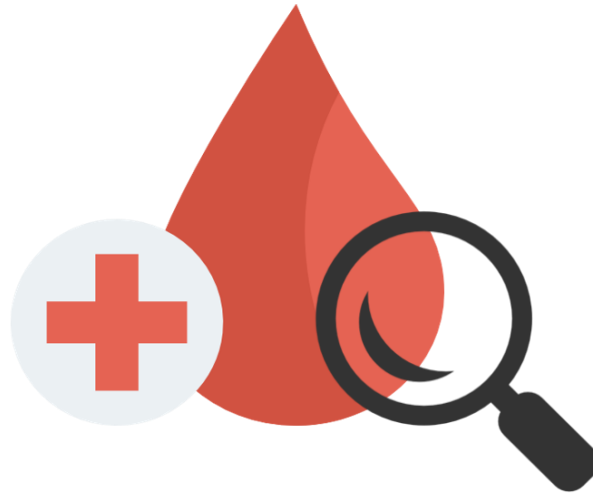




Laurea Magistrale in informatica-Università di Salerno  
Corso di *Gestione dei Progetti Software*- Prof.ssa F.Ferrucci



# System Design Document Digital Donation

Riferimento	
Versione	1.3
Data	2021
Destinatario	Prof.ssa F. Ferrucci
Presentato da	Annamaria Basile, Angela De Martino, Elpidio Mazza, Kevin Pacifico, Mattia Sapere, Fabio Siepe, Marika Spagna Zito
Approvato da	Francesco Abate, Carmine Ferrara



## Revision History

Data	Versione	Descrizione	Autori
26/11/2021	0.1	Mapping Hardware/Software	FS
27/11/2021	0.1	Specifica Design Goals & Trade-offs	FS, KP, MSZ
28/11/2021	0.1	Definizione Architettura del sistema	AB, EM, MS
28/11/2021	0.1	Controlli di accesso e sicurezza	MSZ
29/11/2021	0.1	Gestione dati persistenti	ADM, AB
29/11/2021	0.1	Aggiunta dei punti 1.3, 1.4, 1.5, 3.6	[Tutti]
30/11/2021	0.1	Individuazione dei sottosistemi e relativi servizi	AB, EM, MS
30/11/2021	0.1	Definizione Use Cases per condizioni limite	ADM, KP, FS
02/12/2021	0.1	Integrazione e revisione generale SDD	[Tutti]
02/12/2021	0.1	Aggiornamento Sommario	[Tutti]
02/12/2021	1.0	Approvazione del Documento	FA, CF
10/12/2021	1.1	Aggiornamento dei Design Goals e Revisione di Consistenza dei PM	[Tutti]
13/12/2021	1.2	Aggiornamento dell'architettura del sistema	[Tutti]
13/12/2021	1.2	Aggiornamento dei sottosistemi e relativi servizi	[Tutti]
13/12/2021	1.2	Approvazione del Documento	FA, CF
11/01/2022	1.3	Aggiornamento Gestione Dati Persistenti	KP, MSZ
11/01/2022	1.3	Approvazione del Documento	FA, CF



## Sommario

Revision History .....	2
1. Introduzione .....	4
1.1 Obiettivo del sistema .....	4
1.2 Design Goals .....	5
1.2.1 Design Trade off .....	9
1.3 Definizione Acronimi ed Abbreviazioni .....	9
1.3.1 Definizione Acronimi .....	9
1.3.2 Abbreviazioni .....	10
1.4 Referimenti .....	11
1.5 Panoramica .....	11
2. Architettura dei sistemi simili .....	12
3. Architettura del sistema proposto .....	15
3.1 Panoramica .....	15
3.2 Decomposizione in sottosistemi .....	16
3.2.1 Decomposizione in Layer .....	16
3.2.2 Decomposizione in sottosistemi .....	16
3.3 Mapping HW/SW .....	20
3.4 Gestione Dati persistenti .....	21
3.5 Controllo degli accessi e sicurezza .....	25
3.6 Controllo flusso globale del sistema .....	27
3.7 Definizione Use case per condizioni limite .....	28
3.7.1 Startup del server .....	28
3.7.2 Shutdown del server .....	30
3.7.3 Fallimento .....	31
4. Servizi dei sottosistemi .....	32
5. Glossario .....	34



## 1. Introduzione

---

### 1.1 Obiettivo del sistema

Il sistema che si vuole realizzare nasce dall'esigenza di andare ad automatizzare quelli che sono i processi alla base della donazione del sangue, cercando di velocizzare ed ottimizzare tutte le operazioni necessarie per la donazione del sangue.

Il sistema permette ad un operatore di una specifica sede locale di andare a schedare una nuova seduta di donazione andando ad inviare un invito a tutti i donatori affini alla seduta.

Tutte le sedute possono essere modificate o eliminate entro una certa data dall'operatore, il quale può anche inserire utenti "guest", ovvero utenti che hanno richiesto di partecipare alla seduta di donazione senza essere registrati alla piattaforma.

Una volta avvenuta la donazione l'operatore tramite il sistema può effettuare il salvataggio della donazione, permettendo al sistema di aggiornare sia la tabella della seduta contenente l'elenco dei partecipanti sia il tesserino del donatore.

Nel caso in cui un donatore è stato dichiarato non idoneo a seguito di una visita medica, l'operatore potrà salvare nel sistema l'indisponibilità a donare.

A seguito della richiesta di un donatore, il sistema permette all'operatore la creazione di un tesserino digitale andando a creare automaticamente l'account al donatore.

Il sistema offre un'interfaccia minimale permettendo ad un donatore di prenotarsi ad una seduta in modo semplice e con pochi click.

Un donatore potrà visualizzare il proprio tesserino digitale che contiene sia le informazioni personali sia lo storico delle donazioni effettuate in precedenza.

Inoltre un donatore nel caso in cui non è interessato o impossibilitato a donare per un periodo di tempo potrà compilare in maniera autonoma un'autodichiarazione di indisponibilità ed il sistema automaticamente lo escluderà dalle sedute di donazione.



## 1.2 Design Goals

Rank/Priorità	ID Design Goal	Descrizione design goal	Categoria	Origine	Trade off
<b>14/Media</b>	DG_1.1 Tempo di risposta	Il sistema deve avere un tempo di risposta inferiore ai 7 secondi e l'installazione del sistema richiede al massimo 15 minuti.	Performance	RNF_PR_02 PR_PA_04	Tempo di risposta vs Robustezza
<b>1/Alta</b>	DG_1.2 Throughput	I picchi di carico, fino a circa 1000 utenti simultaneamente collegati, devono essere gestiti dal sistema senza rallentamenti, garantendo fluidità e una latenza molto bassa.	Performance	RNF_PR_01	
<b>6/Alta</b>	DG_2.1 Robustezza	Il sistema deve segnalare con un messaggio di errore eventuali dati invalidi inseriti dall'utente.	Dependability	N/A	Tempo di risposta vs Robustezza
<b>8/Alta</b>	DG_2.2 Disponibilità	Il sistema deve avere un up-time di 22 ore al giorno ed un down-time di 2 ore al giorno.	Dependability	RNF_PR_03	Disponibilità vs tolleranza ai guasti
<b>9/Alta</b>	DG_2.3 Disponibilità	I dati del sistema dovranno risiedere sul	Dependability	PR_LE_03	



		server proprietario dell'Ente Nazionale Ricerca Sangue.			
<b>17/Bassa</b>	DG_2.4 Tolleranza ai guasti	Il sistema deve prevedere un sistema di salvataggio automatico dei dati in caso di vari fallimenti del sistema.	Dependability	N/A	Disponibilità vs tolleranza ai guasti
<b>3/Alta</b>	DG_2.5 Sicurezza	Il sistema deve garantire il rispetto delle norme vigenti sulla privacy (legge 1996, 2003) per il trattamento dei dati personali e le operazioni di scambio dei dati.	Dependability	PR_LE_01 PR_LE_02 PR_LE_04	
<b>4/Alta</b>	DG_2.6 Sicurezza	L'accesso ai dati sensibili e le credenziali di accesso devono essere reperibili solo dagli utenti autorizzati, il sistema deve implementare il criptaggio della password.	Dependability	RNF_RE_01 RNF_RE_02 RNF_RE_03	
<b>11/Alta</b>	DG_3.1 Costi di sviluppo	È stimato un costo complessivo di 350 ore per la progettazione e lo sviluppo del sistema (50 ore per ogni team member).	Cost	documenti di management	Costi di sviluppo vs leggibilità
<b>12/Alta</b>	DG_3.2	Il budget allocato al	Cost	documenti di management	



	Costi di distribuzione	training degli utenti è circa 1000 euro.			
<b>13/Alta</b>	DG_3.3 Costi di manutenzione	E' prevista una manutenzione annuale dal costo di 3000 euro.	Cost	documenti di management	
<b>16/Media</b>	DG_4.1 Adattabilità	Il sistema è adattabile e modulabile in modo da poter essere modificato per svolgere altri tipi di servizi.	Maintenance	RNF_SU_02	
<b>7/Alta</b>	DG_4.2 Portabilità	Il sistema è usufruibile attraverso i propri dispositivi preferiti senza necessità di installazione da parte dell'utenza.	Maintenance	PR_PA_01 RNF_SU_01	
<b>10/Alta</b>	DG_4.3 Tracciabilità dei requisiti	La tracciabilità dei requisiti è possibile grazie alla matrice di tracciabilità, attraverso la quale è garantita la connettività tra gli artefatti realizzati.	Maintenance	documenti di management	
<b>15/Media</b>	DG_4.4 Leggibilità	Il codice dovrà essere leggibile in modo da garantire una migliore interazione tra gli sviluppatori.	Maintenance	N/A	Costi di sviluppo vs leggibilità
<b>2/Alta</b>	DG_5.1 Usabilità	Il sistema deve essere facile da utilizzare e comprensibile.	End user	RNF_US_01 RNF_US_02 RNF_US_03	



5/Alta	DG_5.2 Utilità	Utilizzando questo sistema l'utente potrà ridurre in maniera considerevole la procedura per effettuare una donazione.	End user	N/A	
--------	-------------------	---	----------	-----	--





### ***1.2.1 Design Trade off***

#### ***Tempo di risposta vs Robustezza***

Il sistema sarà implementato in modo da preferire la robustezza al tempo di risposta, in modo da garantire un controllo più accurato dei dati di input a discapito del tempo di risposta del sistema.

#### ***Disponibilità vs tolleranza ai guasti***

Il sistema sarà implementato in modo da preferire la disponibilità alla tolleranza ai guasti. In quanto, andando a prevedere un down-time di 2 ore al giorno, riusciamo a gestire meglio il pericolo di guasti in consistenze di dati senza dover sviluppare un sistema di backup.

#### ***Criteri di Maintenance vs Criteri di Performance***

Il sistema sarà implementato in modo da preferire la manutenibilità alla performance. Questo verrà realizzato andando ad aumentare la granularità del sistema (tramite la decomposizione in sottosistemi) andandolo a decentralizzare riuscendo così a mandare in shut-down solo le funzionalità da mantenere e non l'intera piattaforma.

#### ***Costi di sviluppo vs leggibilità***

Il sistema sarà implementato considerando i costi di sviluppo a discapito della leggibilità. La documentazione del codice non sarà approfondita ma verrà gestita attraverso l'utilizzo di Javadoc e quindi con commenti opportuni all'interno del codice. Questa ottimizzazione del lavoro servirà a garantire uno sviluppo più corretto delle varie funzionalità.

## **1.3 Definizione Acronimi ed Abbreviazioni**

### **1.3.1 Definizione Acronimi**

MySQL: Database open source basato sul linguaggio SQL composto da un client a riga di comando ed un server

Javadoc: è uno strumento che viene utilizzato per generare documentazione dal codice Java.

Presentazione: livello architetturale che contiene tutti gli oggetti boundary di interazione con l'utente.

Logica: livello architetturale che include gli oggetti relativi al controllo e all'entità.

Persistenza: livello architetturale che si occupa di ricercare e gestire i dati persistenti.



Design Goals: Obiettivi di design progettati per il sistema proposto.

Trade off: Scelte e compromessi tra design goals dissonanti.

Access Control List: Metodologia di controllo degli accessi ai sottosistemi individuati in fase di design, dove per ogni sottosistema sono specificate coppie (Attore, Operazione) indicanti ognuno un'operazione disponibile ad un determinato attore del sistema.

Varchar, Integer e Date: Tipologie di dati necessarie per la memorizzazione dei dati persistenti nella base di dati.

Deployment Diagram: Diagramma UML di specifica per le relazioni tra le componenti realizzate (con relative tecnologie implementative) e le risorse Hardware e Software necessarie al corretto funzionamento del sistema.

Diagramma dei servizi: Diagramma di specifica dei servizi offerti dai sottosistemi progettati.

Event-driven: Tipologia di controllo del flusso del sistema basata sull'uso di eventi.

### 1.3.2 Abbreviazioni

SDD: System Design Document;

DB: Database;

DBMS: Database Management System;

RAD: Requirement Analysis Document;

UC: Use Case;

RNF: Non functional requirements;

FR: Functional requirements;

PR: Pseudorequisiti;

TCP/IP: Transmission control protocol;

ENRS: Ente Nazionale Ricerca Sangue;



GUI: Graphical User Interface;

## 1.4 Referimenti

Benrd Bruegge- Allen H. Dutoit, Object-Oriented Software Engineering: Using UML  
RAD Requirement Analysis Document Digital Donation V\_1.0

## 1.5 Panoramica

Capitolo 1: Contiene l'introduzione con l'obiettivo del sistema, i design goals ed un elenco di definizioni, acronimi ed abbreviazioni utili alla comprensione dell'intera documentazione.

Capitolo 2: Descrive, le funzionalità e cenni architetture dei sistemi simili dai cui è stata presa ispirazione per definire l'architettura del sistema proposto.

Capitolo 3: Viene presentata l'architettura del sistema proposto, in cui sarà gestita la decomposizione in sottosistemi, il mapping hardware/software, i dati persistenti, il controllo degli accessi e sicurezza, il controllo del flusso globale del sistema, le condizioni limite.

Capitolo 4: Vengono presentati i servizi dei sottosistemi.



## 2. Architettura dei sistemi simili

Attualmente non esiste un prodotto software finalizzato alla gestione delle Donazioni in quanto la procedura è burocratica e cartacea richiedendo un effort maggiore del dovuto da parte degli Operatori di Sede Locale.

Risorsa: <https://www.avis.it/it>

La procedura attuale consiste nei seguenti punti:

- 1) Un Volontario Donatore deve consultare telefonicamente un Operatore per poter avere informazioni riguardanti donazioni e sedute di donazione.
- 2) L'Operatore di Sede Locale dovrà effettuare una stima cartacea, quindi approssimativa, circa i Volontari Donatori
- 3) L'Operatore di Sede Locale, a sua volta, dovrà contattare telefonicamente il Volontario Donatore per l'effettiva Donazione.
- 4) Effettuata una donazione le informazioni aggiornate verranno riportate su un tesserino cartaceo fornito al Donatore all'atto della prima Donazione.

Come team di progettazione, riteniamo che progettare un'architettura da zero, che vada a fornire servizi specifici per l'ente, sia una soluzione migliore rispetto che il riadattare una generica piattaforma di comunicazione, a questo scopo seguono ora esempi di applicativi che in ambiti diversi, ma che forniscono servizi che ci saranno di ispirazione nell'architettura del sistema proposto.

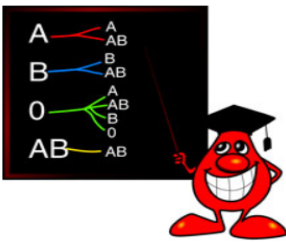
Risorsa: [http://www.avissalerno.it/dove\\_quando.aspx](http://www.avissalerno.it/dove_quando.aspx)

Considerando il sito web dell'AVIS di Salerno abbiamo notato che è presente una struttura tabellare statica che elenca le sedute di donazione in cui la prenotazione può essere effettuata solo telefonicamente. Il nostro obiettivo è quello di dinamizzare la struttura tabellare fornendo una prenotazione che possa essere autonoma da parte del Volontario Donatore.

Inoltre il sito dell'AVIS di Salerno presenta un architettura a tre livelli, corrispondente all'architettura scelta per la realizzazione del progetto Digital Donation.



[Home page](#)  
[AVIS Salerno](#)  
[Donazione](#)  
[Dove e quando donare](#)  
[Legislazione](#)  
[Statistiche](#)  
[News](#)  
[Iscrizione](#)  
[Contatti](#)



Dove e quando donare

Appuntamenti per poter effettuare la donazione				
QUANDO			DOVE	
Giorno	Mese	Anno	Località	Indirizzo
01	GENNAIO	2021	PER INFORMAZIONI LA NOSTRA SEDE E' APERTA AL PUBBLICO NEI GIORNI INDICATI --->	LUNEDI' E MARTEDI' (15.00 - 19.00) - MERCOLEDI' E GIOVEDI' (9.00 - 13.00)
02	GENNAIO	2021	PER DONARE NEI GIORNI STABILITI BISOGNA PRENOTARSI CONTATTANDO I SEGUENTI NUMERI ->	TEL. 089/233600 - CELL. 3381668683 - FAX. 0892754595
03	GENNAIO	2021	LE RACCOLTE C/O LA NOSTRA SEDE VENGONO EFFETTUATE IL VENERDI', SABATO E DOMENICA SU PRENOTAZIONE	VIA PIO XI, 1 - DALLE 8.00 ALLE 10.30 - PRENOTAZIONE OBBLIGATORIA PER PREVENZIONE COVID19
26	NOVEMBRE	2021	SALERNO - SEDE AVIS - VIA PIO XI, 1	DALLE 8.00 ALLE 10.30 PREVIA PRENOTAZIONE TELEFONICA
27	NOVEMBRE	2021	MAIORI - C/O PORTO	DALLE 8.00 ALLE 10.30 PREVIA PRENOTAZIONE TELEFONICA
27	NOVEMBRE	2021	SALERNO - SEDE AVIS - VIA PIO XI, 1	DALLE 8.00 ALLE 10.30 PREVIA PRENOTAZIONE TELEFONICA
28	NOVEMBRE	2021	SALERNO - SEDE AVIS - VIA PIO XI, 1	DALLE 8.00 ALLE 10.30 PREVIA PRENOTAZIONE TELEFONICA
28	NOVEMBRE	2021	PENTA	DALLE 8.00 ALLE 10.30 PREVIA PRENOTAZIONE TELEFONICA
03	DICEMBRE	2021	SALERNO - SEDE AVIS - VIA PIO XI, 1	DALLE 8.00 ALLE 10.30 PREVIA PRENOTAZIONE TELEFONICA

Risorsa: <https://adesionevaccinazioni.soresa.it/adesione/cittadino>


Esistono, inoltre, piattaforme che offrono sistemi e funzionalità simili come il sito web per l'adesione alle vaccinazioni della regione Campania.

Usufruendo di questa piattaforma è possibile effettuare una prenotazione al vaccino che fornisce in output una lista di prenotazioni al centro vaccinale.

Nel momento in cui si effettua una prenotazione al vaccino, viene inviata una mail fornendo le informazioni della seduta di conferma partecipazione per la seduta vaccinale.

Allo stesso modo una volta effettuata una vaccinazione viene erogato un attestato vaccinale; questo corrisponde alla schedulazione della donazione effettuata.

Questi servizi offerti dalla piattaforma delle vaccinazioni della regione Campania potrebbero essere riformulati e riprogettati per il dominio considerato.



Adesione Vaccinazione



Benvenuto nella Piattaforma

Adesione Vaccini - Emergenza COVID-19

① INSERIRE DATI PER ESPRIMERE L'ADESIONE AL PIANO VACCINALE

Codice Fiscale / STP / ENI \*

N. Tessera Sanitaria\* (non necessaria per codice STP o ENI)

Risorsa: [https://easycourse.unisa.it/AgendaStudenti/index.php?view=login&include=login&from=prenotazione&from\\_include=prenotazione\\_home&lang=it](https://easycourse.unisa.it/AgendaStudenti/index.php?view=login&include=login&from=prenotazione&from_include=prenotazione_home&lang=it)

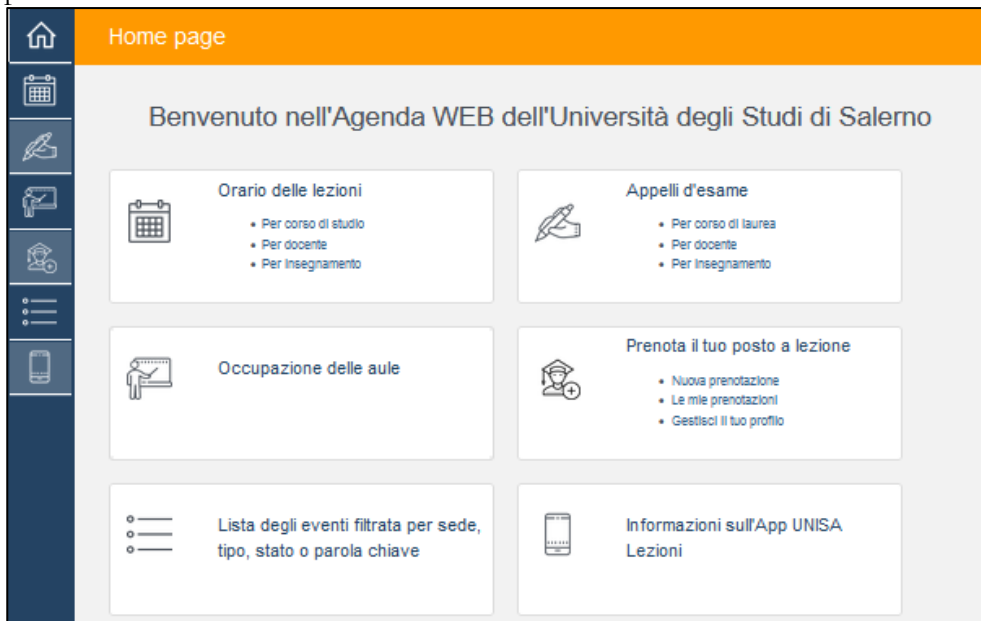


Nel sistema selezionato Unisa Lezioni, è presente la funzionalità per la prenotazione di posti in aula messa a disposizione per gli studenti dell'Università di Salerno, con un numero limitato di partecipanti che possono prenotare il posto in aula per una determinata fascia oraria.

Nel momento in cui uno studente effettua il login alla piattaforma può prenotare il posto in aula e successivamente essere notificato con l'arrivo di una mail autogenerata dal sistema.

Il sistema Unisa Lezioni ha anche la funzionalità di logout per la disconnessione dalla piattaforma.

Queste funzionalità offerte dalla piattaforma Unisa Lezioni potrebbero essere rielaborate e riprogettate per il dominio considerato.



Tali Sistemi sono dotati di un'interfaccia particolarmente semplice ed intuitiva.

In definitiva, dopo aver effettuato un'accurata analisi degli Enti, abbiamo notato che ci sono piattaforme in ambito medico e studentesco che sono simili e riconducibili a quella proposta.



## 3. Architettura del sistema proposto

---

### 3.1 Panoramica

Come Team di progettazione, in accordo con i requisiti non funzionali ed i design goals, propone un applicativo web adeguato e facile da reperire per la tipologia di utente a cui ci si rivolge.

Gli Utenti non registrati alla piattaforma avranno la possibilità di effettuarne la registrazione su richiesta ad un Operatore di Sede Locale.

Gli Utenti potranno effettuare il login per accedere a tutte le funzionalità offerte dalla piattaforma ed il relativo logout per un eventuale disconnessione.

L'obiettivo finale del nostro sistema è quello di sviluppare diverse funzionalità come ad esempio la creazione del tesserino digitale, che permetterà una visualizzazione digitale di tutti i dati appartenenti ad un determinato Donatore, e la visualizzazione di nuove sedute di donazione redatte da un Operatore di Sede Locale per un ottimale organizzazione delle sedute.

Ad ognuna di queste operazioni corrisponderà una notifica; inoltre sarà implementato un servizio di messagistica tramite mail.

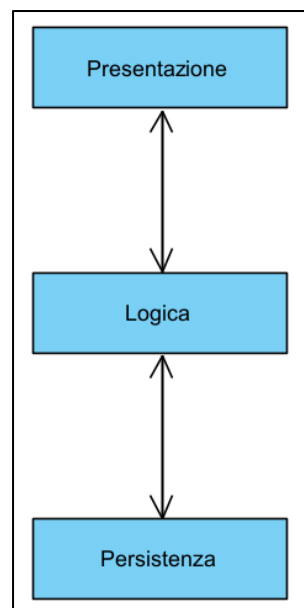
Il pattern architetturale scelto è quello Three-Tier molto diffuso nello sviluppo di interfacce grafiche di sistemi software object-oriented, prevede la suddivisione dell'applicazione in tre livelli differenti che sono interfaccia utente, logica funzionale e logica di persistenza. È un'architettura multi-tier ovvero le varie funzionalità del sito sono logicamente separate e suddivise su più strati o livelli software differenti in comunicazione tra loro.

### 3.2 Decomposizione in sottosistemi

#### 3.2.1 Decomposizione in Layer

La decomposizione prevista per il sistema è composta da tre layer che si occupano di gestire aspetti e funzionalità differenti, in particolare è stato scelto il modello Three-Tier:

- Presentazione: contiene tutti gli oggetti boundary di interazione con l'utente.
- Logica: include gli oggetti relativi al controllo e all'entità.
- Persistenza: si occupa di ricercare e gestire i dati persistenti.



#### 3.2.2 Decomposizione in sottosistemi

Dopo un'attenta analisi funzionale e la presa in considerazione dei Goals & Trade-offs, abbiamo deciso di suddividere le funzionalità per area di gestione.

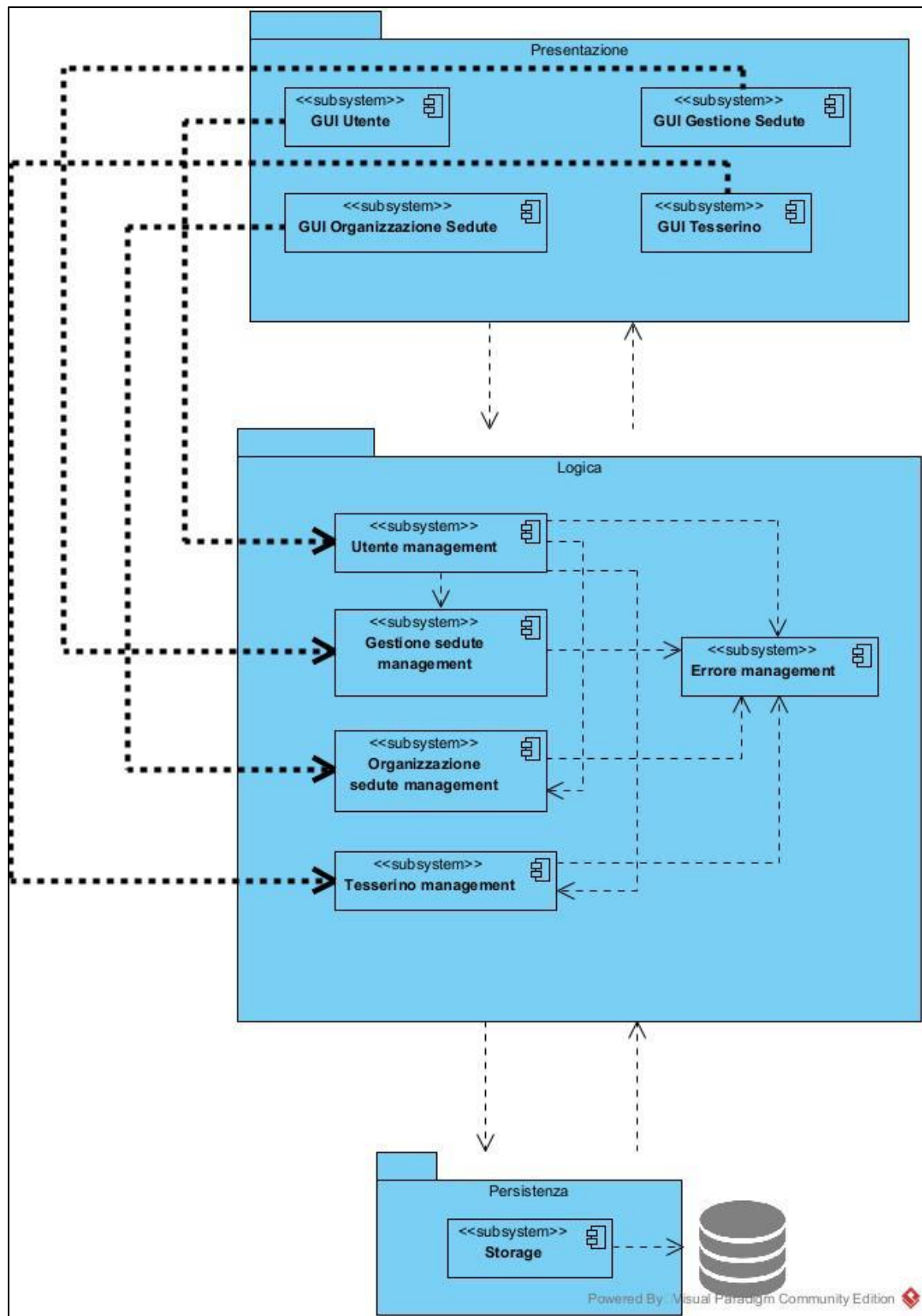
Si è suddiviso il sistema principale in vari sottosistemi che comunicano tra di loro.

In particolare, abbiamo creato l'area funzionale "Utente management" che comprende le Utenze di Guest, Donatore e Operatore.

Si è preso in considerazione un sottosistema che si occupa della gestione degli errori denominato "Errore management", ed infine per la gestione e l'organizzazione delle sedute abbiamo creato rispettivamente due aree funzionali che sono "Gestione sedute management" e "Organizzazione sedute management".

Il sistema si compone di dieci sottosistemi:







Il livello di Presentazione prevede quattro sottosistemi GUI per interfacciarsi con la piattaforma e possiamo identificarli come oggetti “boundary” individuati nel RAD:

- GUI Utente: Sottosistema che fornisce l'interfaccia Utente per interagire con i servizi offerti dalla piattaforma relativi alla gestione dell'utenza.
- GUI Organizzazione Sedute: Sottosistema che fornisce l'interfaccia grafica per i servizi relativi all'organizzazione delle sedute.
- GUI Gestione Sedute: Sottosistema che fornisce l'interfaccia grafica per i servizi relativi alla gestione delle sedute.
- GUI Tesserino : Sottosistema che fornisce l'interfaccia grafica per i servizi relativi al tesserino di un donatore.

Il livello di Logica Applicativa prevede la gestione di cinque sottosistemi:

- Utente Management: Sottosistema che permette all'Utente di usufruire di tutte le funzionalità offerte dal sistema.
- Gestione Sedute Management: Sottosistema che offre funzionalità di gestione delle sedute di donazione in base al tipo di Utente che ha effettuato accesso alla piattaforma.
- Organizzazione Sedute Management: Sottosistema che offre funzionalità di organizzazione delle sedute di donazione in base al tipo di Utente che ha effettuato accesso alla piattaforma.
- Errore Management: Sottosistema che si occupa della gestione degli errori.
- Tesserino Management: Sottosistema che permette la gestione del tesserino digitale.

Il livello di Persistenza prevede la gestione di un sottosistema:

- Storage: Sottosistema che si occupa di immagazzinare e prelevare i dati persistenti dal nostro database, in particolare lo stesso sarà diviso in due macro-componenti logiche:
  - Macro-componente DAOs: sottocomponente del sottosistema Storage che si occuperà di offrire i servizi di manipolazione dei dati persistenti.
  - Macro-componente Entity: sottocomponente del sottosistema Storage che racchiuderà tutte le entità rappresentanti informazioni da immagazzinare nel Database e direttamente manipolabili dal livello di Logica Applicativa.

## Deployment Diagram

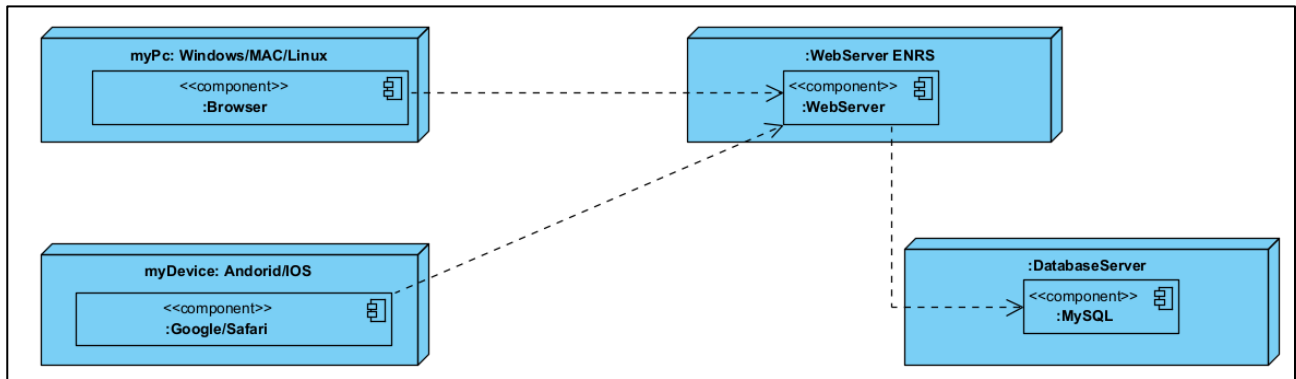
Come team di progetto abbiamo deciso di formalizzare il deployment diagram con componenti di alto livello per focalizzarci sulle dipendenze tra i vari componenti descritti nel Mapping Hardware e Software.

L'utente grazie all'ausilio del suo browser preferito o il proprio device si connette al web server (tramite link ipertestuale).

All'interno del webServer ENRS vi saranno le componenti inerenti al livello di Presentazione e al livello di Logica Applicativa per soddisfare tutte le richieste dell'utente e fornire le risposte che cerca.



DatabaseServer conterrà i dati persistenti che fornirà al webServer nel momento in cui questi vengono richiesti. Appena le informazioni arriveranno al WebServer verranno rigirati al client.

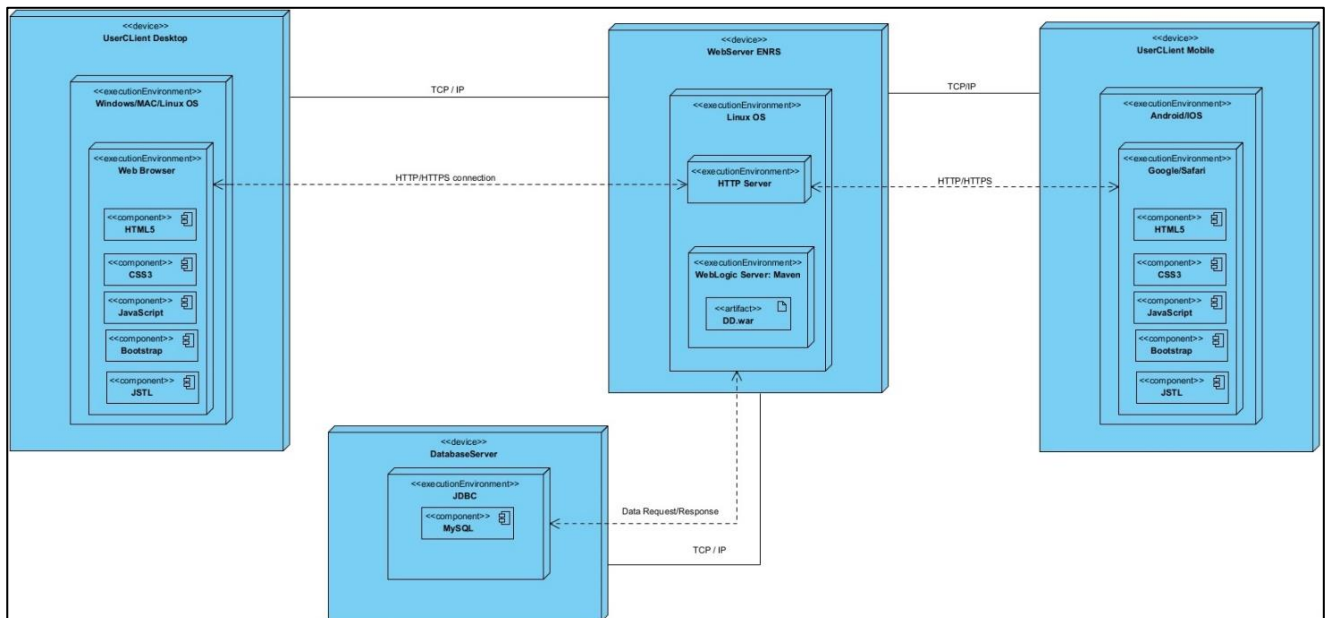


### 3.3 Mapping HW/SW

I client potranno interfacciarsi attraverso i propri browser predefiniti oppure con dispositivi mobili che girano su sistemi operativi Android/IOS. Questi dispositivi utilizzati dagli utenti dovranno avere a disposizione una connessione ad internet per permettere la comunicazione con il WebServer.

Tra l'utente e il WebServer vi sarà una comunicazione basata sul protocollo http in cui il client invierà le richieste e il server provvederà alle risposte. Queste richieste e risposte saranno gestite tramite protocollo sicuro TCP/IP che prevede il three-way handshake quindi un canale di comunicazione sicuro che garantisce che tutti i pacchetti arrivino a destinazione. Inoltre, sul WebServer sarà caricato il sito sotto forma di archivio WAR.

Il WebServer ENRS fa da interfaccia tra l'utente e il DatabaseServer. Questo database verrà interrogato tramite JDBC che è un connettore che consente l'accesso e la gestione della persistenza dei dati sulle basi di dati. Il Database verrà gestito tramite MySQL ovvero un gestore di database relazionali





### 3.4 Gestione Dati persistenti

Il sistema gestisce la memorizzazione di dati persistenti tramite l'utilizzo di un database relazionale per consentire un accesso veloce e sicuro ai dati ed un ampio spazio di archiviazione.

#### *Utente*

Nome	Tipo	Null	Key
codice_fiscale_utente	VARCHAR(255)	Not null	Primary key
nome	VARCHAR(255)	Not null	
cognome	VARCHAR(255)	Not null	
email	VARCHAR(255)	Not null	
password	VARCHAR(255)	Not null	

#### *Donatore*

Nome	Tipo	Null	Key
codice_fiscale_utente	VARCHAR(255)	Not null	Primary key
residenza	VARCHAR(255)	Not null	
luogo_di_nascita	VARCHAR(255)	Not null	
data_di_nascita	DATETIME(6)	Not null	

#### *Tesserino*

Nome	Tipo	Null	Key
id_tessera	BIGINT	Not null	
numero_matricola	INT	Not null	
data_rilascio	DATE	Not null	
gruppo_sanguigno	VARCHAR(255)	Not null	



rh	VARCHAR(20)	Not null	
img_source	VARCHAR(255)	Not null	
codice_fiscale_donatore	VARCHAR(255)	Not null	Primary key

### ***Donazione***

Nome	Tipo	Null	Key
data_donazione	DATETIME(6)	Not null	
Id_donazione	BIGINT	Not null	Primary key
tipo_donazione	VARCHAR(255)	Not null	
cf_tessera	VARCHAR(50)	Not null	Foreign key

### ***Seduta***

Nome	Tipo	Null	Key
Id_seduta	BIGINT	Not null	Primary key
data_seduta	DATETIME(6)	Not null	
luogo	VARCHAR(255)	Not null	
ora_inizio	TIME	Not null	
ora_fine	TIME	Not null	
numero_partecipanti	INT	Not null	
data_inizio_prenotazione	DATETIME(6)	Not null	
data_fine_prenotazione	DATETIME(6)	Not null	
id_sede_locale	BIGINT	Not null	Foreign key



### ***Seduta\_Donatore***

Nome	Tipo	Null	Key
id_seduta	BIGINT	Not null	Foreign key
codice_fiscale_donatore	VARCHAR(255)	Not null	Foreign key

### ***Seduta\_Guest***

Nome	Tipo	Null	Key
id_seduta	BIGINT	Not null	Foreign key
codice_fiscale_guest	VARCHAR(255)	Not null	Foreign key

### ***Guest***

Nome	Tipo	Null	Key
nome	VARCHAR(255)	Not null	
cognome	VARCHAR(255)	Not null	
telefono	VARCHAR(255)	Not null	
codice_fiscale_guest	VARCHAR(255)	Not null	Primary key
patologie	VARCHAR(255)	Not null	
gruppo_sanguigno	VARCHAR(255)	Not null	

### ***Operatore***

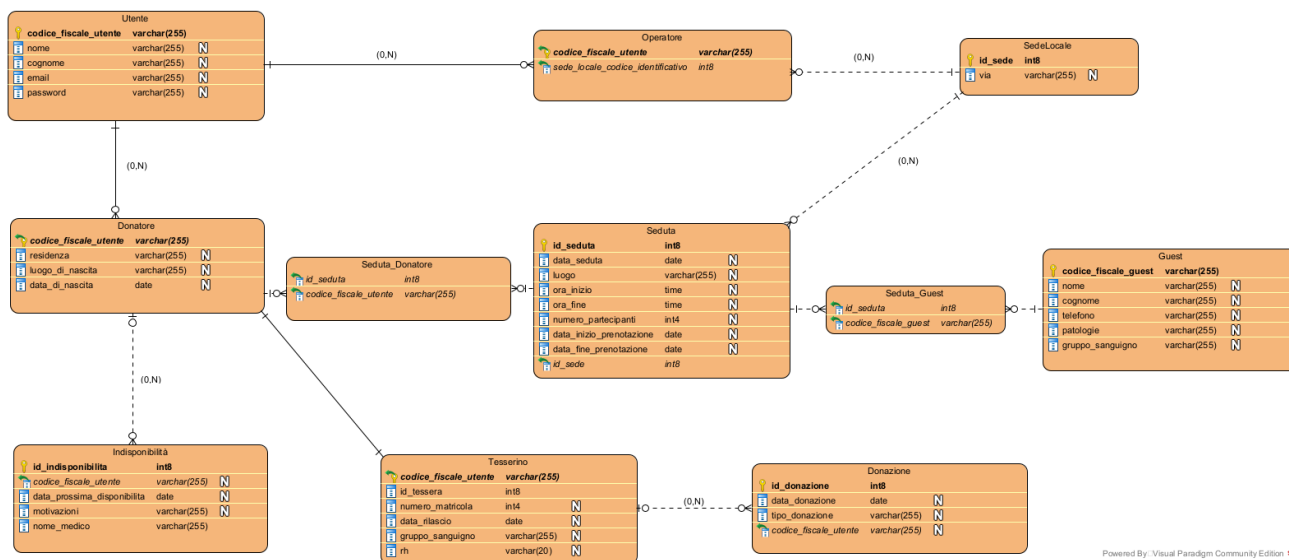
Nome	Tipo	Null	Key
codice_fiscale_utente	VARCHAR(255)	Not null	Primary key
sede_locale_codice_identificativo	BIGINT	Not null	Foreign key

## Indisponibilità

Nome	Tipo	Null	Key
data_prossima_disponibilità	DATETIME(6)	Not null	
id_indisponibilità	BIGINT	Not null	Primary key
motivazioni	VARCHAR(255)	Not null	
nome_medico	VARCHAR(255)		
codice_fiscale_donatore	VARCHAR(255)	Not null	Foreign key

## Sede locale

Nome	Tipo	Null	Key
Id_sede	BIGINT	Not null	Primary key
via	VARCHAR(255)	Not null	



Powered By: Visual Paradigm Community Edition





### 3.5 Controllo degli accessi e sicurezza

Digital Donation è un sistema multi-utente, vi sono differenti attori ognuno avente l'autorizzazione di compiere differenti operazioni.

Il controllo degli accessi è garantito tramite l'utilizzo di username e password. La sicurezza sui dati personali dei donatori è garantita dall'accesso controllato in quanto solo l'operatore di sede locale può avere accesso ai dati relativi ai donatori.

Per schematizzare al meglio il controllo degli accessi abbiamo suddiviso per area di gestione le azioni consentite, al fine di ottenere una visione più compatta e dettagliata grazie a delle matrici degli accessi (più precisamente delle Access Control list) riportate di seguito:

#### ***Gestione Utente***

Attore	Operazione
Utente Registrato	Login()
Utente Registrato	Logout()
Utente Registrato	CambioPassword()

#### ***Organizzazione delle Sedute di Donazione***

Attore	Operazione
Operatore	SchedulazioneSedute()
Operatore	ModificaSeduta()
Operatore	EliminaSeduta()
Operatore	MonitoraggioSeduta()
Operatore	InserimentoGuest()
Donatore	FeedbackPositivo()



Donatore	FeedbackNegativo()
Donatore	VisualizzaSeduta()

### *Gestione del tesserino digitale*

Attore	Operazione
Operatore	CreazioneTesserino()
Donatore	AutodichiarazioneIndisponibilità()
Donatore	VisualizzazioneTesserino()

### *Gestione delle Sedute di Donazione*

Attore	Operazione
Operatore	SalvataggioDonazione()
Operatore	SalvataggioIndisponibilità()
Operatore	VisualizzaElencoSedute()

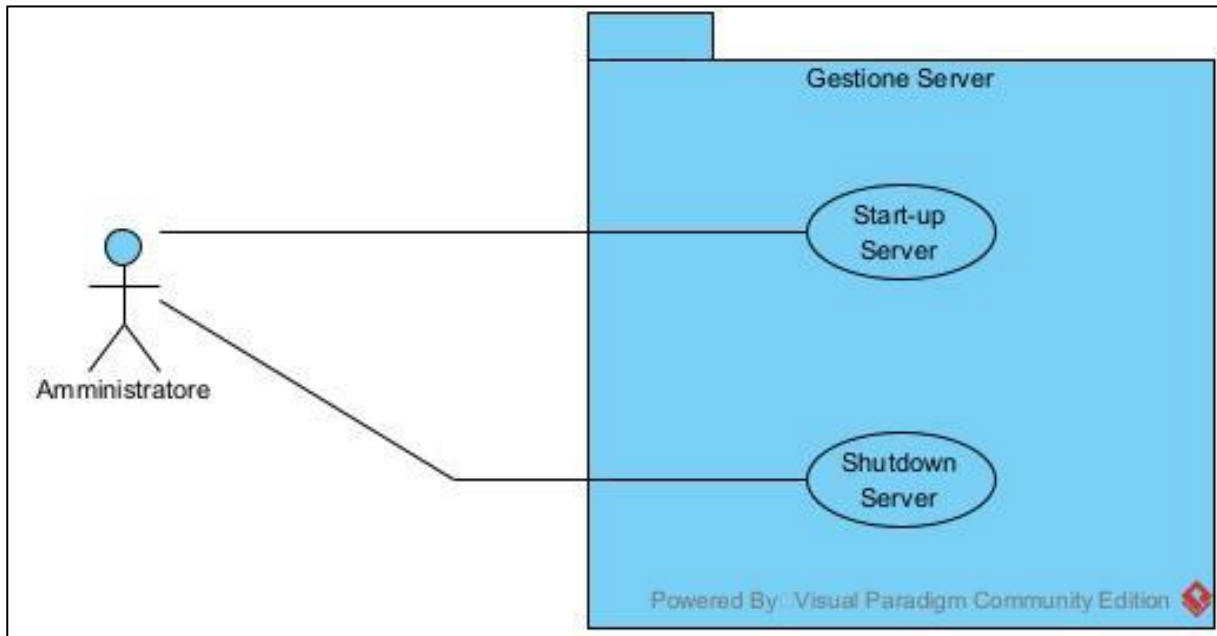


### **3.6 Controllo flusso globale del sistema**

Il flusso del sistema fornisce diverse funzionalità che richiedono una continua interazione da parte dell'utente, ragione per cui, il controllo del flusso globale è di tipo event-driven ovvero guidato dagli eventi. Quando si verifica un evento, vengono selezionati i sottosistemi che si occupano della Logica Applicativa e della gestione dei dati persistenti.

### 3.7 Definizione Use case per condizioni limite

#### *Use Case Diagram Gestione Server*



#### *3.7.1 Startup del server*

<b>Identificativo</b> UC_GS_01	Startup del server	Data	27/11/2021
		Versione	0.00.001
		Autore	ADM, KP, FS
<b>Descrizione</b>	Il Caso d'uso fornisce la funzionalità di avviare il sistema.		
<b>Attore Principale</b>	<b>Amministratore</b> L'attore è interessato ad avviare il sistema.		
<b>Attori secondari</b>	N/A.		
<b>Entry Condition</b>	L'amministratore ha accesso al server.		
<b>Exit condition</b> On success	Il sistema è avviato correttamente.		
<b>Exit condition</b> On failure	Il sistema non è avviato correttamente.		



<b>Rilevanza/User Priority</b>		Priorità bassa
<b>Frequenza stimata</b>		5 volte/anno
<b>Extension point</b>		N/A
<b>Generalization of</b>		N/A
<b>FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO</b>		
1	<b>Amministratore:</b>	Stabilisce una connessione con il database.
2	<b>Amministratore:</b>	Avvia uno script di configurazione del database.
3	<b>Sistema:</b>	Inizializza il database.
4	<b>Amministratore:</b>	Esegue il deploy dell'archivio "dd.war" del sistema sul server.
5	<b>Sistema:</b>	Verifica che il server è stato avviato correttamente.
6	<b>Sistema</b>	Legge i dati dal database.
<b>Scenario/Flusso di eventi Alternativo: lo script inserito non è corretto</b>		
3.1	<b>Sistema:</b>	Notifica l'errore all'amministratore.
<b>Scenario/Flusso di eventi Alternativo: Il server non si è avviato correttamente.</b>		
5.1	<b>Sistema:</b>	Invia un messaggio d'errore all'amministratore.
<b>Scenario/flusso di errore: N/A</b>		
<b>Note: N/A</b>		
<b>Special Requirements: N/A</b>		



### 3.7.2 Shutdown del server

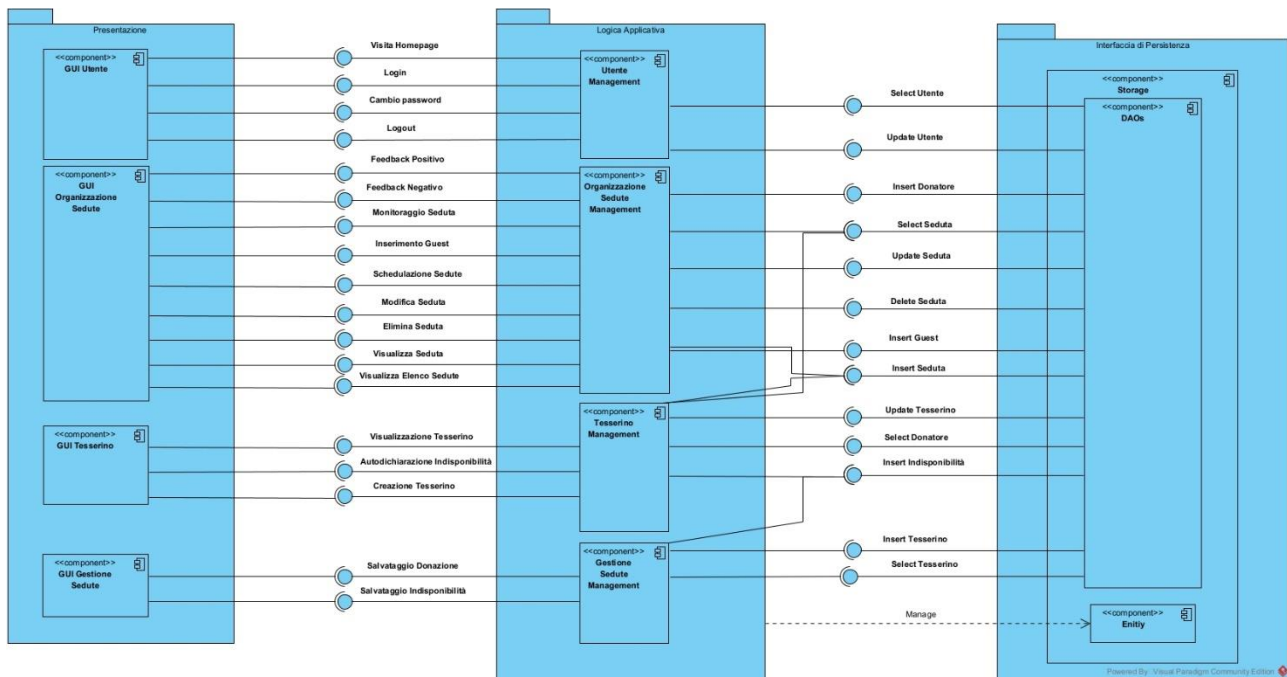
Identificativo  UC_GS_02	Shutdown del server	Data	27/11/2021
		Versione	0.00.001
		Autore	ADM, KP, FS
Descrizione	Il Caso d'uso fornisce la funzionalità di terminazione del server da parte dell'amministratore.		
Attore Principale	Amministratore  L'amministratore è interessato a terminare il server.		
Attori secondari	N/A.		
Entry Condition	Il server è avviato e l'amministratore ha effettuato l'accesso.		
Exit condition  On success	Il sistema è terminato correttamente.		
Exit condition  On failure	Il sistema non termina.		
Rilevanza/User Priority	Priorità bassa		
Frequenza stimata	5 volte/anno		
Extension point	N/A		
Generalization of	N/A		
FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO			
1	Amministratore:	Avvia uno script per terminare la connessione al database.	
2	Amministratore:	Esegue la terminazione del server.	
Scenario/Flusso di eventi Alternativo: lo script inserito non è corretto.			
1.1	Sistema:	Notifica l'errore all'amministratore.	
Scenario/Flusso di errore: il sistema non termina			
2.1	Sistema:	Il sistema comunica l'errore all'amministratore.	
Note: N/A			
Special Requirements:N/A			



### **3.7.3 Fallimento**

1. Il team di manutenzione del sistema, periodicamente, si occuperà di aggiornare manualmente uno script in cui vengono memorizzati i dati del Database, per evitare grosse perdite di dati in caso di problemi o guasti del Database.
2. Nel caso in cui si verifichi un'interruzione inaspettata dell'alimentazione non sono previsti metodi che ripristino lo stato del Sistema precedente allo spegnimento non voluto.
3. Un altro caso di fallimento potrebbe derivare dal software stesso che causa una chiusura inaspettata dovuta ad errori commessi durante la fase di implementazione. Non essendo previste politiche correttive, l'unica operazione consentita in questa particolare situazione è la chiusura del sistema e il suo successivo riavvio.
4. Un altro caso di fallimento potrebbe essere dovuto ad un errore critico nell'hardware, contro il quale non è prevista alcuna contromisura.

## 4.Servizi dei sottosistemi



### Presentazione

**GUI Utente** utilizza quattro servizi offerti dalla Logica Applicativa del Sistema:

- Visita Homepage
- Login
- Logout
- Cambio Password

**GUI Organizzazione Sedute Donazione** utilizza nove servizi offerti dalla Logica Applicativa del Sistema:

- Schedulazione Sedute
- Modifica Seduta
- Elimina Seduta
- Monitoraggio Seduta
- Inserimento Guest
- Visualizza Elenco Sedute
- Visualizza Seduta
- Feedback Positivo
- Feedback Negativo





**GUI Gestione Tesserino** utilizza tre servizi offerti dalla Logica Applicativa del Sistema:

- Creazione Tesserino
- Autodichiarazione Indisponibilità
- Visualizzazione Tesserino

**GUI Gestione Sedute Donazione** utilizza due servizi offerti dalla Logica Applicativa del Sistema:

- Salvataggio Donazione
- Salvataggio Indisponibilità

## Logica Applicativa

**Utente Management** utilizza due servizi offerti dalla Logica di Persistenza:

- Select Utente
- Update Utente

**Organizzazione Sedute Management** utilizza sei servizi offerti dalla Logica di Persistenza:

- Insert Donatore
- Insert Seduta
- Select Seduta
- Update Seduta
- Delete Seduta
- Insert Guest

**Tesserino Management** utilizza cinque servizi offerti dalla Logica di Persistenza:

- Insert Seduta
- Select Seduta
- Update Tesserino
- Select Donatore
- Insert Indisponibilità

**Gestione Sedute Management** utilizza tre servizi offerti dalla Logica di Persistenza:

- Insert Tesserino
- Select Tesserino
- Insert Indisponibilità.



## 5.Glossario

---

### A

#### **Architettura**

L'architettura software è l'organizzazione fondamentale di un sistema, definita dai suoi componenti, dalle relazioni reciproche tra i componenti e con l'ambiente, e i principi che ne governano la progettazione e l'evoluzione.

### C

#### **Condizioni limite**

Le condizioni limite sono quelle che hanno a che fare con la configurazione del sistema e più in generale con quello che riguarda l'amministrazione quando c'è per esempio un crash o un riavvio.

### H

#### **Hardware**

Insieme delle component fisiche, non modificabili, di un sistema di elaborazione dati.

### L

#### **Layer**

Layer deriva dal concetto di layering ovvero consentire ad un sistema di essere organizzato come una gerarchia di sottosistemi in cui ognuno fornisce servizi al sistema di livello superiore utilizzando i servizi forniti dal sistema al livello inferiore.



# M

## **Multi-tier**

È un'architettura dove le varie funzionalità del sito sono logicamente separate e suddivise su più strati o livelli software differenti in comunicazione tra loro.

# O

## **Object Oriented (OO)**

È un paradigma di programmazione che permette di definire, modellare ed istanziare oggetti.

# S

## **Startup**

Per startup si intende la fase di avvio del sistema.

## **Shutdown**

La procedura di spegnimento del computer alla fine della sessione di lavoro, che comporta l'interruzione di tutti i programmi attivi e la chiusura del sistema operativo.

## **Software**

È l'insieme delle procedure e definizioni in un sistema di elaborazione dati.

## **Sottosistema**

Un sottosistema, nella sua accezione più generica, è un sistema subalterno e secondario rispetto a un sistema più complesso, dal quale dipende e senza il quale non può funzionare autonomamente.



# T

## **Throughput**

Quantità di dati trasmessi in un'unità di tempo.

## **TCP/IP**

Transmission control protocol è un protocollo di comunicazione che offre una connessione affidabile per quanto riguarda la trasmissione e ricezione di dati.

## **Three-way handshake**

è un metodo utilizzato in una rete TCP/IP per creare una connessione tra un host/client locale e un server. Si tratta di un metodo in tre fasi che richiede sia al client che al server di scambiare i pacchetti prima dell'inizio della comunicazione dei dati.

# W

## **Web-server**

E' un'applicazione software che, in esecuzione su un server, è in grado di gestire le richieste di trasferimento di pagine web.