fascia oraria per una visita medica.
Diagnosi	Determinazione, da parte del modulo di intelligenza artificiale, della natura di una malattia in base alla valutazione dei sintomi.
Account	Rappresentazione dell'utente che utilizza l'applicazione.
Utente	Persona fisica che sta utilizzando il sistema MediCare.
MediCare	Nome del software in questione che verrà realizzato.
GUI	Acronimo che sta per Graphic User Interface – Interfaccia mediante il quale l'Utente interagisce con il sistema
DBMS	DataBase Management System