

# **Progettazione e Sviluppo di Sistemi Software**

## **Processo di sviluppo**

Andrea D'Arco M63001063

Pipicelli Giuseppe 000117607

## Processo di sviluppo

Per lo sviluppo del sistema “CampaniaCovidFree” abbiamo adottato un processo di sviluppo iterativo ed incrementale, con incrementi pianificati volta per volta di durata prefissata, seguendo il processo UP (Unified Process).

Tuttavia, abbiamo anche sfruttato tecniche del processo di sviluppo su integrazione e configurazione, riguardanti il riuso di componenti software usati come black box che sono stati solo configurati ed integrati tramite interfacce, come le interfacce JPA (Java Persistence API) per la gestione della persistenza.

Abbiamo realizzato un’applicazione web, con una forte possibilità di avere cambiamenti futuri, utilizzando metodologie agili, ovvero produrre solo la documentazione necessaria in modo da avere un veloce feedback sui documenti prodotti.

Nella prima iterazione, abbiamo implementato il caso d’uso basilare, che include tutti gli altri, di riconoscimento dell’utente. Tuttavia, nel nostro progetto, il requisito funzionale prioritario è InserisciRisultatoTampone in quanto permette di attraversare completamente l’architettura e ne delinea la struttura completa del sistema; lo abbiamo sviluppato nella seconda iterazione, la cui fase di Costruzione ci ha impiegato quasi 2 settimane, invece che 1 come per tutte le altre fasi. Esso è anche il caso d’uso a più alto rischio poiché il più complesso.

Di seguito, sono mostrate le durate delle varie fasi:

Fase	N° Iterazioni	Inizio	Fine
Ideazione	1	7/09/2020	13/09/2020
Elaborazione 1°	1	14/09/2020	20/09/2020
Costruzione 1°	1	21/09/2020	27/09/2020
Elaborazione 2°	1	30/09/2020	6/10/2020
Costruzione 2°	2	7/10/2020	18/10/2020
Transizione	1	19/10/2020	25/10/2020

Prima della fase di ideazione, c’è stato un Workshop sui requisiti, della durata di 5 giorni (solo di mattina) per stabilire cosa il nostro sistema doveva fare e quali erano gli obiettivi. È durato un po’ di più del previsto, in quanto l’idea iniziale era rivolta ad un altro sistema software, abbandonato al terzo giorno di workshop.

Inoltre, dopo la fase di Costruzione 1°, c’è stato un secondo Workshop sui requisiti, della durata di 2 giorni (28-29/09/2020), dove sono state modificate alcune funzionalità e aggiunte altre che inizialmente non avevamo considerato.

In quest'altra tabella sono mostrati i documenti prodotti in ogni fase:

Fase	Descrizione	Milestone
Ideazione	Si svilupperanno i requisiti informali del prodotto per il progetto di PSSS.	Use Case Diagram, Glossario, SRS
Elaborazione 1°	Sviluppo del primo prototipo architetturale attraverso il design architetturale MVC e lo stile client-server. Analisi e design dei requisiti selezionati per la prima iterazione. Nominazione progetto: <b>'Gestione Emergenza in Campania'</b>	Prima versione del prodotto
Elaborazione 2°	Sviluppo di un secondo prototipo architetturale attraverso il design architetturale MVC, con l'utilizzo del framework Spring. Analisi e design dei requisiti selezionati per la seconda iterazione. Ridenominazione del progetto: <b>"CampaniaCovidFree"</b>	Seconda versione del prodotto
Costruzione 1°	Sviluppo e implementazione del primo prototipo e Analisi e design degli use case rimanenti.	Primo rilascio del software
Costruzione 2°	Sviluppo e implementazione del secondo prototipo e Analisi e design degli use case rimanenti.	Secondo rilascio del software
Transizione	Preparazione per la distribuzione del prodotto. Include il supporto per l'installazione.	Rilascio sul mercato Del software

## Tools utilizzati

Durante la realizzazione del progetto CampaniaCovidFree il team ha utilizzato diversi strumenti sia per l'organizzazione, che per il lavoro di progettazione:

Organizzazione:

- Microsoft Teams per le riunioni;
- Google Drive per la condivisione del materiale relativo al progetto, dai documenti al codice

Analisi e progettazione:

- Visual Paradigm 16.2

Implementazione codice:

- Eclipse IDE 2020;

## **Prodotto distribuito**

La versione del prodotto rilasciata conterrà le seguenti funzionalità:

- Login, Registrazione e Visualizzazione del resoconto giornaliero per tutti gli utenti;
- Inserimento Sintomi, Aggiunta di un familiare e gestione di tale richiesta, Visualizzazione dell'esito di un tampone, Visualizzazione dello stato generale dei familiari per il Paziente;
- Monitoraggio della situazione generale dei propri pazienti e possibilità di prenotargli un tampone per i Medici di Base;
- Inserimento del risultato di un tampone per il Dipendente dell'ASL.

# **Progettazione e Sviluppo di Sistemi Software**

## **Documento specifica dei requisiti**

Andrea D'Arco M63001063

Pipicelli Giuseppe 000117607

# Indice

1. Introduzione
  - 1.1. Propositi
  - 1.2. Obiettivi
  - 1.3. Panoramica
2. Descrizione generale
  - 2.1. Prospettive del prodotto
  - 2.2. Funzionalità del prodotto
  - 2.3. Caratteristiche utente
  - 2.4. Vincoli generali del software
3. Requisiti specifici
  - 3.1. Requisiti di interfaccia esterna
    - 3.1.1. Interfaccia utente
    - 3.1.2. Interfaccia hardware
    - 3.1.3. Interfaccia software
    - 3.1.4. Interfaccia di comunicazione
  - 3.2. Requisiti funzionali
    - 3.2.1. Registrati
      - 3.2.1.1. Introduzione
      - 3.2.1.2. Input
      - 3.2.1.3. Elaborazione
      - 3.2.1.4. Output
    - 3.2.2. Login
      - 3.2.2.1. Introduzione
      - 3.2.2.2. Input
      - 3.2.2.3. Elaborazione
      - 3.2.2.4. Output
    - 3.2.3. Monitora situazione generale del paziente
      - 3.2.3.1. Introduzione
      - 3.2.3.2. Input
      - 3.2.3.3. Elaborazione
      - 3.2.3.4. Output
    - 3.2.4. Prenota tampone
      - 3.2.4.1. Introduzione
      - 3.2.4.2. Input
      - 3.2.4.3. Elaborazione
      - 3.2.4.4. Output
    - 3.2.5. Inserisci Sintomi
      - 3.2.5.1. Introduzione
      - 3.2.5.2. Input
      - 3.2.5.3. Elaborazione
      - 3.2.5.4. Output
    - 3.2.6. Inserisci risultato tampone

- 3.2.6.1. Introduzione
  - 3.2.6.2. Input
  - 3.2.6.3. Elaborazione
  - 3.2.6.4. Output
- 3.2.7. Richiesta aggiungi familiare
  - 3.2.7.1. Introduzione
  - 3.2.7.2. Input
  - 3.2.7.3. Elaborazione
  - 3.2.7.4. Output
- 3.2.8. Gestisci richiesta di aggiunta familiare
  - 3.2.8.1. Introduzione
  - 3.2.8.2. Input
  - 3.2.8.3. Elaborazione
  - 3.2.8.4. Output
- 3.2.9. Visualizza sintomi
  - 3.2.9.1. Introduzione
  - 3.2.9.2. Input
  - 3.2.9.3. Elaborazione
  - 3.2.9.4. Output
- 3.2.10. Visualizza stato generale dei familiari
  - 3.2.10.1. Introduzione
  - 3.2.10.2. Input
  - 3.2.10.3. Elaborazione
  - 3.2.10.4. Output
- 3.2.11. Visualizza esito tampone
  - 3.2.11.1. Introduzione
  - 3.2.11.2. Input
  - 3.2.11.3. Elaborazione
  - 3.2.11.4. Output
- 3.2.12. Calcola resoconto giornaliero
  - 3.2.12.1. Introduzione
  - 3.2.12.2. Input
  - 3.2.12.3. Elaborazione
  - 3.2.12.4. Output
- 3.2.13. Visualizza resoconto giornaliero
  - 3.2.13.1. Introduzione
  - 3.2.13.2. Input
  - 3.2.13.3. Elaborazione
  - 3.2.13.4. Output
- 3.3. Requisiti di prestazione
- 3.4. Vincoli di progetto
- 3.5. Attributi del sistema software
- 3.6. Altri requisiti
  - 3.6.1. Database
- 4. Glossario dei dati

## **1. Introduzione**

### **1.1. Propositi**

Il proposito di questo documento è quello di specificare i requisiti del sistema software “CampaniaCovidFree” per facilitarne la realizzazione e la validazione.

Questo documento è redatto seguendo le indicazioni contenute nel documento “IEEE Recommended Practice for Software Requirements Specifications” avente riferimento IEEE Std 830-1993 (Revision of IEEE Std 830-1984).

### **1.2. Obiettivi**

Il sistema “CampaniaCovidFree” realizzato si propone di monitorare la situazione di emergenza dovuta alla pandemia per il Covid-19, gestendo le seguenti 7 ASL, caratterizzate da nome e indirizzo, presenti nella regione Campania:

- ASL Avellino;
- ASL Benevento;
- ASL Caserta;
- ASL Napoli 1 Centro;
- ASL Napoli 2 Nord;
- ASL Napoli 3 Sud;
- ASL Salerno.

Un utente può accedere o registrarsi al sistema come paziente, medico di base o dipendente dell'ASL, specificando nome, cognome, password, e-mail e ASL di competenza.

Il sistema permette ai pazienti, caratterizzati ulteriormente da uno stato (attualmente positivo/negativo al virus) e un indirizzo di domicilio, di inserire all'interno della piattaforma i propri sintomi.

Inizialmente, il paziente viene registrato nel sistema come un nucleo familiare a sé stante, però ha la possibilità di richiedere l'aggiunta di un altro paziente al proprio nucleo familiare, e, se questo ne accetta la richiesta, sarà aggiunto. Ogni membro del nucleo familiare può visualizzare lo stato generale dei suoi familiari, visualizzandone sia i sintomi che l'eventuale positività

Il medico di base, che esercita in una determinata sede, monitora i sintomi e la positività dei membri del nucleo familiare del paziente, e decide se è opportuno prenotare un tampone, per verificare o meno la presenza del virus, presso il domicilio del paziente, facendo iniziare a quest'ultimo un periodo di quarantena precauzionale. Sarà poi compito di un dipendente dell'ASL, di inserire il risultato del tampone nel sistema. Se il tampone è positivo, sarà prenotato per il paziente un altro tampone a distanza di 7 giorni e, se non lo è già, sarà modificato lo stato del paziente in positivo; se è negativo e lo stato del paziente è negativo (è il primo tampone), la quarantena terminerà, mentre se è negativo il tampone e lo stato del paziente è positivo, quest'ultimo cambierà in transizione, ovvero, per essere definitivamente negativo, dovrà risultare negativo a un ulteriore tampone, prenotato per 3 giorni dopo.

A fine giornata, il sistema in automatico, calcolerà il resoconto giornaliero dei tamponi il cui esito è stato comunicato in quel determinato giorno (numero di tamponi effettuati e numero di positivi), prelevando i dati da ciascuna ASL del territorio campano. Il risultato potrà, poi, essere visualizzato da ciascun utente.

### **1.3. Panoramica**

La restante parte di questo documento contiene una descrizione dettagliata e approfondita



delle funzionalità richieste al sistema software “Gestione emergenza Covid-19 in Campania” secondo gli obiettivi espressi al punto 1.2.

## **2. Descrizione generale**

### **2.1. Prospettive del prodotto**

Il sistema software “CampaniaCovidFree” non è parte e non deve integrarsi in alcun sistema software, essendo un sistema completamente autonomo.

### **2.2. Funzionalità del prodotto**

Il sistema software “CampaniaCovidFree” deve permettere di:

- registrarsi al sistema;
- loggarsi al sistema;
- monitorare la situazione generale del paziente;
- prenotare un tampone;
- inserire sintomi;
- inserire risultato di un tampone;
- aggiungere un paziente al proprio nucleo familiare;
- gestire la richiesta di aggiunta familiare;
- visualizzare i sintomi;
- visualizzare stato generale del nucleo familiare;
- visualizzare esito tampone;
- calcolare il resoconto giornaliero;
- visualizzare il resoconto giornaliero;

Le informazioni trattate sono presenti su un server ed accessibili tramite web application.

### **2.3. Caratteristiche utente**

Il sistema software “CampaniaCovidFree” è rivolto ad una utenza senza particolari conoscenze informatiche. All’utente è richiesta solamente una conoscenza informatica di base (alfabetizzazione informatica).

### **2.4. Vincoli generali del software**

I vincoli per il tampone è che non può essere effettuato se prima non è stato prenotato.

Al paziente è concesso visualizzare l’esito dei tamponi e i sintomi dei soli componenti del suo nucleo familiare (compreso se stesso).

Il medico di base può visualizzare l’esito dei tamponi e i sintomi dei propri pazienti. Un dipendente dell’ASL può inserire il risultato di un tampone solo per i pazienti associati a tale ASL.

## **3. Requisiti specifici**

### **3.1. Requisiti di interfaccia esterna**

#### **3.1.1. Interfacce**

Il sistema software “CampaniaCovidFree” deve essere dotato di un’interfaccia amichevole, con finestre e pulsanti per l’utente non registrato, il paziente, il medico di base e il dipendente dell’ASL.

### **3.2. Requisiti funzionali**

#### **3.2.1. Registrati**

##### **3.2.1.1. Introduzione**

Permette all’utente di registrarsi al sistema.

#### 3.2.1.2. **Input**

Nome, Cognome, Email e Password dell'Utente e in più se è un Paziente, anche stato (di default negativo), Indirizzo Civico e se è in Quarantena (di default no), mentre se è un Medico di Base, agli inputi dell'utente va aggiunto l'Indirizzo della Sede dove esercita il lavoro.

#### 3.2.1.3. **Elaborazione**

L'Utente inserisce nel sistema le sue credenziali e il sistema, se le credenziali sono valide, lo registra.

#### 3.2.1.4. **Output**

L'Utente viene registrato nel sistema come Paziente, Medico di Base o dipendente dell'ASL.

### 3.2.2. **Login**

#### 3.2.2.1. **Introduzione**

Permette a un utente di accedere al sistema come Paziente, Medico di Base o dipendente dell'ASL.

#### 3.2.2.2. **Input**

Email e Password.

#### 3.2.2.3. **Elaborazione**

L'Utente inserisce le sue credenziali, che, se sono presenti nel sistema, ne consentono l'accesso.

#### 3.2.2.4. **Output**

L'Utente accede al sistema come Paziente, Medico di Base o Dipendente dell'ASL.

### 3.2.3. **Monitora situazione generale del paziente**

#### 3.2.3.1. **Introduzione**

Permette al medico di base di visualizzare i sintomi di un paziente e lo stato (positivo o negativo al virus) del paziente e dei suoi familiari.

#### 3.2.3.2. **Input**

L'email del paziente.

#### 3.2.3.3. **Elaborazione**

Vengono elencati i sintomi del paziente, se la email è presente nella piattaforma, e viene fatto un controllo sui familiari, per verificare se almeno uno di essi è positivo al Covid-19, per fornire le informazioni necessarie al Medico di Base per stabilire se serve o meno prenotare un tampone per il paziente in esame.

#### 3.2.3.4. **Output**

Visualizzazione di sintomi e se almeno uno o più familiari è positivo al Covid-19.

### 3.2.4. **Prenota tampone**

#### 3.2.4.1. **Introduzione**

Permette al medico di base o al sistema (dipendente ASL) di prenotare un tampone per un dato paziente.

#### 3.2.4.2. **Input**

Email del paziente, data in cui effettuare il tampone.

#### 3.2.4.3. **Elaborazione**

Viene registrata la data in cui verrà effettuato il tampone al paziente nell'archivio destinato a contenerla.

#### 3.2.4.4. **Output**

Viene creata la prenotazione nel sistema.

### **3.2.5. Inserisci sintomi**

#### **3.2.5.1. Introduzione**

Permette al paziente di inserire i propri sintomi.

#### **3.2.5.2. Input**

Numero dei sintomi, nomi, intensità e data in cui sono comparsi.

#### **3.2.5.3. Elaborazione**

Il paziente inserisce il numero dei sintomi che vuole registrare nel sistema. Il sistema permette l'inserimento solo dei sintomi associabili al Covid-19 (tosse secca, spossatezza, congiuntivite, diarrea, febbre, dolori muscolari, difficoltà respiratorie, perdita di gusto, perdita di olfatto) con relativa intensità. Il paziente sceglie di inserire un numero di sintomi pari a quello specificato in ingresso, scegliendoli tra questi.

#### **3.2.5.4. Output**

Vengono registrati i sintomi del paziente nell'apposito archivio.

### **3.2.6. Inserisci risultato tampone**

#### **3.2.6.1. Introduzione**

Permette a un dipendente dell'ASL di inserire il risultato di un tampone per un paziente.

#### **3.2.6.2. Input**

Id ed esito del tampone.

#### **3.2.6.3. Elaborazione**

Il sistema controlla che esiste un tampone con quell'id, in attesa di riceverne l'esito e, se individuato, viene registrato. Inoltre, in automatico, il sistema aggiornerà i dati del paziente come segue: se il tampone è positivo, sarà prenotato per il paziente un altro tampone a distanza di 7 giorni e, se non lo è già, sarà modificato lo stato del paziente in positivo; se è negativo e lo stato del paziente è negativo (è il primo tampone), la quarantena terminerà, mentre se è negativo il tampone e lo stato del paziente è positivo, quest'ultimo cambierà in transizione, ovvero, per essere definitivamente negativo, dovrà risultare negativo a un ulteriore tampone, prenotato per 3 giorni dopo.

#### **3.2.6.4. Output**

Il risultato del tampone è inserito nel sistema e i dati del paziente sono aggiornati.

### **3.2.7. Richiesta aggiungi familiare**

#### **3.2.7.1. Introduzione**

Permette ad un paziente di inviare una richiesta di aggiunta al suo nucleo familiare ad un altro paziente registrato al sistema.

#### **3.2.7.2. Input**

Email del paziente da aggiungere

#### **3.2.7.3. Elaborazione**

Il paziente specifica l'email del potenziale familiare che vuole aggiungere al proprio nucleo e quest'ultimo ne riceve la notifica di aggiunta.

#### **3.2.7.4. Output**

Il paziente da aggiungere riceve una notifica.

### **3.2.8. Gestisci richiesta di aggiunta familiare**

#### **3.2.8.1. Introduzione**

Permette a un paziente di gestire una richiesta di aggiunta ad un nucleo familiare

#### **3.2.8.2. Input**

Decisione sulla richiesta di aggiunta (SI/NO).

**3.2.8.3. Elaborazione**

Il paziente deve scegliere se aggiungersi o meno al nucleo familiare per cui ha ricevuto la richiesta.

**3.2.8.4. Output**

Il paziente verrà aggiunto al nucleo familiare, se accetta la richiesta, altrimenti quest'ultima verrà eliminata dal sistema.

**3.2.9. Visualizzare i sintomi**

**3.2.9.1. Introduzione**

Permette al medico di base di visualizzare i sintomi di tutti i suoi pazienti, mentre permette al paziente di visualizzare i sintomi dei soli familiari.

**3.2.9.2. Input**

Email del paziente.

**3.2.9.3. Elaborazione**

Per un dato paziente, è visualizzata la lista dei sintomi inseriti precedentemente dal paziente stesso.

**3.2.9.4. Output**

Restituisce la lista dei sintomi del paziente.

**3.2.10. Visualizza stato generale dei familiari**

**3.2.10.1. Introduzione**

Permette al paziente di visualizzare i sintomi e lo stato di positività/negatività di tutti i suoi familiari.

**3.2.10.2. Input**

Email del paziente.

**3.2.10.3. Elaborazione**

Per ogni familiare, si richiedono i sintomi e lo stato.

**3.2.10.4. Output**

Viene stampato a video la lista dei sintomi e lo stato di ogni familiare del paziente.

**3.2.11. Visualizza esito tampone**

**3.2.11.1. Introduzione**

Permette al paziente di visualizzare l'esito dell'ultimo tampone effettuatoogli

**3.2.11.2. Input**

La sua email.

**3.2.11.3. Elaborazione**

Occorre verificare che per il paziente in esame esiste un tampone registrato nel sistema e che ne sia presente l'esito. Se è così, verrà visualizzato.

**3.2.11.4. Output**

Viene stampato a video l'esito del tampone e la data in cui è stato effettuato.

**3.2.12. Calcola resoconto giornaliero**

**3.2.12.1. Introduzione**

Il sistema, a fine giornata, calcola il resoconto giornaliero (tamponi effettuati e tamponi positivi del giorno).

**3.2.12.2. Input**

Lista delle ASL presenti in Campania.

**3.2.12.3. Elaborazione**

Per ciascuna ASL della regione Campania, viene richiesto il numero dei tamponi il cui esito è stato comunicato in data odierna e il numero dei positivi. Verranno sommati i risultati provenienti da tutte le ASL.

#### **3.2.12.4. Output**

Viene registrato nel sistema nell'apposito archivio, il resoconto giornaliero.

### **3.2.13. Visualizza resoconto giornaliero**

#### **3.2.13.1. Introduzione**

Permette a un utente generico di visualizzare il resoconto giornaliero

#### **3.2.13.2. Input**

Data di cui si vuole visualizzare il resoconto

#### **3.2.13.3. Elaborazione**

Inserita la data, il sistema restituisce il numero dei tamponi il cui esito è stato comunicato in quella data specificata e il numero dei positivi.

#### **3.2.13.4. Output**

Viene stampato a video il resoconto giornaliero per una specifica data.

### **3.3. Requisiti di prestazione**

Sono richiesti tempi di risposta ragionevolmente contenuti sia per le operazioni sugli archivi che per l'inizio di una stampa.

### **3.4. Vincoli di progetto**

Nessuno

### **3.5. Attributi del sistema software**

Il sistema sarà una Web Application che girerà su un server in locale, ovvero Tomcat 9.0 in una macchina montante il sistema operativo Windows 10.

Usabilità: l'applicazione web deve avere sezioni ben definite e deve essere facilmente utilizzabile e navigabile.

Evolubilità: Si deve garantire la facile modificabilità del sistema.

Il sistema è stato sviluppato adoperando l'attributo di Alta Coesione e Basso

Accoppiamento e ogni componente, package o classe svolge delle funzioni secondo certi criteri di coesione (es. il package Service oppure diverse classi Controller nel package omonimo separati per attori, etc.).

Persistenza: ogni entità del sistema viene memorizzata su un database per scongiurare perdite di dati dovute, ad esempio, alla mancanza di corrente durante l'attività del locale.

Separazione degli interessi e delle responsabilità: l'applicazione è stata pensata con un'architettura basata sull'MVC che permette una separazione degli interessi e delle responsabilità di questo tipo:

- User Interface: la parte di interazioni con l'utente è nel package View;
- Business Logic: la logica dell'applicazione è presente nei controller del package Controller;

- Domain: gli elementi di dominio sono presenti nel package Entity del Model.

Ed in particolare il Model presenta:

- Low-Level Business Services: i servizi di basso livello utilizzati dall'applicazione sono quelli relativi al package Service (es. operazioni CRUD)

### **3.6. Altri requisiti**

#### **3.6.1. Database**

Il sistema software utilizzerà un sistema di supporto di dati che è il "MySQL 8.0".

#### 4. Glossario dei dati

##### Dati Utente

- Nome  
Nome dell'utente
- Cognome  
Cognome dell'utente
- Email  
Email dell'utente
- Password  
Password dell'utente

##### Dati Paziente

- Stato Paziente  
Positivo, se il paziente è attualmente positivo al Covid-19, negativo, se non lo è, o in transizione, se è in attesa del secondo tampone consecutivo negativo, per essere definitivamente negativo
- Indirizzo  
Domicilio del paziente
- In Quarantena  
True, se il paziente è attualmente in quarantena, false altrimenti
- Id Famiglia  
I pazienti con lo stesso Id appartengono allo stesso nucleo familiare

##### Dati Medico di Base

- Indirizzo Sede  
Dove esercita il medico di Base

##### Dati Dipendente ASL

- Matricola  
Numero di matricola del dipendente

##### Dati Sintomo

- Nome  
Che tipo di sintomo è (quelli comuni al Covid-19 sono tosse secca, spossatezza, congiuntivite, diarrea, febbre, dolori muscolari, difficoltà respiratorie, perdita di gusto, perdita di olfatto)
- Intensità  
Tre tipi di intensità (debole, media, forte)
- Data Comparsa  
Giorno in cui sono comparsi i sintomi

##### Dati tampone

- Id  
Identificativo del tampone
- Data Prenotazione

Il giorno in cui è stato effettuato o sarà effettuato il tampone (serve per l'eventuale inizio della quarantena, poiché il paziente sarà in quarantena dal giorno in cui viene effettuato il tampone)

#### Dati Risultato Tampone

- Id  
Identificativo del risultato
- Esito  
Positivo, se positivo al Covid-19, negativo, se negativo al Covid-19, o null se l'esito non è stato ancora inserito nel sistema
- Data Inserimento  
Il giorno in cui è stato inserito il risultato del tampone (serve per stilare il resoconto giornaliero, che terrà in considerazione l'esito dei tamponi il cui esito è stato comunicato in quella determinata giornata)

#### Dati ASL

- Nome  
Nome del centro ASL
- Indirizzo  
Indirizzo civico in cui è situato il centro

#### Dati Resoconto Giornaliero

- Data  
Giorno del resoconto
- Tamponi Effettuati  
Numero di tamponi il cui esito è stato comunicato nella data sopra specificata
- Tamponi positivi  
Numero di tamponi positivi il cui esito è stato comunicato nella data sopra specificata

# **Progettazione e Sviluppo di Sistemi Software**

## **Documento Analisi dei Requisiti**

Andrea D'Arco M63001063

Pipicelli Giuseppe 000117607



# **Indice**

1. Attori
2. Descrizione degli scenari
3. Diagramma dei casi d'uso
4. Context Diagram
5. System Domain Model
6. Descrizione dettagliata dei requisiti da implementare
7. SSD dei requisiti da implementare

## 1. Attori

- Utente (attore primario);
- Paziente (attore primario);
- Medico Di Base (attore primario);
- Dipendente ASL (attore primario);

## 2. Descrizione degli scenari

- **Registrazione**

Caso d'uso: Registrazione – Specializzazione: Registrazione Paziente
ID: UC01.1
Breve descrizione: L'Utente inserisce le credenziali per iscriversi al sistema come Paziente
Attori primari: Utente
Attori secondari: Nessuno
Precondizioni: Nessuna
Sequenza degli eventi principale: 1. L'Utente specifica che vuole registrarsi come Paziente 2. L' Utente inserisce nome, cognome, email, password, Indirizzo Civico e seleziona a quale ASL afferisce 3. Il sistema registra l'Utente nel sistema come Paziente.
Postcondizioni: L'Utente viene registrato nel sistema come un Paziente, assegnandone uno stato negativo e non in quarantena
Sequenze degli eventi alternative:

2.1 Il sistema rileva che l'email è già presente nel sistema, il sistema riporterà l'Utente al punto 1.
Caso d'uso: Registrazione – Specializzazione: Registrazione del Medico di Base
ID: UC01.2
Breve descrizione: L'Utente inserisce le credenziali per iscriversi al sistema come Medico di Base
Attori primari: Utente
Attori secondari: Nessuno
Precondizioni: Nessuna
Sequenza degli eventi principale: 1. L'Utente specifica che vuole registrarsi come Medico di Base 2. L' Utente inserisce nome, cognome, email, password, Indirizzo della Sede in cui esercita e seleziona a quale ASL afferisce 3. Il sistema registra l'Utente nel sistema come Medico di Base
Postcondizioni: L'Utente viene registrato nel sistema come un Medico di Base
Sequenze degli eventi alternative:
2.1. Il sistema rileva che l'email è già presente nel sistema, il sistema riporterà l'Utente al punto 1.
Caso d'uso: Registrazione – Specializzazione: Registrazione del Dipendente dell'ASL
ID: UC01.3
Breve descrizione: L'Utente inserisce le credenziali per iscriversi al sistema come Dipendente dell'ASL
Attori primari: Utente

Attori secondari: Nessuno
Precondizioni: Nessuna
Sequenza degli eventi principale: 1. L'Utente specifica che vuole registrarsi come Dipendente dell'ASL 2. L' Utente inserisce nome, cognome, email e password, e seleziona a quale ASL afferisce 3. Il sistema registra l'Utente nel sistema come Dipendente dell'ASL
Postcondizioni: L'Utente viene registrato nel sistema come un Dipendente dell'ASL
Sequenze degli eventi alternative:
2.1. Il sistema rileva che l'email è già presente nel sistema, il sistema riporterà l'Utente al punto 1.

### ● Login

Caso d'uso: Login
ID: UC02
Breve descrizione: L'utente inserisce le credenziali per accedere al sistema
Attori primari: Utente
Attori secondari: Nessuno
Precondizioni: Nessuna

<p>Sequenza degli eventi principale:</p> <ol style="list-style-type: none"> <li>1. L'utente inserisce email e password</li> <li>2. Il sistema trova una corrispondenza, il sistema autenticcherà l'Utente come un Paziente, un Medico di Base o un Dipendente dell'ASL</li> <li>3. Il sistema mostrerà l'homepage del sito.</li> </ol>
<p>Postcondizioni:</p> <p>L' Utente viene autenticato dal sistema come un Paziente, un Medico di Base o un Dipendente dell'ASL</p>
<p>Sequenze degli eventi alternative:</p>
<p>2.1a. Il sistema non trova mail, il sistema riporterà l'Utente al punto 1.</p> <p>2.1b. La password non è corretta, il sistema riporterà l'Utente al punto 1.</p>

● **Monitora situazione generale del paziente**

Caso d'uso: Monitora situazione generale del paziente
ID: UC03
<p>Breve descrizione:</p> <p>Il medico di Base consulta la situazione generale del paziente controllandone sintomi e stato di positività suoi e del proprio nucleo familiare</p>
<p>Attori primari:</p> <p>Medico di Base</p>
<p>Attori secondari:</p> <p>Nessuno</p>
<p>Precondizioni:</p> <p>Il medico di Base deve aver effettuato l'accesso al sistema ed il paziente e i suoi familiari devono essere registrati nel sistema</p>

<p>Sequenza degli eventi principale:</p> <ol style="list-style-type: none"> <li>1. Il Medico di Base inserisce l'email del paziente</li> <li>2. Il sistema trova una corrispondenza con il paziente cercato</li> <li>3. Il Medico di Base richiede di visualizzare i sintomi del paziente</li> <li>4. Il sistema restituisce la lista dei sintomi del paziente</li> <li>5. Il Medico di Base richiede al sistema di verificare se sono presenti o meno, persone positive nel nucleo familiare del paziente</li> <li>6. Il sistema verificherà se ci sono o meno familiari positivi</li> </ol>
<p>Postcondizioni:</p> <p>Visualizzazione dei sintomi del paziente e di eventuale presenza di positivi nel suo nucleo familiare</p>
<p>Sequenze degli eventi alternative:</p>
<p>2.1 Il sistema non trova una corrispondenza col paziente e restituisce un messaggio di errore</p> <p>5.1 Il sistema non trova familiari per il paziente</p>

### ● Prenota tampone

<p>Caso d'uso: Prenota Tampone – Specializzazione: Prenota Tampone per il Medico di Base</p>
<p>ID: UC04.1</p>
<p>Breve descrizione:</p> <p>Il Medico di Base prenota un tampone specificando l'utente e la data in cui effettuarlo</p>
<p>Attori primari:</p> <p>Medico di Base</p>
<p>Attori secondari:</p> <p>Nessuno</p>
<p>Precondizioni:</p> <p>Il Medico di Base deve aver prima monitorato la situazione generale del paziente</p>
<p>Sequenza degli eventi principale:</p> <ol style="list-style-type: none"> <li>1. Il Medico di Base inserisce l'email e la data in cui effettuare il tampone</li> <li>2. Il sistema crea un tampone con paziente e data specificati</li> </ol>

<b>Postcondizioni:</b> Il sistema ha creato una istanza di tampone
Caso d'uso: Prenota Tampone – Specializzazione: Prenota Tampone automaticamente
<b>ID:</b> UC04.2
<b>Breve descrizione:</b> Il Sistema prenota automaticamente un tampone per un paziente al verificarsi di determinate condizioni, dopo che il Dipendente dell'ASL ne ha inserito l'esito del tampone, specificando l'utente e la data in cui effettuarlo
<b>Attori primari:</b> Dipendente ASL (attiva caso d'uso Inserisci risultato tampone)
<b>Attori secondario:</b> nessuno
<b>Precondizioni:</b> Il Dipendente dell'ASL deve aver inserito il risultato del tampone per il paziente, che necessita di un altro tampone
<b>Sequenza degli eventi principale:</b> 1. Viene automaticamente inserito email e data in cui effettuare il tampone 2. Il sistema crea un tampone con paziente e data specificati
<b>Postcondizioni:</b> Il sistema ha creato una istanza di tampone

### ● **Inserisci sintomi**

Caso d'uso: Inserisci Sintomi
<b>ID:</b> UC05
<b>Breve descrizione:</b> il Paziente inserisce i propri sintomi nel sistema
<b>Attori primari:</b> Paziente

### 1.1.1

Attori secondari: Nessuno
Precondizioni: Il paziente deve aver effettuato l'accesso al sistema
Sequenza degli eventi principale: <ol style="list-style-type: none"> <li>1. Il Paziente inserisce il numero dei sintomi</li> <li>2. Il sistema chiede per ogni sintomo di inserirne la tipologia</li> <li>3. Il paziente inserisce la tipologia di ciascun sintomo</li> </ol>
Postcondizioni: Il sistema registra i sintomi del paziente

- **Inserisci Risultato tampone**

Caso d'uso: Inserisci Risultato tampone
ID: UC06
Breve descrizione: Il dipendente dell'ASL inserisce il risultato del tampone nel sistema. L'esito del tampone determinerà l'aggiornamento di informazioni sul paziente
Attori primari: Dipendente ASL
Attori secondari: Nessuno
Precondizioni: L'istanza del tampone per un determinato paziente deve esistere, perché già è stato prenotato, e il dipendente dell'ASL deve aver effettuato l'accesso al sistema
Sequenza degli eventi principale: <ol style="list-style-type: none"> <li>1. Il dipendente dell'ASL inserisce l'id della prenotazione del tampone con relativo esito</li> <li>2. Il sistema trova una corrispondenza e crea il risultato del tampone, inserendone anche la data</li> <li>3. Il sistema aggiorna i dati sullo stato e sulla quarantena del paziente e, se necessario, prenoterà un ulteriore tampone</li> </ol>



Postcondizioni: Il risultato del tampone è creato e i dati del paziente sono aggiornati
Sequenze degli eventi alternative:
2.1 Il sistema non trova una corrispondenza, il sistema riporterà il dipendente dell'ASL al punto 1.

● **Richiesta aggiungi familiare**

Caso d'uso: Richiesta Aggiungi familiare
ID: UC07
Breve descrizione: Permette ad un paziente di inviare una richiesta di aggiunta al suo nucleo familiare ad un altro paziente registrato al sistema.
Attori primari: Paziente
Attori secondari: Nessuno
Precondizioni: Il paziente ha effettuato l'accesso al sistema
Sequenza degli eventi principale: 1. Il paziente inserisce l'email del potenziale familiare 2. Il sistema trova una corrispondenza e invia una richiesta di aggiunta al familiare
Postcondizioni: Il potenziale familiare riceve una richiesta di aggiunta ad un nucleo familiare
Sequenze degli eventi alternative:
2.1 Il sistema non trova una corrispondenza, il sistema riporterà il paziente al punto 1.

- **Gestisci richiesta**

Caso d'uso: Gestisci richiesta
ID: UC08
Breve descrizione: Permette a un paziente di gestire una richiesta di aggiunta ad un nucleo familiare
Attori primari: Paziente
Attori secondari: Nessuno
Precondizioni: Il paziente ha effettuato l'accesso al sistema e gli è stata inviata una richiesta di aggiunta
Sequenza degli eventi principale: 1. Il paziente clicca sul pulsante SI 2. Il sistema inserisce il paziente nel nucleo familiare dell'altro paziente che ne ha inviato la richiesta
Postcondizioni: Il paziente viene aggiunto al nucleo familiare dell'altro paziente che ne ha inviato la richiesta
Sequenze degli eventi alternative:
1.1 Il paziente clicca sul pulsante NO 1.2 Il sistema cancella la richiesta di aggiunta familiare

- **Visualizza sintomi**

Caso d'uso: Visualizza Sintomi – Specializzazione: Visualizza sintomi per il Medico di Base
ID: UC09.1

<p>Breve descrizione: Permette al medico di base di visualizzare i sintomi di tutti i suoi pazienti</p>
<p>Attori primari: Medico di Base</p>
<p>Attori secondari: Nessuno</p>
<p>Precondizioni: Il medico di base sta monitorando un determinato paziente</p>
<p>Sequenza degli eventi principale:</p> <ol style="list-style-type: none"> <li>1. Il medico di base chiede di visualizzare i sintomi del paziente</li> <li>2. Il sistema fornisce la lista dei sintomi del paziente</li> </ol>
<p>Postcondizioni: Il medico di base visualizza la lista dei sintomi</p>
<p>Caso d'uso: Visualizza sintomi – Specializzazione: Visualizza sintomi per il paziente</p>
<p>ID: UC09.2</p>
<p>Breve descrizione: Permette al paziente di visualizzare i sintomi dei soli familiari</p>
<p>Attori primari: Paziente</p>
<p>Attori secondari: Nessuno</p>
<p>Precondizioni: Il paziente ha chiesto di vedere la situazione generale del proprio nucleo familiare</p>
<p>Sequenza degli eventi principale:</p> <ol style="list-style-type: none"> <li>1. Il paziente chiede di visualizzare i sintomi di un familiare</li> <li>2. Il sistema fornisce la lista dei sintomi del paziente</li> </ol>
<p>Postcondizioni: Il paziente visualizza la lista dei sintomi</p>

- **Visualizza stato generale dei familiari**

Caso d'uso: Visualizza stato generale dei familiari
ID: UC10
Breve descrizione: Permette al paziente di visualizzare i sintomi e lo stato di positività/negatività di tutti i suoi familiari
Attori primari: Paziente
Attori secondari: Nessuno
Precondizioni: Il paziente ha effettuato l'accesso al sistema
Sequenza degli eventi principale: 1. Il paziente chiede di visualizzare i sintomi e lo stato di ogni familiare 2. Il sistema restituisce i sintomi e lo stato per ogni familiare del paziente
Postcondizioni: Il paziente visualizza i sintomi e lo stato di positività/negatività dei suoi familiari
Sequenze degli eventi alternative:
2.1. Il paziente non ha familiari, il sistema non restituisce nulla

- **Visualizza esito tampone**

Caso d'uso: Visualizza stato generale dei familiari
ID: UC11

<p>Breve descrizione: Permette al paziente di visualizzare l'esito dell'ultimo tampone effettuatoogli</p>
<p>Attori primari: Paziente</p>
<p>Attori secondari: Nessuno</p>
<p>Precondizioni: Il paziente ha effettuato l'accesso al sistema</p>
<p>Sequenza degli eventi principale:</p> <ol style="list-style-type: none"> <li>1. Il paziente chiede di visualizzare il risultato dell'ultimo tampone a cui è stato sottoposto</li> <li>2. Il sistema restituisce l'esito dell'ultimo tampone</li> </ol>
<p>Postcondizioni: Il paziente visualizza l'esito del tampone</p>

### ● Calcola resoconto giornaliero

Caso d'uso: Calcola resoconto giornaliero
ID: UC12
<p>Breve descrizione: Si vuole calcolare per ogni giorno il numero di tamponi effettuati ed il numero di positivi rilevati in quel giorno</p>
<p>Attori primari: Responsabile Regione Campania</p>
<p>Attori secondari: Nessuno</p>
<p>Precondizioni: Il Responsabile della Regione Campania deve aver effettuato l'accesso al sistema</p>

Sequenza degli eventi principale:

1. Per ciascuna ASL del territorio campano, il Responsabile della Regione Campania richiede il numero dei tamponi effettuati ed il numero di tamponi risultati positivi del giorno
2. Il sistema restituisce il numero dei tamponi effettuati e positivi del giorno ed aggiorna i conteggi
3. Il sistema restituisce il resoconto giornaliero contenente il totale

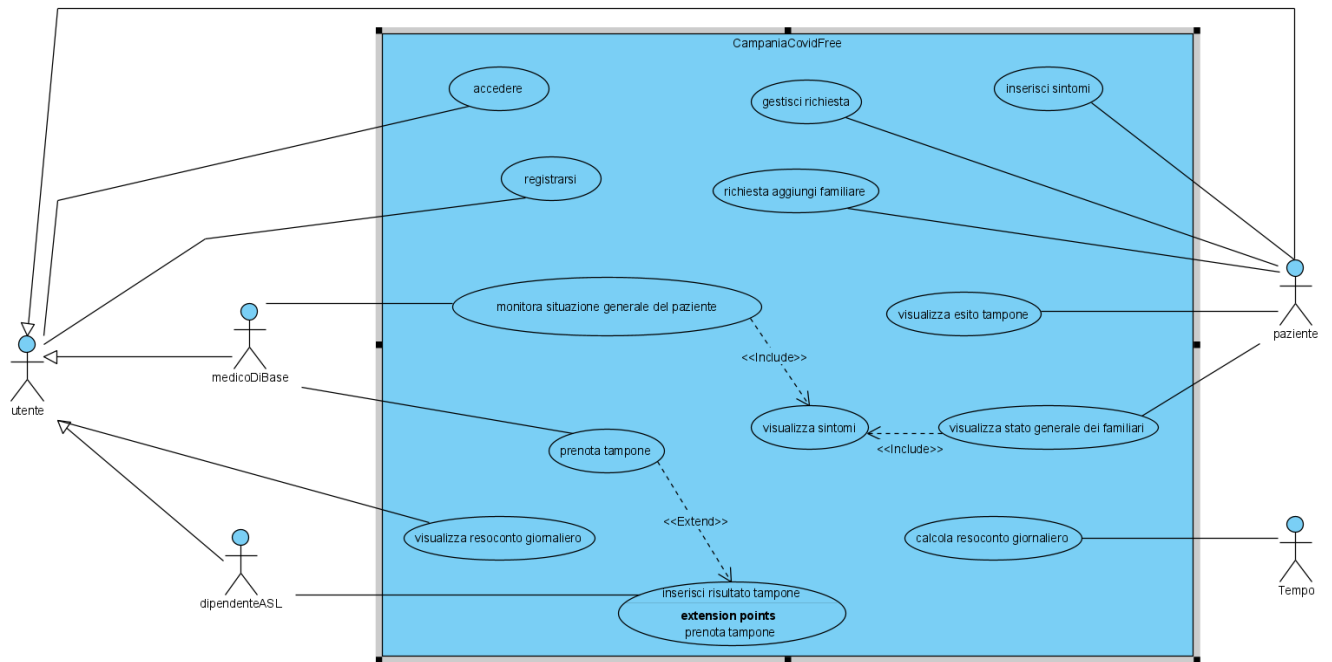
Postcondizioni:

Il resoconto giornaliero viene inserito nel sistema

### ● Visualizza resoconto giornaliero

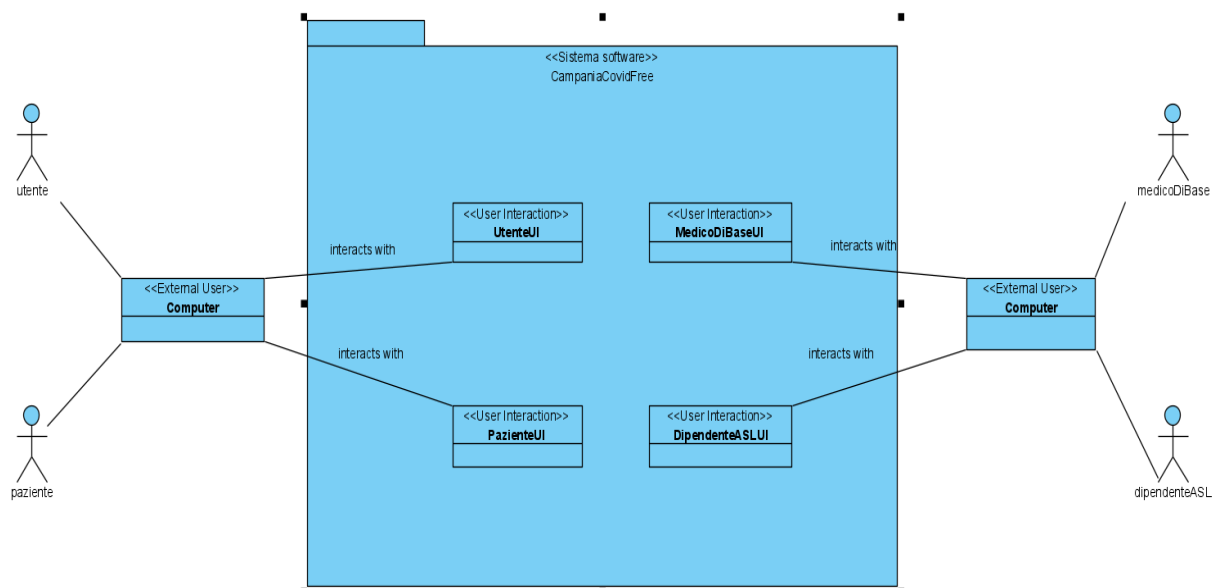
Caso d'uso: Visualizza resoconto giornaliero
ID: UC13
Breve descrizione: L'Utente non autenticato inserisce le credenziali presenti sul tavolo per accedere alla possibilità di effettuare un ordine
Attori primari: Utente
Attori secondari: Nessuno
Precondizioni: Nessuna
Sequenza degli eventi principale: 1. L'Utente specifica la data 2. Il sistema restituisce il resoconto giornaliero per quella data
Postcondizioni: L'Utente visualizza il resoconto giornaliero
Scenari alternativi:
2.1 Il sistema non trova nessun resoconto giornaliero per quella data

### 3. Diagramma dei casi d'uso

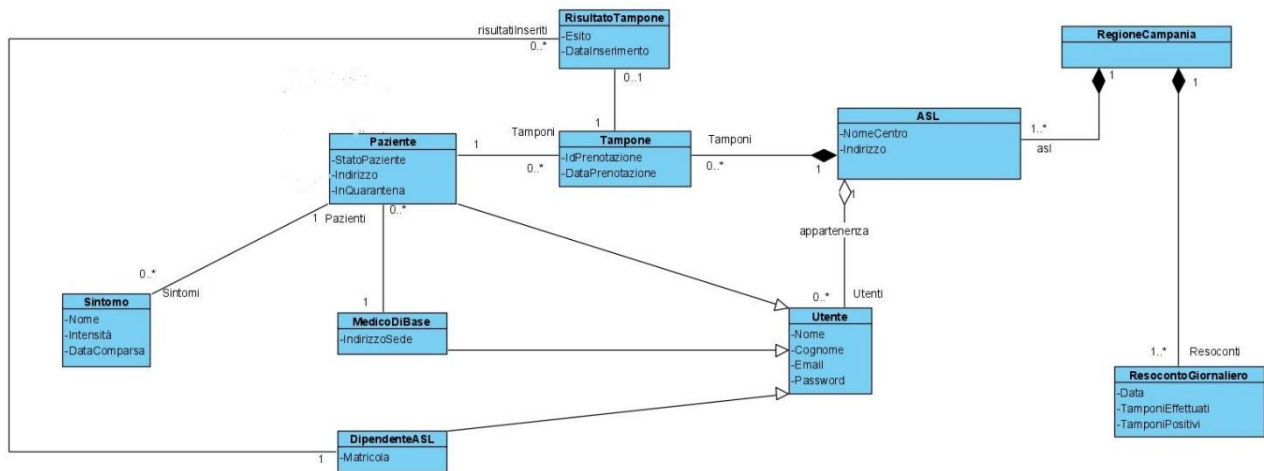


### 4. Context Diagram

Questo modello presenta il contesto in cui si trova il nostro sistema, con le interazioni che ha con gli attori, i quali accedono al sistema da computer, con relative user interface.



## 5. System Domain Model



## 6. Descrizione dettagliata dei requisiti da implementare

Le funzionalità che abbiamo scelto di implementare è “Inserisci risultato tampone”, che è esteso da “Prenota tampone”.

Per inserire il risultato di un tampone, il dipendente dell’ASL, dopo aver effettuato l’accesso, fornisce al sistema l’id di un tampone e relativo esito.

Se il tampone viene trovato, il cui esito non è stato ancora registrato, allora il sistema creerà un nuovo risultato associato a quel tampone.

Inoltre, in automatico, il sistema aggiornerà i dati del paziente come segue: se il tampone è positivo, sarà prenotato per il paziente un altro tampone a distanza di 7 giorni e, se non lo è già, sarà modificato lo stato del paziente in positivo; se è negativo e lo stato del paziente è negativo (è il primo tampone), la quarantena terminerà, mentre se è negativo il tampone e lo stato del paziente è positivo, quest'ultimo cambierà in transizione, ovvero, per essere definitivamente negativo, dovrà risultare negativo a un ulteriore tampone, prenotato per 3 giorni dopo.

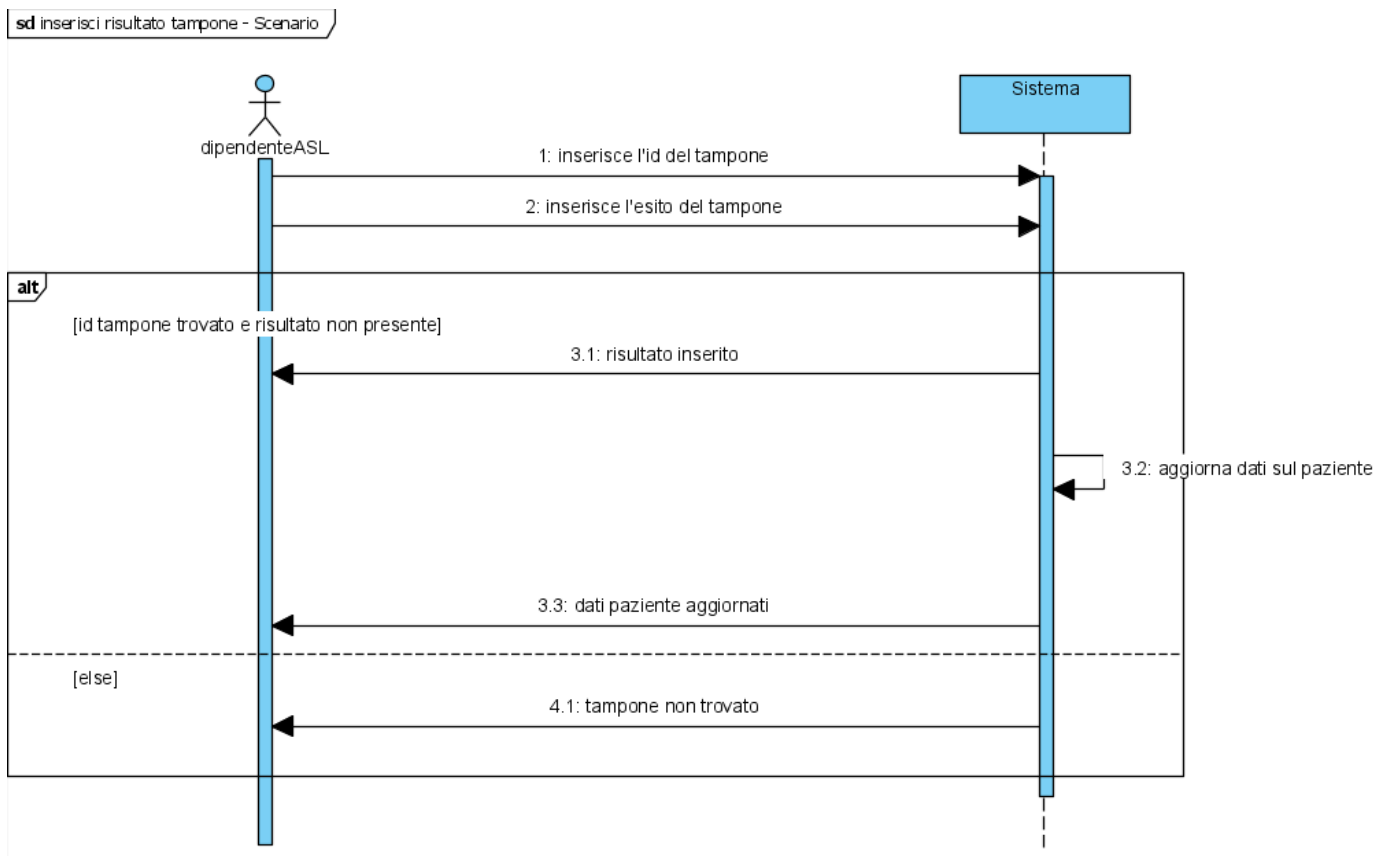
La funzionalità “Prenota tampone”, in questo caso interamente a carico del sistema, permette di creare un nuovo tampone con data di prenotazione e email del paziente forniti dal caso d’uso precedente.



## 7. SSD dei requisiti da implementare

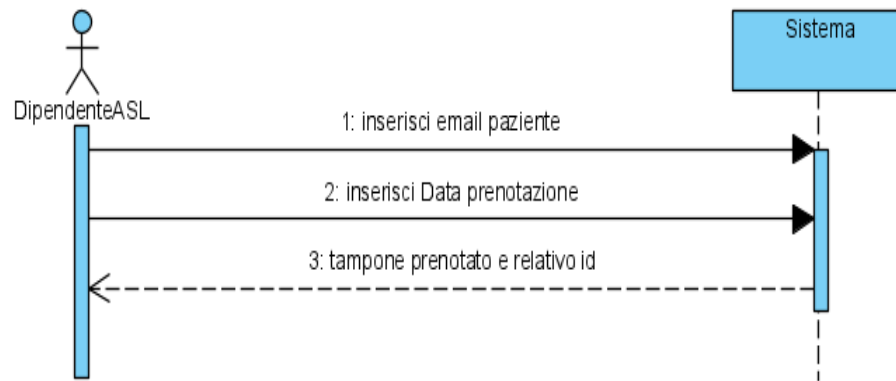
I seguenti diagrammi mostrano le interazioni dei vari attori mentre interagiscono con il sistema, descrivendo un primo flusso delle azioni. Si visualizzano gli SSD solo per le funzionalità da implementare, mentre le altre si trovano nel file “ProgettoPSSS.vpp”

### 1. “Inserisci risultato tampone”



## 2. “Prenota tampone”

sd prenota tampone - Scenario2



# **Progettazione e Sviluppo di Sistemi Software**

## **Architettura Progetto Software**

Andrea D'Arco M63001063

Pipicelli Giuseppe 000117607

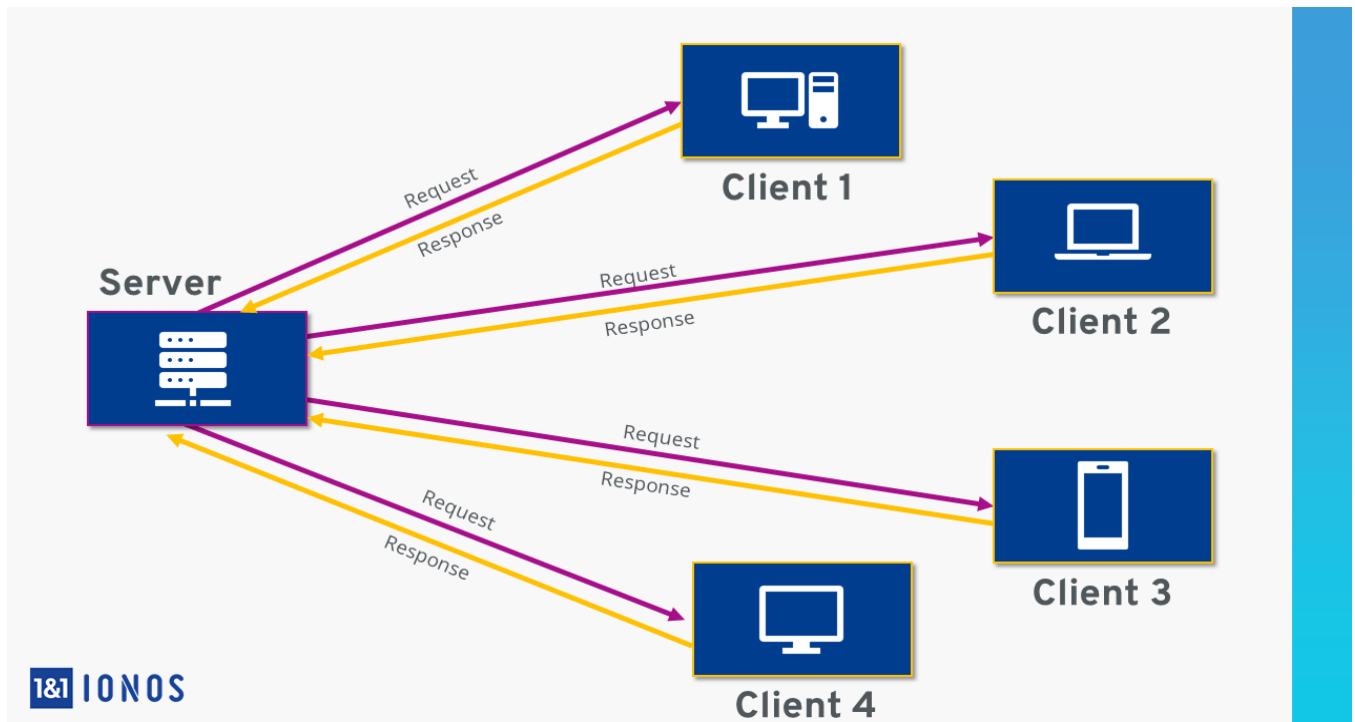
# **Indice**

1. Stile Architetturale
  - 1.1. Client-Server
2. Pattern usati
  - 2.1. MVC
  - 2.2. Repository
  - 2.3. State
3. Architettura Software
  - 3.1. Framework Spring MVC
4. Design
  - 4.1. Diagramma dei Componenti
  - 4.2. Diagramma di Deploy
5. Class Diagram di dettaglio
6. Diagramma Architetturale
  - 6.1. Vista DipendenteASL
7. Sequence Diagram di dettaglio
  - 7.1. “Inserisci Risultato Tampone”
  - 7.2. “Prenota Tampone”
  - 7.3. “Login Paziente”

## 1. Stile Architetturale

### 1.1. Client – Server

Lo stile architetturale scelto nel nostro sistema è il Client Server, in cui vi è un Client che fa richieste al Server e quest'ultimo le soddisfa. In particolare, siamo di fronte ad un “thin” Client, il quale non fa quasi nessuna elaborazione dei dati, ed un “fat” Server, che invece si occupa di tutta l’elaborazione e gestione dei dati.



## 2. Pattern usati

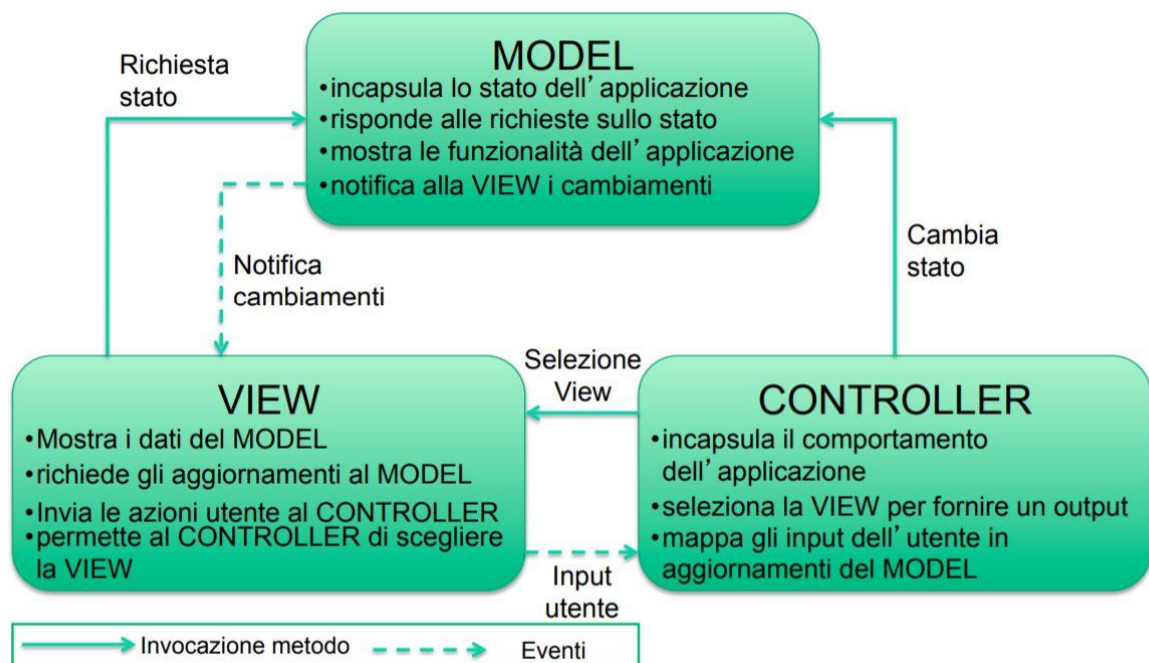
### 2.1. MVC

L'MVC implementa tre oggetti tra loro disaccoppiati: il controller, la vista ed il model.

- Controller - Il Controller `e la parte dell'applicazione che recepisce gli input dell'utente ed effettua, nel caso, richieste di aggiornamento verso il Model. Il Controller dovrà, inoltre, selezionare la vista per fornire un output. Esso conterrà gli oggetti che gestiranno l'interazione con gli altri 2 livelli. Il Controller è, quindi, un componente avente la responsabilità di trasformare le interazioni dell'utente sulla View in azioni eseguite dal Model, implementando la logica di controllo dell'applicazione.
- Model - Il Model contiene le classi le cui istanze rappresentano i dati da visualizzare e manipolare, ed è del tutto ignaro della presenza del Controller e della View. Dunque, il Model incapsula

lo stato dell'applicazione e risponde a richieste su di esso. È fondamentale per mantenere i dati e definire le regole di business, oltre che fornire le funzionalità necessarie per accesso ed aggiornamento a Controller e View, al fine di presentare dati sempre aggiornati e coerenti.

- View - La View contiene oggetti usati per rendere in output, nella user interface, i dati contenuti nel Model. Ha la funzionalità di richiedere aggiornamenti al model e di inviare le interazioni utente verso il Controller, il quale, in risposta, selezionerà la vista corretta da mostrare all'utente. È, dunque, la parte del sistema delegata a gestire la logica di presentazione.



A questa tipologia di MVC detta di tipo push, si affianca una seconda tipologia di tipo pull tipicamente utilizzata nelle soluzioni distribuite. Nel MVC di tipo pull la sequenza di azioni tra i tre componenti è leggermente diversa.

In particolare:

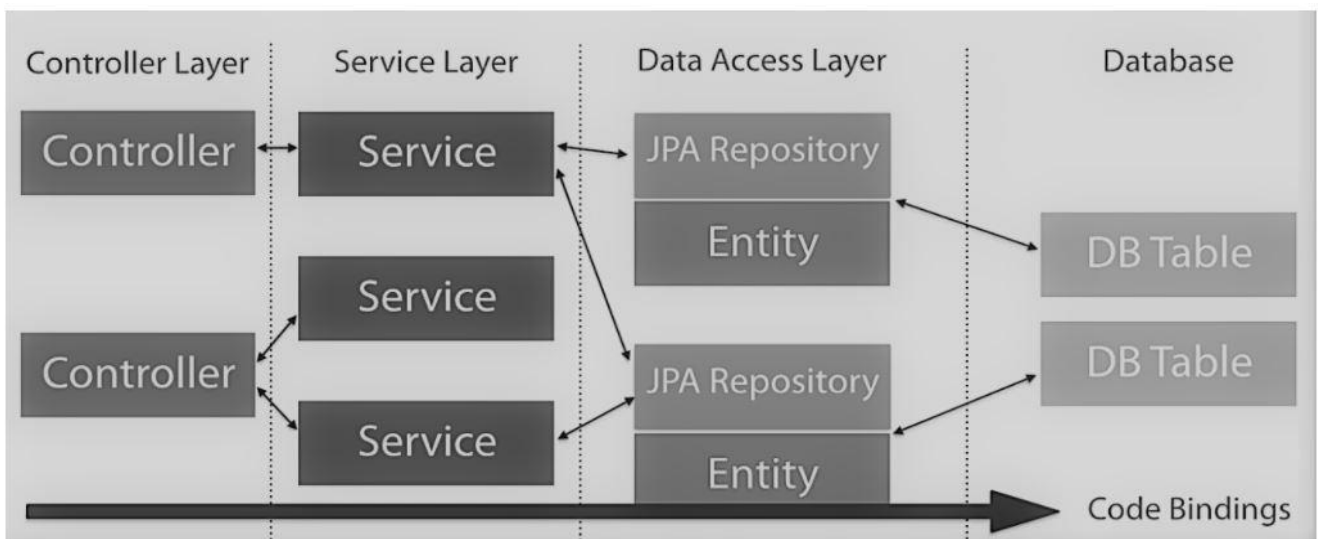
- La View riceve un input utente e lo invia al controller;
- Il Controller seleziona l'azione corrispondente all'input, che dovrà essere eseguita sul Model;
- Il Model esegue l'azione, si aggiorna e restituisce al controller il nuovo stato;

- Il Controller è a sua volta delegato di restituire alla View il nuovo stato del model.

## 2.2. Repository

Il Pattern Repository risponde al problema di forte accoppiamento tra le logiche di business e l'accesso ai dati introducendo un ulteriore livello, per l'appunto il Repository, il quale sarà il solo "responsabile" del recupero e salvataggio dai dati. Rende più facile la scrittura e la leggibilità del codice, favorisce il riuso, e permette di concentrarsi sull'implementazione dei requisiti anziché sull'interazione con il database.

L'architettura del pattern repository è così strutturata:

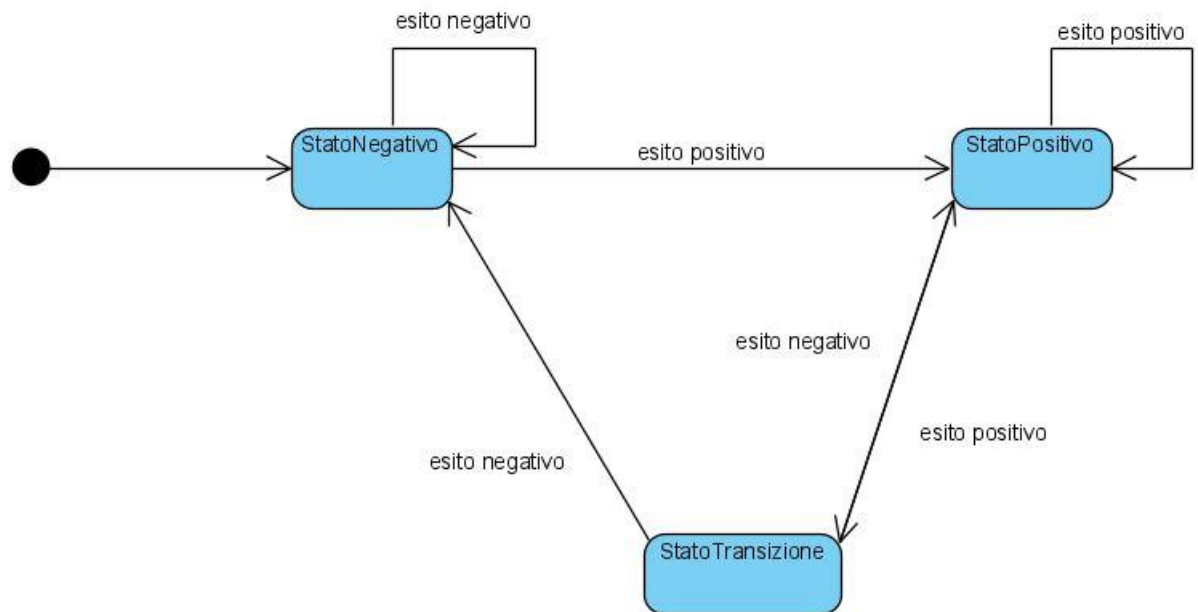


- Controller Layer: contiene la logica di business e coordina l'esecuzione delle funzionalità del sistema;
- Service Layer: consente di effettuare le operazioni elementari quali ricerca, creazione, aggiornamento e cancellazione;
- Data Access Layer: formato dalle interfacce JPA del Repository, che consentono l'interfacciamento col Database, e le entità del sistema, annotate con Hibernate per consentirne il "mapping" con le tabelle del Database.

## 2.3. State

Il Pattern State consente ad un oggetto modificare il suo comportamento quando il suo stato interno cambia. Il pattern incapsula, all'interno di una classe, il modo particolare in cui le operazioni di un oggetto (il Context) vengono svolte quando lo si trova in quello stato. Tipicamente, è caratterizzata da una classe astratta (o interfaccia) estesa (o realizzata, nel caso

dell'interfaccia) da delle classi che rappresentano gli stati concreti del Context, che implementano i metodi che permettono lo switch tra i vari stati. Nel nostro caso, l'abbiamo utilizzato per descrivere i tre stati del paziente, che sono StatoPositivo, StatoNegativo e StatoTransizione, il cui cambiamento di stato avviene come mostrato dalla seguente Macchina a stati:



### 3. Architettura software

Il nostro sistema software “CampaniaCovidFree” sfrutta lo stile architetturale Client-Server ed implementa il pattern MVC all'interno del server per realizzare una Web Application.

Per la sua realizzazione ci siamo serviti del framework Spring MVC, che implementa un pattern MVC di tipo pull.

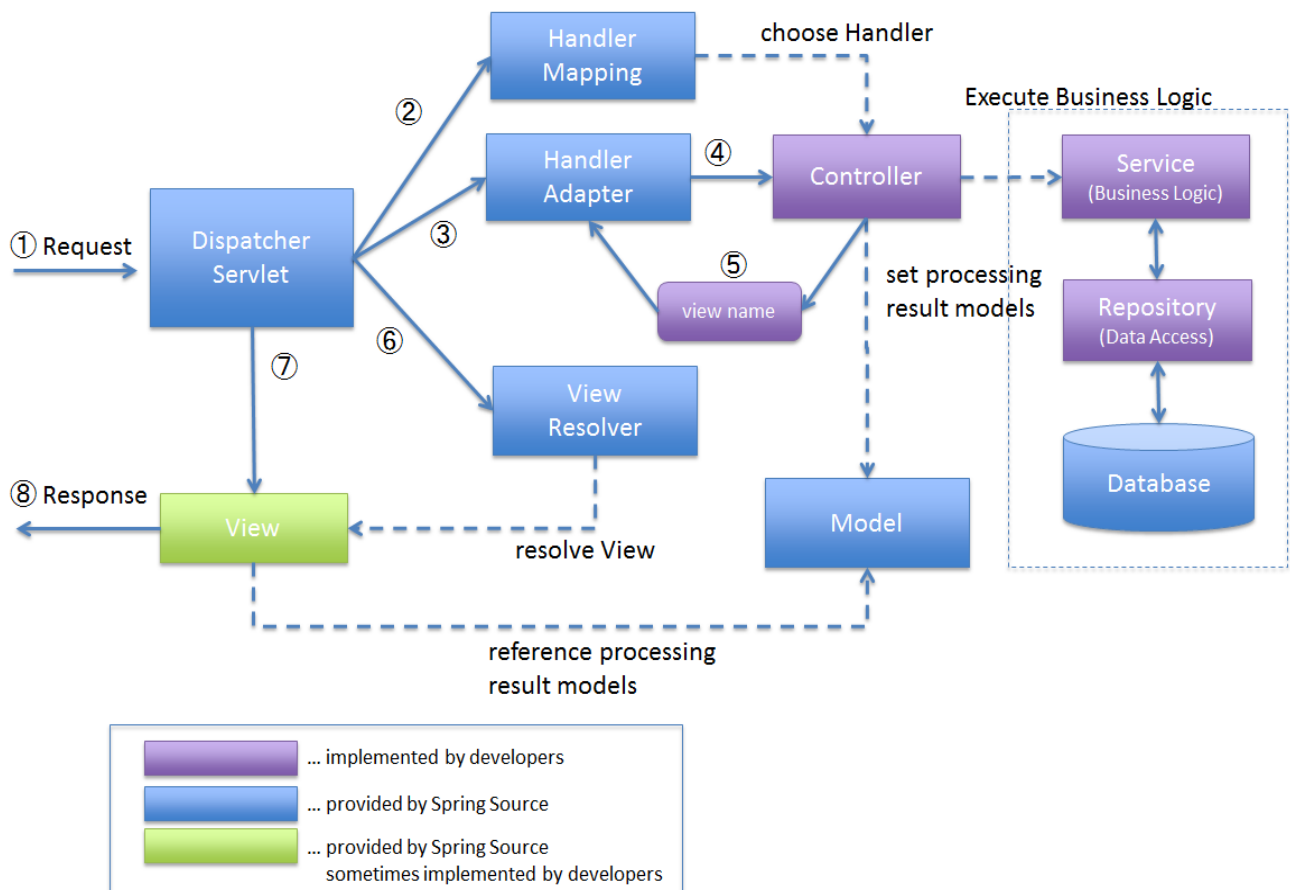
#### 3.1. Framework Spring MVC

In questa architettura, le richieste del client sono sottoposte al sistema mediante richieste HTTP, le quali sono intercettate dal Servlet Dispatcher e, tramite la logica implementata dall'Handler Mapping, sottopone le richieste al Controller opportuno.

Il Controller coordina la logica dell'applicazione e si serve dei vari Service e Repository del sistema per interfacciarsi con il Database.

Dopodichè il Controller fornirà il nome della View da mostrare al Client, selezionata dal View Resolver.





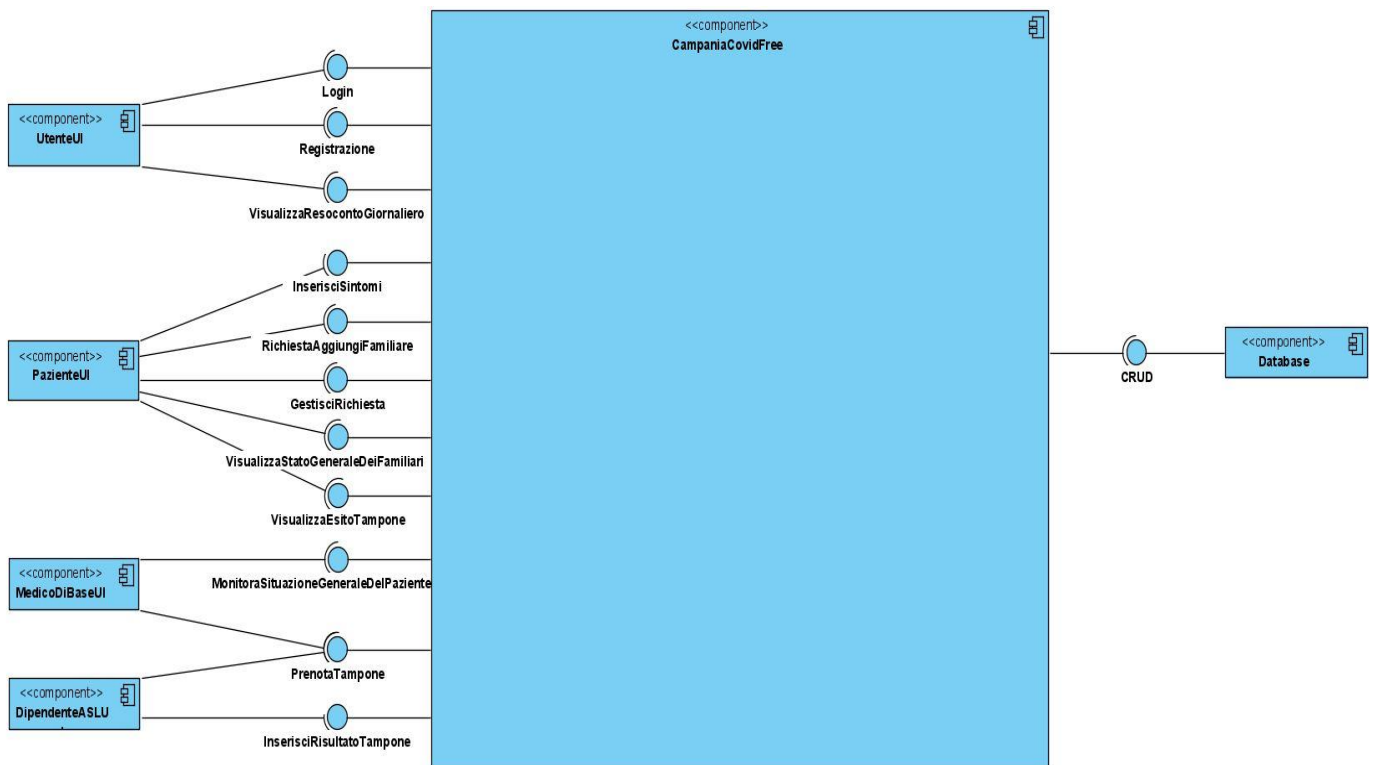
## 4. Design

### 4.1. Diagramma dei componenti

Il sistema “CampaniaCovidFree” si presenta come un unico componente, eseguibile autonomamente, a cui è possibile aggiungere ulteriori componenti che ne richiedono le stesse interfacce.

Nel dettaglio si ha:

- Un componente che permette di gestire la persistenza dei dati del sistema (Database MySQL Server)
- Un componente per ogni tipologia di utente con le relative interfacce per interagire con l'applicazione. Questi ultimi corrispondono a pagine HTML che verranno inviate al Client ed eseguite dal browser engine.



## 4.2. Diagramma di Deploy

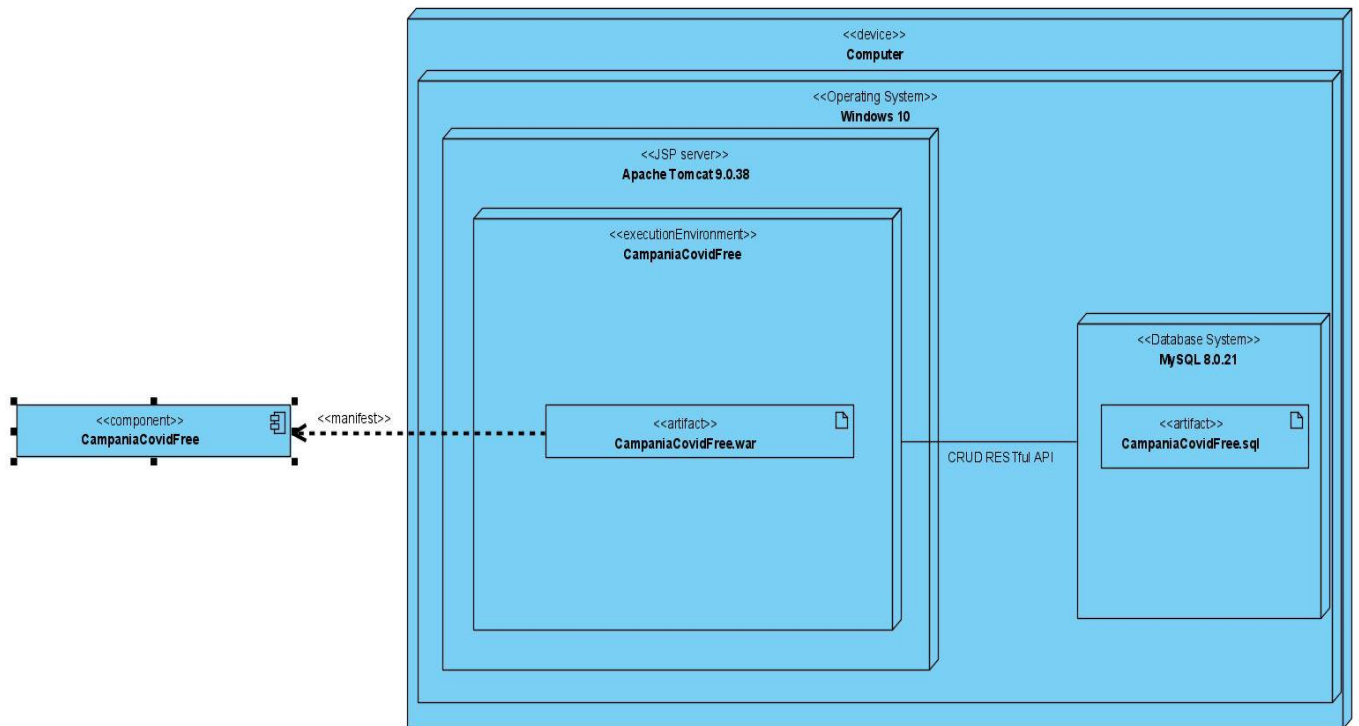
Il sistema "CampaniaCovidFree" può essere "deployato" su un unico nodo di elaborazione per funzionare.

Avremo il calcolatore, che funge da server, su cui saranno installati e configurati due software:

- Un Web Server Apache Tomcat versione 9.0.38;
- Un database MySQL Server 8.0.21 su cui installare il file "sql" (Structured Query Language) che contiene lo schema del database ed i dati necessari alla corretta esecuzione del sistema.

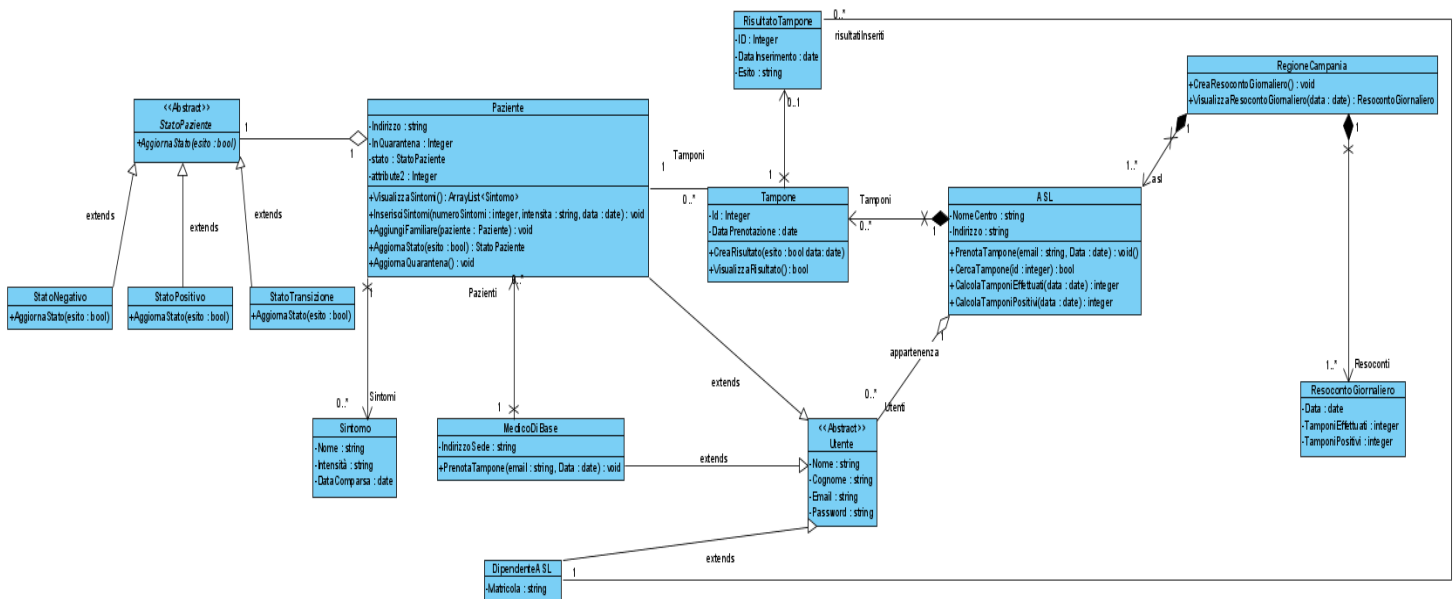
La comunicazione tra i due software avviene tramite MySQL JDBC driver, che fornisce l'accesso al database e le operazioni CRUD di base e permette all'applicazione di essere RESTful.

Infine, la manifestazione dell'artifact "CampaniaCovidFree.war" viene rappresentata dal componente CampaniaCovidFree.



## 5. Class Diagram di dettaglio

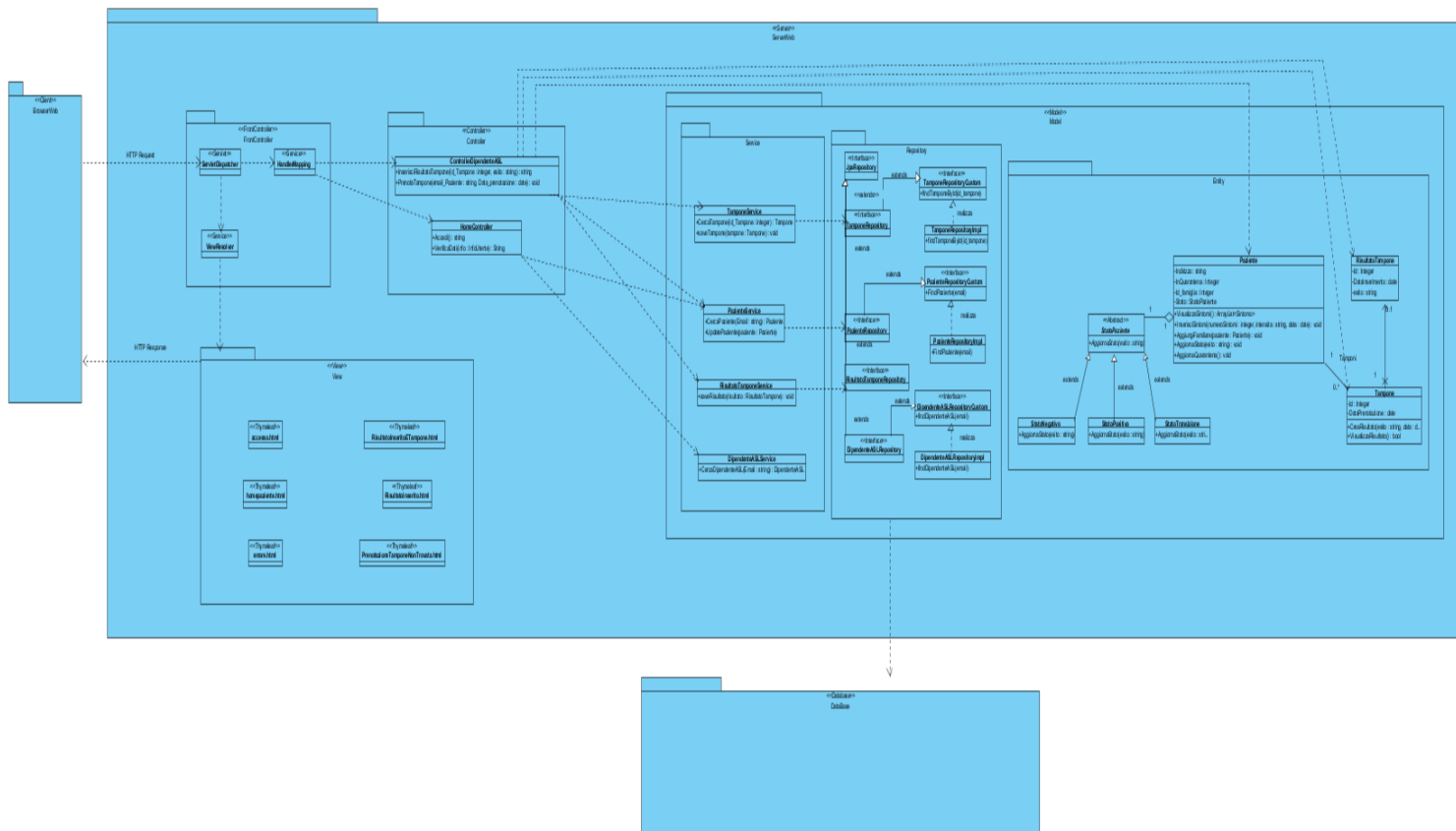
Nel class diagram di dettaglio sono stati inseriti tutti i metodi (con relativi parametri di ingresso e uscita) per manipolare i dati delle varie entità. In particolare, è stato introdotto il pattern State che spiega i vari stati associabili al paziente.



## 6. Diagramma Architeturale

### 6.1. Vista DipendenteASL

In questa vista sono rappresentati solo le classi di dominio interessate al caso d'uso "Inserisci risultato tampone" che può essere esteso da "PrenotaTampone", e dal caso d'uso di Login per Paziente e DipendenteASL, per facilitare la lettura del diagramma.



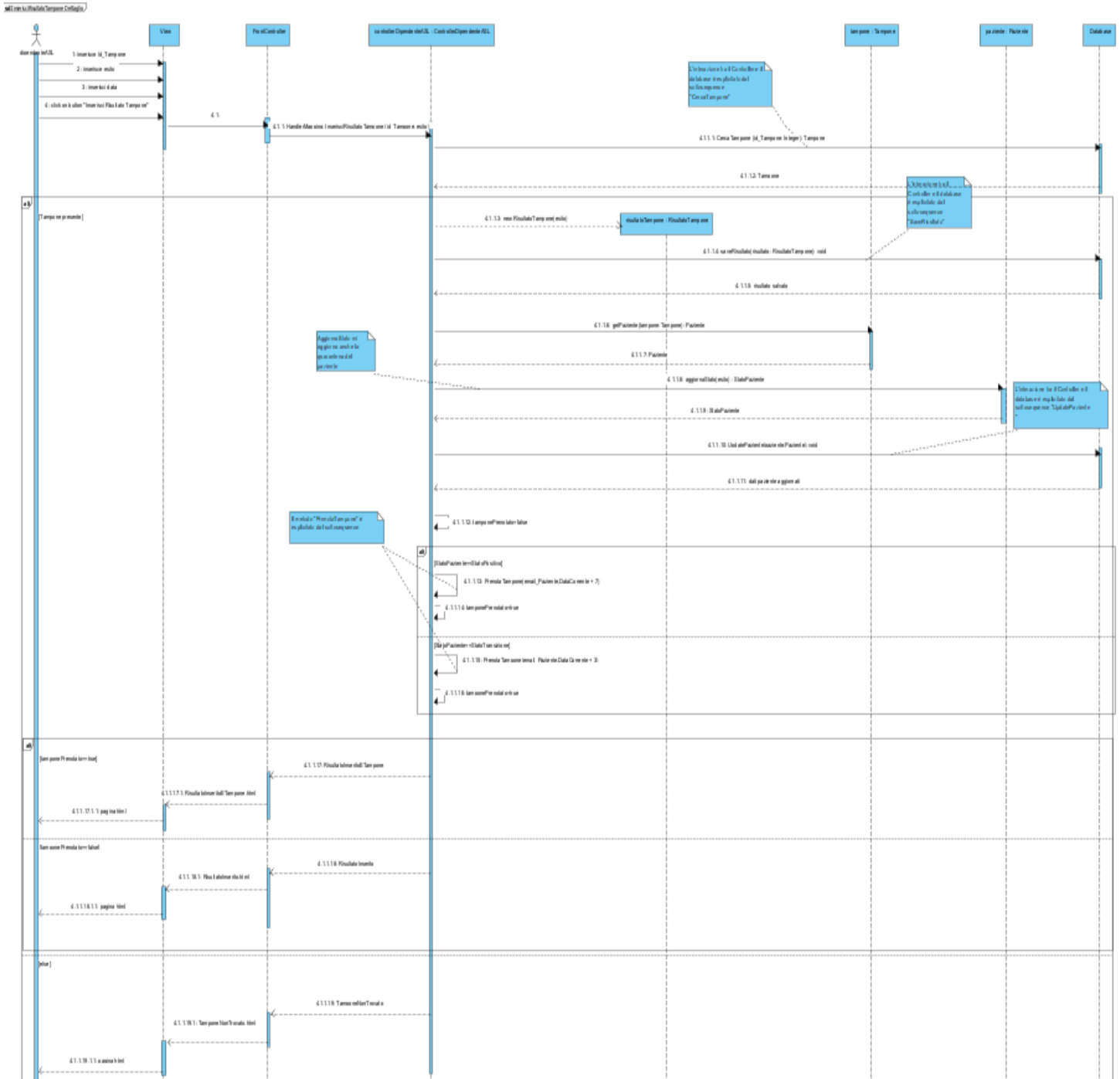
Le viste offerte dalla View sono scritte tramite Thymeleaf, un elaboratore di template JAVA XML/HTML, che unisce JSP ad HTML con una migliore integrazione per il framework Spring MVC.

Esternamente alla struttura Client-Server troviamo il componente logico database realizzato tramite MySQL Server che si trova esterno al sistema per garantirne maggiore modularità e scalabilità, riservandosi la possibilità di distribuirla in futuro anche su nodi diversi.

Dunque, si può vedere l'architettura come un Client-Server a 3 livelli dove il client richiede i servizi al server il quale a sua volta richiede i dati al database per poterli poi restituire al client.

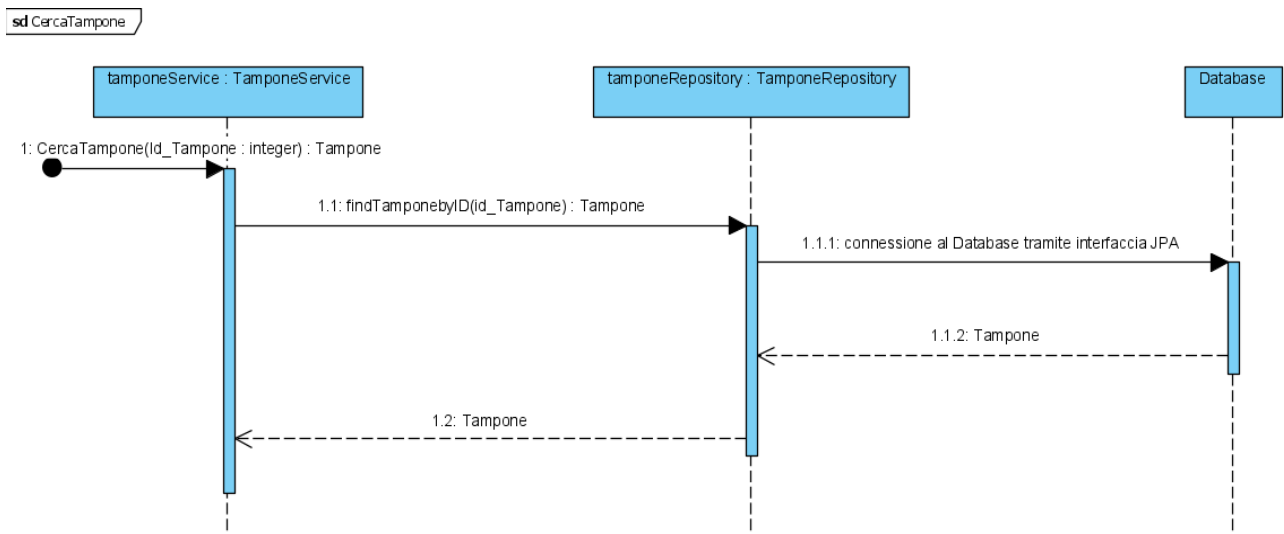
## 7. Sequence Diagram di dettaglio

### 7.1. Inserisci Risultato Tampone

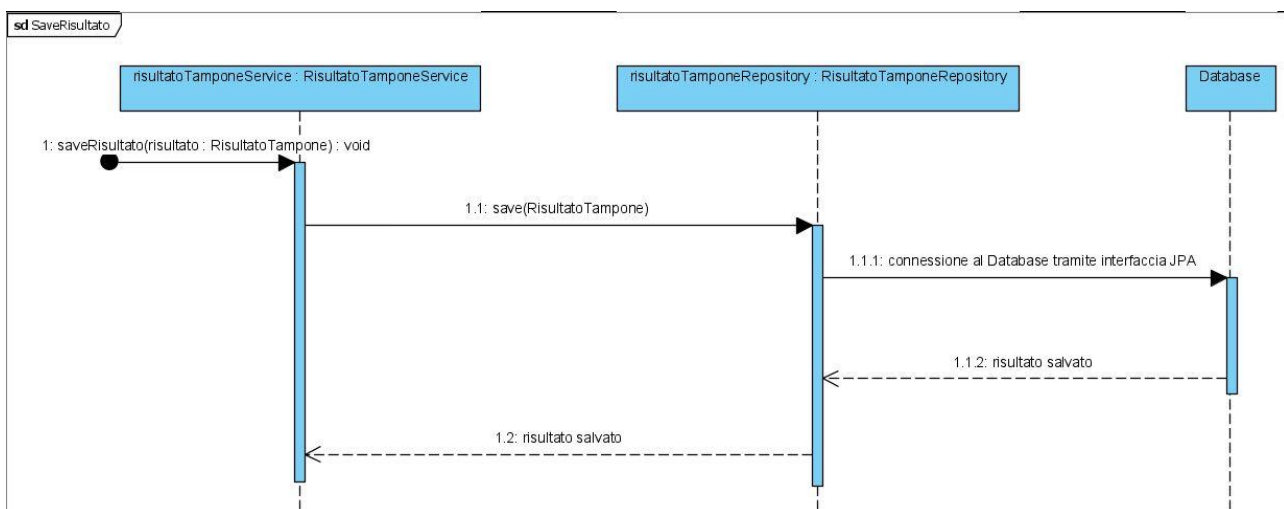


Inoltre, per rendere più leggibile questo sequence diagram, si è deciso di dividerlo in sottosequence;

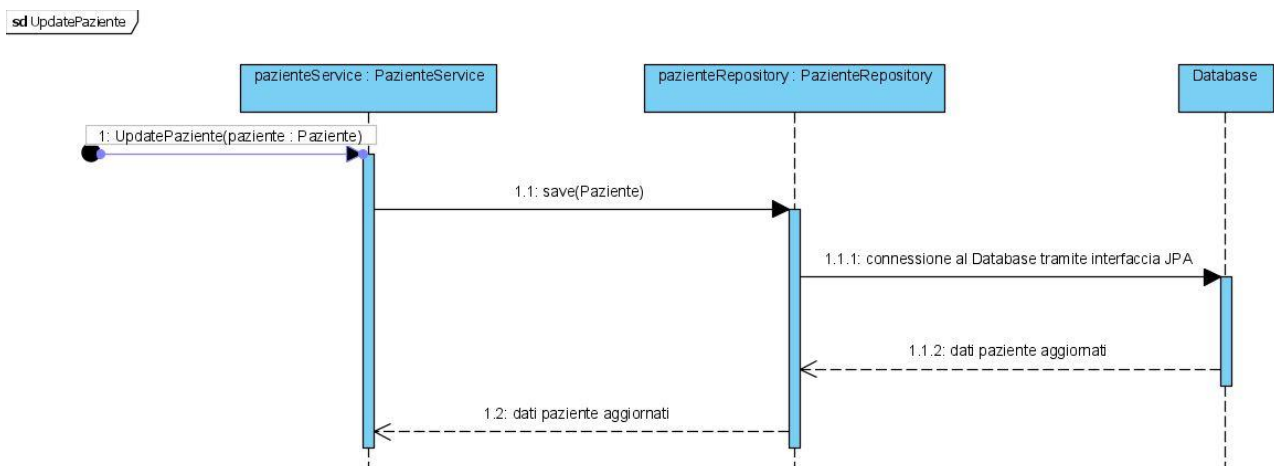
Il primo ad essere richiamato è “CercaTampone”, il cui sequence è questo:



Poi verrà richiamato “SaveRisultato”

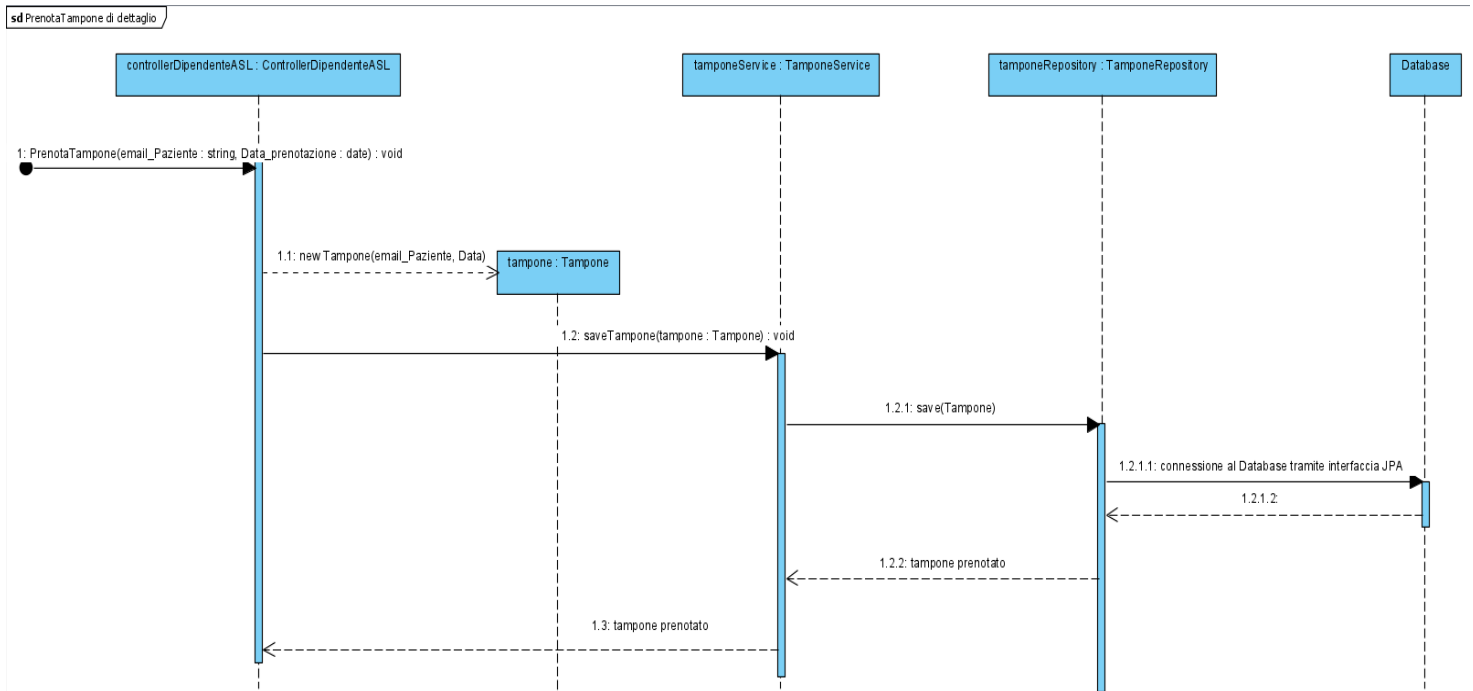


Infine, avremo “UpdatePaziente”



## 7.2. Prenota Tampone

Questo Diagramma di sequenza è riferito al caso d'uso “PrenotaTampone” che può estendere “InserisciRisultatoTampone”

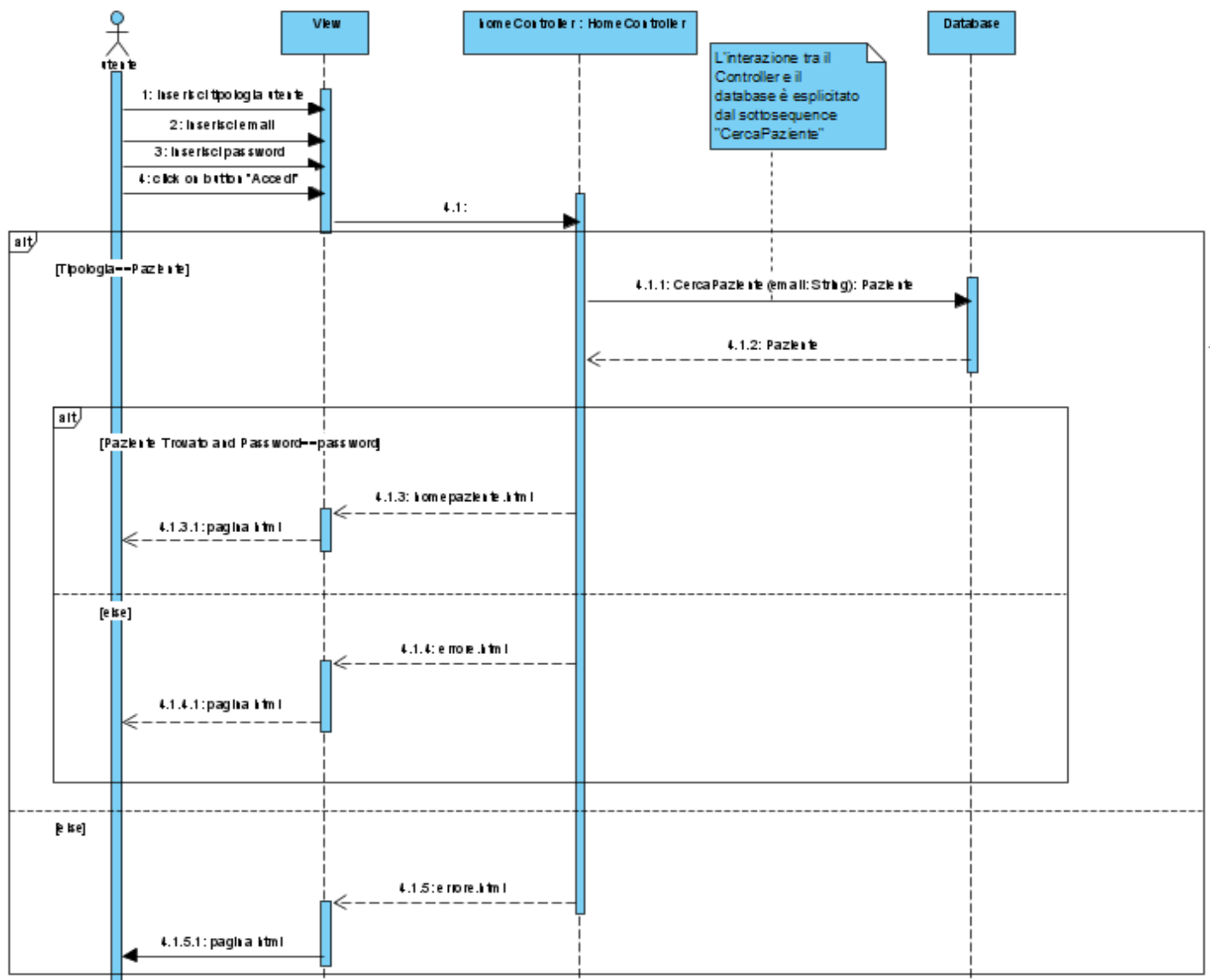


## 7.3. “Login Paziente”

Per semplicità, si è rappresentato solo il caso in cui la tipologia scelta sia Paziente. Nel caso la tipologia sia Medico di base o Dipendente ASL, il diagramma di sequenza è identico, tranne che per il fatto che useranno rispettivamente MedicoDiBaseService e MedicoDiBaseRepository il primo e DipendenteASLService e DipendenteASLRepository il secondo.

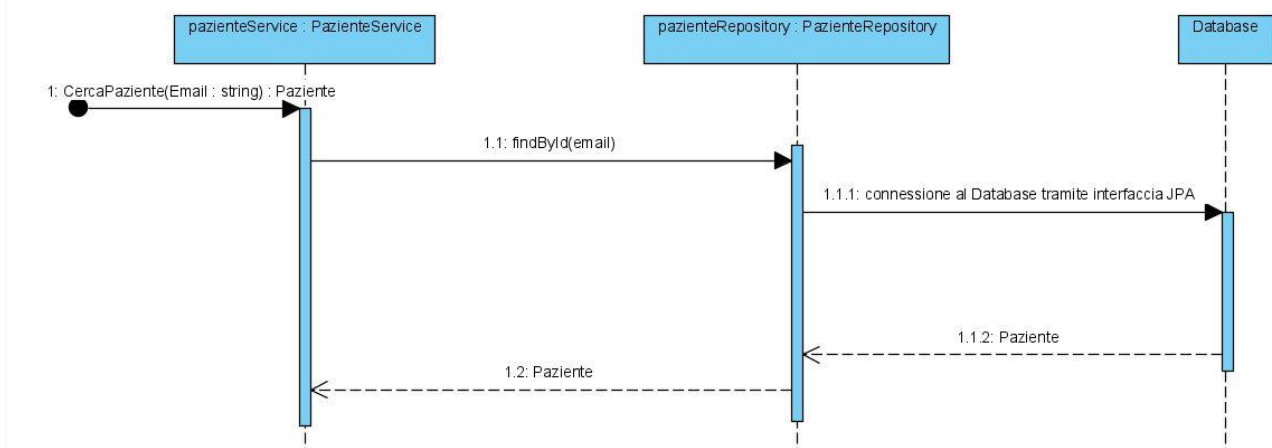
“LoginPaziente” userà il sottosequence “CercaPaziente” per cercare nel database se esiste un paziente registrato con mail e password stabilite in ingresso.

# sd LoginPaziente



Questo è, invece, "CercaPaziente"

# sd CercaPaziente





# **Progettazione e Sviluppo di Sistemi Software**

## **Documento di implementazione**

Andrea D'Arco M63001063

Pipicelli Giuseppe 000117607

# **Implementazione e deploy del sistema CampaniaCovidFree**

## **Strumenti per l'implementazione del software**

L'applicazione CampaniaCovidFree è stata sviluppata utilizzando il framework di sviluppo Spring. È stato utilizzato:

- Spring Boot per creare la struttura di base dell'applicazione già pronta all'uso con tanto di server associato (Tomcat 9.0) su cui eseguire (internamente all'IDE in fase di sviluppo), selezionando come dipendenze “Web”, “MySQL”, “JPA”, “DevTools” e “Thymeleaf”;
- Spring MVC per usufruire della sua architettura MVC particolarmente indicata per applicazioni web.
- Eclipse IDE 2020, per la scrittura del codice;

I linguaggi utilizzati sono stati:

- Java;
- HQL (Hibernate Query Language) per l'esecuzione di query personalizzate sul database;
- HTML associato a Thymeleaf e CSS.

## **Deploy del software**

Per effettuare il deploy dell'applicazione occorrono un Server ed un Database installati su un PC. L'accesso al sistema è attualmente limitato alla rete privata del nostro localhost, senza esporlo alla rete Internet esterna, ma, nelle implementazioni future, con le dovute modifiche e dopo aver gestito, in particolare, la sicurezza, è possibile rendere accessibile l'applicazione anche dall'esterno tramite connessione internet.

## **Installazione database**

Come database esterno di supporto all'applicazione si è scelto di utilizzare MySQL Server 8.0.

## Collegamento con il database

Nel file pom.xml, che serve a definire l'identità e la struttura del progetto tramite Maven, sono state aggiunte tutte le dipendenze che abbiamo selezionato con Spring Boot. Per collegare l'applicazione al database e le indicazioni ad Hibernate per la creazione ed aggiornamento del database, nel file application.properties, sono state inserite le seguenti istruzioni.

```
1 # == DATASOURCE ==
2 spring.datasource.url = jdbc:mysql://localhost:3306/CampaniaCovidFree
3 # ?useUnicode=true&useJDBCCompliantTimezoneShift=true&useLegacyDatetimeCode=false&serverTimezone=UTC
4 spring.datasource.username = CampaniaCovidFree
5 spring.datasource.password = Darco_Pipicelli
6
7 # == JPA / HIBERNATE ==
8 spring.jpa.show-sql = true
9 spring.jpa.hibernate.ddl-auto = update
10
11 # == THYMELEAF ==
12 spring.thymeleaf.cache = false
```

## Installazione e configurazione database

Per utilizzare il database, occorre innanzitutto scaricarlo all'indirizzo <https://dev.mysql.com/>, sezione downloads, versione installer per Windows.

Dopodiché occorre procedere all'installazione, selezionando il server, il connettore Java ed anche il workbench in modo da interagire in maniera più user-friendly con il database.

Continuando l'installazione, occorre definire la porta da assegnare al database. Come specificato anche nei file di configurazione del progetto, la porta è 3306, ossia quella di default, e la connessione TCP/IP.

Riguardo alla definizione dell'utente, attualmente le credenziali di accesso sono username="CampaniaCovidFree",

password="Darco\_Pipicelli".

Importante specificare anche il nome del servizio associato al server, MySQL di default, che dovrà essere in esecuzione ogni volta che si vorrà eseguire l'applicazione.

Installato il database, si crea una nuova connessione e si importa lo schema del database ed i dati al suo interno per una corretta esecuzione del sistema.

Dalla GUI fornita dal workbench è possibile accedere tramite le credenziali specificate in precedenza ed importare il file CampaniaCovidFree.sql.

## Server in uso

Come Application Web Server si è scelto di utilizzare Apache Tomcat 9.0.

## Istruzioni per l'installazione

Dopo aver decompilato il file “CampaniaCovidFree.zip”, da Eclipse, importiamo il progetto in questo modo:

File→Open Project From File System→ e importiamo tutta la Cartella CampaniaCovidFree.

Cliccando sul tasto destro del progetto, Run As→ Java Application → CampaniaCovidFreeApplication, partirà l'esecuzione del programma, e sarà in ascolto sul porto 8080 del localhost, all'indirizzo “localhost:8080”.


## Esecuzione software “CampaniaCovidFree”

Possiamo, ora, accedere all'applicazione all'indirizzo “localhost:8080” da un qualunque web browser.

La homepage è la seguente, dedicata agli utenti ancora non loggati, che posso loggarsi, effettuare il login, o visualizzare il resoconto giornaliero.



Cliccando su Login si apre una pagina per permettere agli utenti di autenticarsi al sistema



LOGIN

HOME

Log in

Benvenuto nella pagina di accesso!

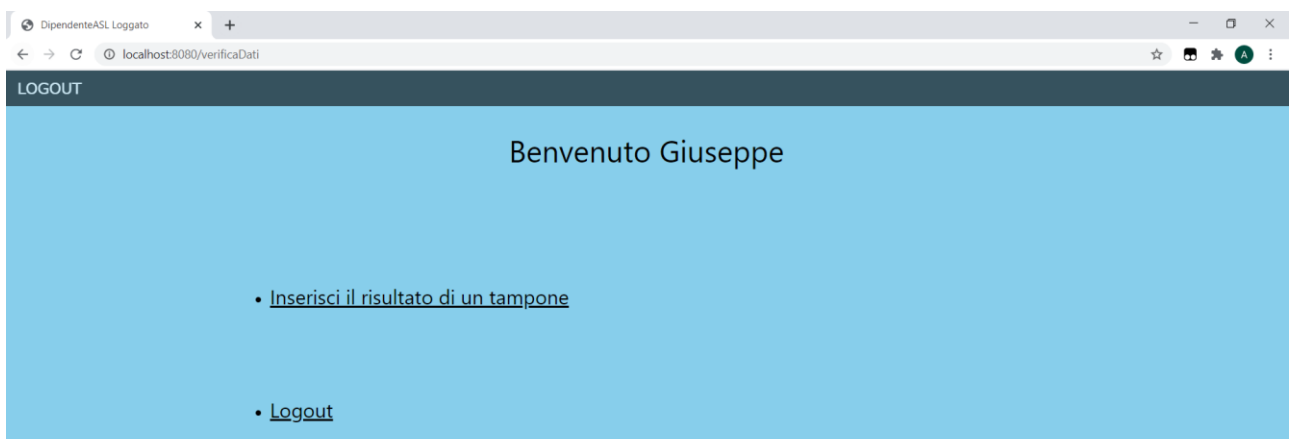
Tipologia :  
Dipendente ASL

Email :  
glu@hotmail.it

Password :  
1234

Accedi

Poiché il caso d'uso che abbiamo sviluppato permette al Dipendente ASL di inserire il risultato di un tampone, inserendo le credenziali valide, si apre una pagina dove ci sono dei link a cosa può fare.



DipendenteASL Loggato

LOGOUT

Benvenuto Giuseppe

- [Inserisci il risultato di un tampone](#)
- [Logout](#)

Cliccando su “Inserisci il risultato di un tampone”, possiamo inserire i dati del tampone da inserire.

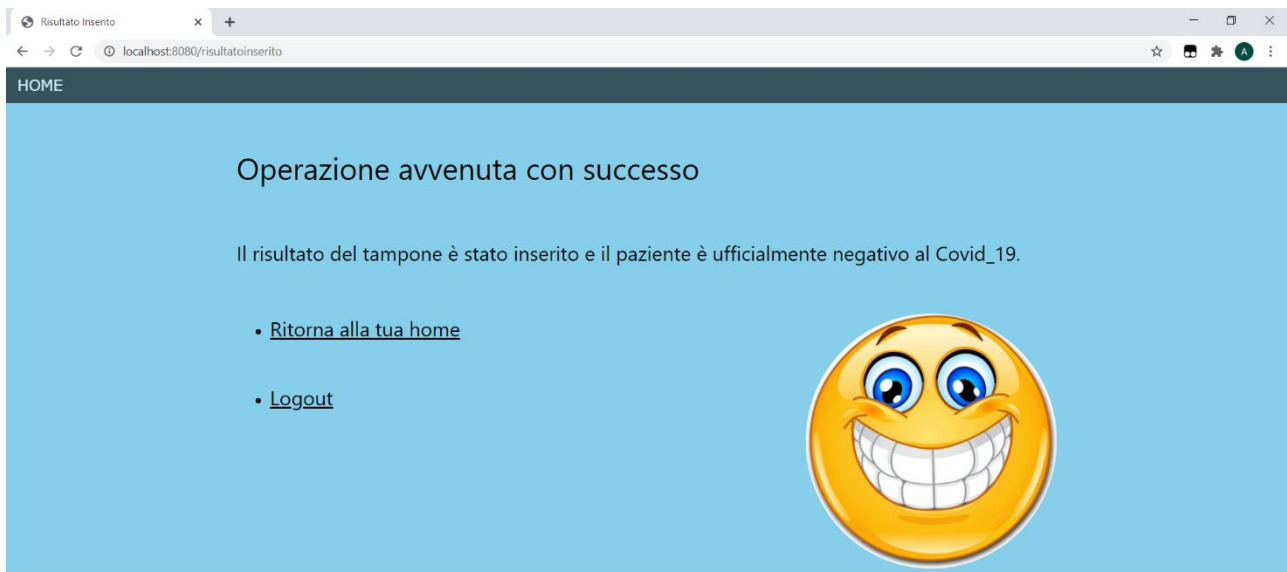


The screenshot shows a web browser window with the address bar displaying 'localhost:8080/InserisciRisultato'. The page has a dark blue header with the word 'HOME'. The main content area has a light blue background with the title 'Inserisci il risultato del tampone'. Below the title are three input fields: the first contains '11', the second contains '2020-11-26', and the third contains 'negativo'. A black button with the text 'Inserisci' is positioned below the third field. To the right of the button is a small image of a person in a blue medical uniform holding a swab stick.

Il tampone 11 che si è in procinto di inserire, si riferisce al seguente paziente, in cui il valore “1” identifica che il paziente è in quarantena, mentre lo stato è attualmente “transizione”, cioè sta attendendo il secondo tampone negativo per essere ufficialmente negativo.

andreadarco@hotmail.it	DArco	1	vincenzo Janfolla	Andrea	1234	ASL NAPOLI 1 CENTRO	porfi@hotmail.it	2	transizione
------------------------	-------	---	-------------------	--------	------	---------------------	------------------	---	-------------

Inserendo il tampone, ci uscirà questo.



Infatti, se andiamo a vedere i dati del paziente, saranno stati modificati i valori che si riferiscono alla quarantena del paziente (ora “0”), e allo stato (ora “negativo”)

andreadarco@hotmail.it	DArco	0	vincenzo Janfolla	Andrea	1234	ASL NAPOLI 1 CENTRO	porfi@hotmail.it	2	negativo
------------------------	-------	---	-------------------	--------	------	---------------------	------------------	---	----------

Cliccando nella pagina “Operazione avvenuta” il link “Ritorna alla tua home” (o, ugualmente, HOME dalla navbar), si ritorna alla home del dipendente, dove si ha la possibilità di inserire un altro tampone da parte dello stesso dipendente.

Analizziamo, ora, il seguente paziente, con valore “1”, che dice che il paziente è in quarantena, e con stato “negativo”.

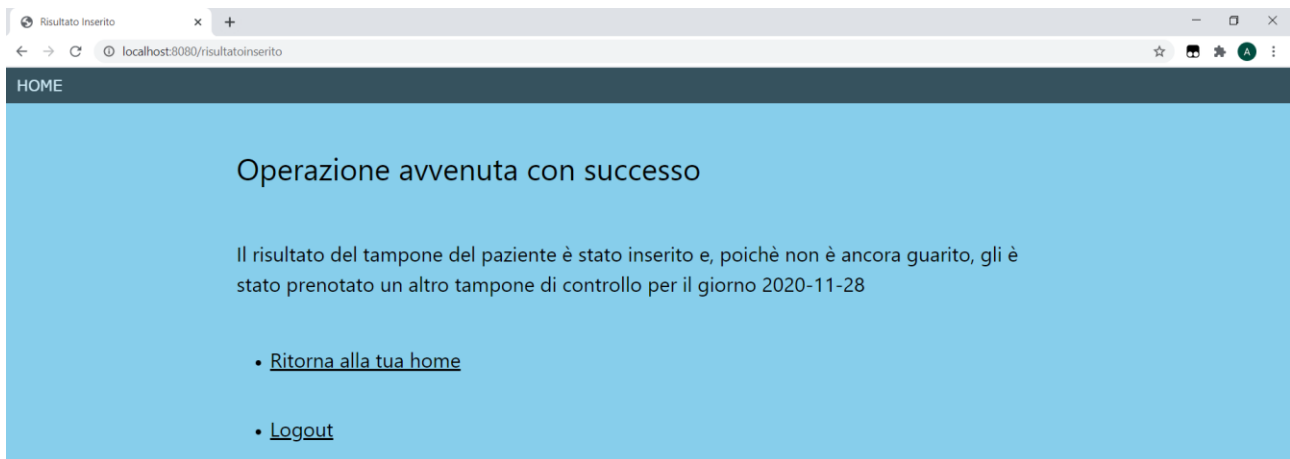
giuseppe.pipicelli96@gmail.com	Pipicelli	1	via cavallino	Giuseppe	5678	ASL NAPOLI 1 CENTRO	porfi@hotmail.it	2	negativo
--------------------------------	-----------	---	---------------	----------	------	---------------------	------------------	---	----------

Si provi a inserire i seguenti dati e cliccando “inserisci”:



The screenshot shows a web browser window with the address bar displaying 'localhost:8080/InserisciRisultato'. The page has a dark blue header with the word 'HOME' in white. The main content area has a light blue background with the title 'Inserisci il risultato del tampone' in dark blue. Below the title are three input fields: the first contains '12', the second contains '2020-11-22', and the third contains 'positivo'. Below these fields is a black button with the text 'Inserisci' in white. To the right of the button is a small image of a person in a blue uniform using a red swab stick to collect a sample from a test tube.

Si aprirà, poi, questa pagina:



The screenshot shows a web browser window with the address bar displaying 'localhost:8080/risultatoinserito'. The page has a dark blue header with the word 'HOME' in white. The main content area has a light blue background with the title 'Operazione avvenuta con successo' in dark blue. Below the title is a paragraph of text: 'Il risultato del tampone del paziente è stato inserito e, poichè non è ancora guarito, gli è stato prenotato un altro tampone di controllo per il giorno 2020-11-28'. Below the paragraph are two links: 'Ritorna alla tua home' and 'Logout', both underlined.

Infatti, il paziente, avendo ricevuto un risultato “positivo”, si vedrà modificare i suoi dati in questo modo:



giuseppe.pipicelli96@gmail.com	Pipicelli	1	via cavallino	Giuseppe	5678	ASL NAPOLI 1 CENTRO	porfi@hotmail.it	2	positivo
--------------------------------	-----------	---	---------------	----------	------	---------------------	------------------	---	----------

Infine, gli sarà prenotato automaticamente un tampone di controllo per il 28/11/2020, cioè 7 giorni dopo il precedente, come mostrato dalla seguente riga della tabella “tamponi”, del database.

13	2020-11-28	giuseppe.pipicelli96@gmail.com	ASL NAPOLI 1 CENTRO	NULL
----	------------	--------------------------------	---------------------	------

Il valore “null” si riferisce all’id del risultato del tampone, che è “null” in quanto non è stato ancora inserito il risultato.

# **Progettazione e Sviluppo di Sistemi Software**

## **Prove di test**

Andrea D'Arco M63001063

Pipicelli Giuseppe 000117607

# TESTING

La fase di testing è una fase fondamentale nella produzione e sviluppo di sistemi software, poiché consente di individuare carenze di correttezza, completezza e affidabilità all'interno del software sviluppato.

È buona pratica effettuare dei test alla fine di ogni iterazione al fine di controllare il corretto funzionamento del sistema dopo l'aggiunta di nuove funzionalità e di nuovi moduli.

Sviluppando il nostro progetto, il testing è stato svolto, principalmente, durante la seconda iterazione.

Abbiamo effettuato test per il caso d'uso "Login" e il caso d'uso "Inserisci Risultato Tampone", mantenendone i risultati nella seguente tabella. Un caso d'uso contrassegnato da **OK** è un caso d'uso dove ciò che ci aspettavamo coincideva con ciò che realmente è stato visualizzato in output.

N°Test	Azione	Aspettative	Realtà	Passato?
1	Login Paziente con input corretti	Dopo il login, viene indirizzato nella home	Dopo il login viene indirizzato nella home	OK
2	Login Dipendente con input corretti	Dopo il login viene indirizzato nella home	Dopo il login viene indirizzato nella home	OK
3	Login Medico con input corretti	Dopo il login viene indirizzato nella home	Dopo il login viene indirizzato nella home	OK
4	Login con input scorretti	Dopo il login viene indirizzato in una pagina di errore	Dopo il login viene indirizzato in una pagina di errore	OK
5	Logout	Dalla home di ciascun utente, effettuando il logout si viene indirizzati nella home dell'applicazione	Dalla home di ciascun utente, effettuando il logout si viene indirizzati nella home dell'applicazione	OK
6	Refresh della pagina	Per il Dipendente, quando viene refreshata la pagina, l'applicazione mantiene il Dipendente	Non sempre questo accade	
7	Ritorno alla home da errore	Dalla pagina di errore inserimento dati per login, cliccando su home si viene indirizzati nella home	Dalla pagina di errore inserimento dati per login, cliccando su home si viene indirizzati nella home	OK
8	Clicca su "Inserisci Risultato di un Tampone" dalla home del Dipendente	Si viene indirizzati in una pagina dove inserire il risultato di un tampone	Si viene indirizzati in una pagina dove inserire il risultato di un tampone	OK
9	Clicca su "Logout" dalla home del Dipendente	Si viene indirizzati nella home dell'applicazione	Si viene indirizzati nella home dell'applicazione	OK

10	Id_Tampone: corretto DataInserimento > DataPrenotazione e formato corretto Esito: positivo Paziente attualmente positivo	Si viene indirizzati nella pagina “Risultato inserito e tampone prenotato tra 7 giorni”	Si viene indirizzati nella pagina “Risultato inserito e tampone prenotato tra 7 giorni”	OK
11	Id_Tampone: corretto DataInserimento > DataPrenotazione e formato corretto Esito: negativo Paziente attualmente positivo	Si viene indirizzati nella pagina “Risultato inserito e tampone prenotato tra 3 giorni” ed inoltre viene cambiato lo stato del paziente in “transizione”	Si viene indirizzati nella pagina “Risultato inserito e tampone prenotato tra 3 giorni” ed inoltre viene cambiato lo stato del paziente in “transizione”	OK
12	Id_Tampone: corretto DataInserimento > DataPrenotazione e formato corretto Esito: negativo Paziente attualmente in transizione	Si viene indirizzati nella pagina “Risultato inserito” ed inoltre viene cambiato lo stato del paziente in “negativo” e inQuarantena=0	Si viene indirizzati nella pagina “Risultato inserito” ed inoltre viene cambiato lo stato del paziente in “negativo” e inQuarantena=0	OK
13	Id_Tampone: corretto DataInserimento > DataPrenotazione e formato corretto Esito: positivo Paziente attualmente in transizione	Si viene indirizzati nella pagina “Risultato inserito e tampone prenotato tra 7 giorni ed inoltre viene cambiato lo stato del paziente in “positivo”	Si viene indirizzati nella pagina “Risultato inserito e tampone prenotato tra 7 giorni ed inoltre viene cambiato lo stato del paziente in “positivo”	OK
14	Id_Tampone: corretto DataInserimento > DataPrenotazione e formato corretto Esito: positivo Paziente attualmente negativo	Si viene indirizzati nella pagina “Risultato inserito e tampone prenotato tra 7 giorni ed inoltre viene cambiato lo stato del paziente in “positivo”	Si viene indirizzati nella pagina “Risultato inserito e tampone prenotato tra 7 giorni ed inoltre viene cambiato lo stato del paziente in “positivo”	OK
15	Id_Tampone: corretto DataInserimento > DataPrenotazione e formato corretto Esito: negativo Paziente attualmente negativo	Si viene indirizzati nella pagina “Risultato inserito” ed inoltre viene cambiato inQuarantena=0	Si viene indirizzati nella pagina “Risultato inserito” ed inoltre viene cambiato inQuarantena=0	OK
16	Id_Tampone: non presente DataInserimento > DataPrenotazione e formato corretto Esito: corretto	Si viene indirizzati nella pagina “Tampone non trovato”	Si viene indirizzati nella pagina “Tampone non trovato”	OK
17	Id_Tampone: corretto DataInserimento < DataPrenotazione e formato corretto Esito: corretto	Non mi deve far inserire il risultato per la data	Si viene indirizzati in una delle pagine “Risultato Inserito”, “Risultato Inserito e Tampone Prenotato” o “Tampone non trovato”	

18	Id_Tampone: corretto DataInserimento > DataPrenotazione e formato corretto Esito: scorretto	Mi ritorna un messaggio di errore	Mi ritorna un messaggio di errore	OK
19	Si prova ad inserire il risultato di un tampone il cui id si riferisce ad un paziente che appartiene ad un ASL diversa da quella del Dipendente	Non mi deve far inserire il risultato	Si viene indirizzati in una delle pagine “Risultato Inserito”, “Risultato Inserito e Tampone Prenotato” o “Tampone non trovato”	
20	Input tutti scorretti	Si viene indirizzati nella pagina “Tampone non trovato”	Si viene indirizzati nella pagina “Tampone non trovato”	OK
21	Si prova ad inserire il risultato di un tampone il cui id ha già un risultato presente nel database	Non mi deve far inserire il risultato	Si viene indirizzati in una delle pagine “Risultato Inserito”, “Risultato Inserito e Tampone Prenotato” o “Tampone non trovato” e il risultato viene sovrascritto	
22	Dalla pagina “risultato inserito e tampone prenotato” clicco home dipendente	Si viene indirizzati nella home del dipendente e si può inserire un altro risultato	Si viene indirizzati nella home del dipendente e si può inserire un altro risultato	OK
23	Dalla pagina “risultato inserito e tampone prenotato” clicco logout	Si viene indirizzati nella home dell’applicazione	Si viene indirizzati nella home dell’applicazione	OK
24	Dalla pagina “risultato inserito” clicco home dipendente	Si viene indirizzati nella home del dipendente e si può inserire un altro risultato	Si viene indirizzati nella home del dipendente e si può inserire un altro risultato	OK
25	Dalla pagina “risultato inserito” clicco logout	Si viene indirizzati nella home dell’applicazione	Si viene indirizzati nella home dell’applicazione	OK
26	Dalla pagina “tampone non trovato” clicco home dipendente	Si viene indirizzati nella home del dipendente e si può inserire un altro risultato	Si viene indirizzati nella home del dipendente e si può inserire un altro risultato	OK
27	Dalla pagina “tampone non trovato” clicco logout	Si viene indirizzati nella home dell’applicazione	Si viene indirizzati nella home dell’applicazione	OK
28	Data Inserimento non valida in “Inserisci Risultato Tampone”	Mi ritorna un messaggio di errore	Mi apre una pagina di errore del formato Data	
29	Tampone non intero in “Inserisci Risultato Tampone”	Mi ritorna un messaggio di errore	Mi apre una pagina di errore del formato Integer	
30	Dopo aver visualizzato il messaggio d’errore, si prova ad inserire un risultato i cui input sono corretti	Mi inserisce il risultato	Mi inserisce il risultato	OK

31	Dopo aver visualizzato il messaggio d'errore, si prova ad inserire un risultato i cui input sono scorretti	Mi ritorna un messaggio di errore	Mi ritorna un messaggio di errore	OK
----	--	-----------------------------------	-----------------------------------	----

Alcuni casi di test hanno evidenziato degli errori opportunamente corretti, quali la possibilità di inserire un risultato di un tampone, per un dipendente, ad un paziente che appartiene ad un'altra ASL, oppure la possibilità di inserire il risultato per un tampone che ha già un risultato nel database, o infine la possibilità di inserire un risultato per un tampone con una data precedente alla sua prenotazione. Tutti questi errori sono stati raggruppati in un "if" in modo che nella pagina html di "inserisci il risultato", mi apparisse un messaggio d'errore in presenza di uno o più errori precedentemente evidenziati ed invitasse il dipendente a reinserire gli input.