

UNIVERSITÀ DEGLI STUDI DI NAPOLI PARTHENOPE
SCUOLA INTERDIPARTIMENTALE DELLE SCIENZE, DELL'INGEGNERIA E DELLA SALUTE
INFORMATICA
CORSO DI RETI DI CALCOLATORI E LABORATORIO DI RETI DI CALCOLATORI



Vanilla Green Pass

Proponenti:	
Calcopietro Francesco	0124002090
Caruso Denny	0124002062

Data di Consegna:
//2022

Anno Accademico:
2021 – 2022

Categoria:
Green Pass

Questa pagina è stata lasciata bianca di proposito

Questa pagina è stata lasciata bianca di proposito

Indice

1 – Descrizione del progetto.....	1
2 – Descrizione e schemi dell'architettura.....	?
3 – Descrizione e schemi del protocollo applicazione.....	?
4 – Dettagli implementativi dei client.....	?
5 – Dettagli implementativi dei server.....	?
6 - Manuale utente.....	?
6.1 – Istruzioni per la compilazione.....	?
6.2 – Istruzioni per l'esecuzione.....	?
7 – Sviluppi futuri.....	?

Elenco delle figure

Figura x.y: Diagramma Architettura di Rete Vanilla Green Pass.....?
Figura x.y: Esecuzione di ServerV.....?
Figura x.y: Esecuzione di ServerG.....?
Figura x.y: Esecuzione di CentroVaccinale.....?
Figura x.y: Esecuzione di ClientCitizen.....?
Figura x.y: Esecuzione di ClientS.....?
Figura x.y: Esecuzione di ClientT.....?

Elenco delle tabelle

Bibliografia

GaPiL – Guida alla Programmazione in Linux, Simone Piccardi

Sitografia

<https://www.un.org/en/file/45419>

<https://github.com/dennewbie/VanillaGreenPass>

<https://www.dgc.gov.it/web/>

<https://gapil.gnulinix.it/>

Note Aggiuntive

- Per informazioni aggiornate e complete in merito all'attuale regolamentazione del Green Pass si faccia riferimento al terzo link riportato nella sitografia di questo documento.
- La repository GitHub verrà resa pubblica solo a partire da Gennaio 2023.

Descrizione del progetto

1 - Descrizione del progetto

La Certificazione verde “COVID-19 - *EU digital COVID certificate*” (più semplicemente detta Green Pass) nasce su proposta della Commissione europea per agevolare la libera circolazione in sicurezza dei cittadini nell’Unione europea durante la pandemia di COVID-19.

È una certificazione digitale e stampabile (cartacea), che contiene un codice a barre bidimensionale (QR Code) e un sigillo elettronico qualificato. In Italia, viene emessa soltanto attraverso la Piattaforma nazionale DGC del Ministero della Salute. Il Green Pass facilita i viaggi in Europa e nel mondo. Nel nostro Paese rende più sicuri i cittadini al lavoro, a scuola e in molte attività quotidiane. Il Green Pass presenta le seguenti caratteristiche:

- formato digitale e/o cartaceo
- verificabile con QR code
- gratis per tutti
- in italiano e in inglese, più francese o tedesco
- sicura e protetta
- valida in tutta l’UE e altri Paesi non-UE

Il Green Pass attesta una delle seguenti condizioni:

- aver fatto la vaccinazione anti COVID-19 (in Italia viene emessa dopo ogni dose di vaccino);
- essere negativi al test antigenico rapido nelle ultime quarantotto ore o al test molecolare nelle ultime settantadue ore;
- essere guariti dal COVID-19 da non più di sei mesi.

Il Regolamento europeo sulla Certificazione è entrato in vigore il 1 luglio 2021 in tutti i Paesi dell’Unione e avrà durata di un anno. L’Italia ha anticipato l’emissione della Certificazione verde COVID-19 al 17 giugno 2021 e ne ha esteso progressivamente l’utilizzo sul territorio nazionale. Per maggiori informazioni consultare il terzo link riportato all’interno della sitografia di questo documento.

Ora che è stato introdotto cosa sia il Green Pass e come viene utilizzato nella vita reale, si propone l’esemplificazione a scopo didattico dello stesso. Al fine di non creare confusione si farà riferimento al Green Pass semplificato realizzato nell’ambito di questo progetto come “Vanilla Green Pass” ovvero un Green Pass senza particolari personalizzazioni e semplificato nel suo funzionamento. Un Vanilla Green Pass è valido a partire dal momento del suo rilascio dopo la vaccinazione fino al giorno in cui passeranno sei mesi rispetto al primo giorno del mese nel quale è stato rilasciato. Ciò vuol dire che nel momento in cui un Vanilla Green Pass è rilasciato nel giorno 15/01/2022, verrà preso in considerazione il primo giorno del mese di gennaio (ovvero 01/01/2022) e sommati sei mesi (01/06/2022). A partire dalla data risultante sarà possibile effettuare una nuova dose di vaccino. Si precisa che con Vanilla Green Pass non si fa distinzione tra quello ottenuto dopo la prima dose, la seconda dose o la dose booster. Inoltre, un Vanilla Green Pass è ottenibile solo mediante vaccinazione.

In base a quanto detto finora un Vanilla Green Pass rilasciato l’ultimo giorno del mese di gennaio, avrà la stessa data limite di scadenza di un Vanilla Green Pass rilasciato nei primi giorni del mese. Queste sono le prime semplificazioni più importanti apportate a un Vanilla Green Pass rispetto al Green Pass reale. In base a ciò, si noti che non sempre la validità del Vanilla Green Pass risulterà essere di sei mesi, bensì oscillante tra i cinque ed i sei mesi a seconda se il vaccino è stato inoculato rispettivamente a fine mese oppure ad inizio mese. Tale semplificazione è stata apportata così da velocizzare e facilitare il

calcolo del periodo di validità di un Vanilla Green Pass evitando inoltre l'ottenimento di date il cui giorno non è realmente esistente; si pensi al 29/02/2022 o al 31/04/2022.

La traccia del progetto richiede di progettare e implementare un servizio di gestione dei Green Pass. La traccia consultabile dal file "tracciaProgetto.pdf" è stata ampliata ed arricchita. Di seguito si riportano le specifiche risultanti di Vanilla Green Pass in quanto sistema software.

- Un utente, una volta effettuata la vaccinazione, tramite un client si collega ad un centro vaccinale e comunica il codice della propria tessera sanitaria. Si suppone che ogni utente, che chiameremo da qui in poi cittadino, abbia un codice di tessera sanitaria univoco, permanente e senza una scadenza associata alla relativa tessera sanitaria. Si suppone che tale codice non segua un particolare formato, ma abbia un numero minimo di caratteri pari a venti. I caratteri possono essere lettere, numeri, simboli.
- Il centro vaccinale comunica al ServerV il codice ricevuto dal client ed il periodo di validità del Vanilla Green Pass. Si suppone che il periodo di validità venga indicato mediante una data che corrisponderà alla data a partire dalla quale sarà possibile essere sottoposti all'inoculazione di una nuova dose di vaccino. Di conseguenza, rappresenterà anche il periodo di validità del Vanilla Green Pass associato a quel codice di tessera sanitaria. Si suppone che il formato della data sia del tipo DD-MM-YYYY dove DD rappresenta il giorno espresso da un intero non negativo che va da "01" a "31" in accordo con i giorni dei singoli mesi, MM rappresenta il mese espresso da un intero non negativo che va da "01" a "12" e YYYY rappresenta l'anno espresso in quattro cifre. Il centro vaccinale oltre a comunicare col ServerV, dialoga anche con il client del cittadino che sta cercando di sottoporsi all'inoculazione del vaccino. Quest'ultimo attenderà l'esito della richiesta di vaccinazione. Qualora l'esito sarà positivo, allora il vaccino può essere inoculato, altrimenti non sarà possibile inoculare il vaccino.
- Un ClientS, per verificare se un Vanilla Green Pass è valido, invia il codice di una tessera sanitaria al ServerG il quale richiede al ServerV il controllo della validità. Il ClientS può essere considerato come quello installato presso il terminale di un ristorante, il cui proprietario o l'addetto apposito utilizza per verificare i Vanilla Green Pass dei clienti all'ingresso. Si suppone che l'esito della verifica di un Vanilla Green Pass dipenda da due fattori: la data di scadenza del Vanilla Green Pass (o periodo di validità) in base alla definizione che ne è stata data nei paragrafi precedenti e uno stato che può assumere due valori, attivo oppure non attivo (rispettivamente uno o zero). Questo vuol dire che un Vanilla Green Pass sebbene sia ancora valido in termini di date perché non è ancora scaduto, può risultare non valido in quanto il cittadino possessore del codice di tessera sanitaria associata a quel Vanilla Green Pass è risultato positivo al (o alternativamente è stato contagiato) COVID-19.

In maniera del tutto analoga ad un Vanilla Green Pass può essere associato uno stato valido, sebbene sia scaduto in base al periodo di validità ad esso associato. Ciò può accadere nel momento in cui si supera il giorno di scadenza previsto. Però non rappresenta comunque un problema. Infatti, un Vanilla Green Pass per essere considerato valido deve presentare entrambe le caratteristiche. Tali caratteristiche sono adeguatamente controllate dal ServerV incaricato.

- Un ClientT, inoltre, può invalidare o ripristinare la validità di un Vanilla Green Pass comunicando al ServerG il contagio o la guarigione di una persona attraverso il codice della tessera sanitaria. Il ServerG provvederà a mettersi in comunicazione con il ServerV, il quale a sua volta aggiornerà lo stato del corrispettivo Vanilla Green Pass se esistente. Da quest'ultimo aspetto si intende che così come tante altre operazioni effettuate a livello implementativo, anche la richiesta di invalidare o ripristinare la validità di una Vanilla Green Pass può non andare a buon fine. Per esempio, come detto poc'anzi, se non esiste un Vanilla Green Pass associato al codice di tessera sanitaria inviato dal ClientT. Per il ClientS, il controllo della validità di un Vanilla Green Pass può altresì fallire.

Inoltre, si mettono in luce i seguenti aspetti del sistema software realizzato.

- Si consideri la circostanza secondo la quale un cittadino possa tentare di richiedere nuovamente il vaccino dopo qualche giorno dalla precedente inoculazione, o comunque prima che passi il tempo stabilito in base alla regolamentazione vigente. Il sistema software da realizzare deve prevedere un controllo tale per cui, in tali circostanze non venga inoculato al cittadino un'ulteriore dose.
- Si consideri la circostanza in cui possano esservi errori di input da parte dell'utente. In tali circostanze il sistema software deve riconoscere, individuare e segnalare l'errore evitando di ritrovarsi uno stato inconsistente.
- Si consideri la circostanza in cui possano esservi indirizzi IP di configurazione non validi. In tali circostanze il sistema software deve riconoscere, individuare e segnalare l'errore evitando di ritrovarsi uno stato inconsistente. In maniera del tutto analoga il sistema software in tutto il suo ciclo di vita deve intercettare e riconoscere stati d'errore generati da chiamate a funzioni o altro.
- La persistenza dei dati relativi ai Vanilla Green Pass, ovvero codice di tessera sanitaria, periodo di validità e stato di validità, sono assicurati mediante il salvataggio degli stessi sul file system, più in particolare su un supporto di memorizzazione di massa in un apposito file "serverV.dat", situato nella sottocartella "data". I file di configurazione delle varie entità del sistema software che prenderemo in considerazione in maniera più approfondita nelle prossime sezioni, sono nel formato ".conf" e sono memorizzati sul supporto di memorizzazione di massa nella sottocartella "conf". Infine, i sorgenti sono memorizzati nella sottocartella "src", mentre tutti gli eseguibili da testare nella sottocartella "bin". Questa documentazione è invece salvata nella cartella "doc".
- Le entità che verranno individuate sono eseguibili anche su macchine differenti, andando a modificare opportunamente i relativi file di configurazione.
- Non si gestisce la risoluzione dei nomi a dominio.
- Non si gestisce la risoluzione degli indirizzi IPv6, né la doppia gestione di indirizzi sia IPv4 che IPv6, ma soltanto IPv4.
- Non si gestisce l'implementazione della persistenza dei dati mediante database.
- Non si gestisce l'implementazione di tecniche di crittografia per i dati scambiati via rete, né di serializzazione dei dati stessi all'interno dei dispositivi di memorizzazione di massa.

Descrizione e schemi dell'architettura

2 - Descrizione e schemi dell'architettura

Vanilla Green Pass sfrutta un'architettura di rete di tipo Client/Server multilivello in cui i vari client e server possono essere in esecuzione sulla stessa macchina oppure su macchine differenti. Un'architettura Client/Server è un'architettura all'interno della quale vi sono una o più entità Client che richiedono un servizio e una o più entità Server che offrono un servizio. Se si tratta di un'architettura Client/Server multilivello, questo implica che vi possano essere una o più entità intermedie costituenti dei livelli per l'appunto tra il Server che offre un servizio e il Client che lo richiede. Nel caso di Vanilla Green Pass si individuano tre livelli.

In particolare, un primo livello è quello al quale accedono i client che a vario titolo possono segnalare la propria volontà di effettuare il vaccino per ricevere il Vanilla Green Pass e attendere l'esito della richiesta, oppure possono verificare la validità di un Vanilla Green Pass associato a un codice di tessera sanitaria fornito in input, o ancora invalidare o riattivare un Vanilla Green Pass associato a un codice di tessera sanitaria fornito in input. Un secondo livello dell'architettura è rappresentato da quelli che sono il ServerG e il CentroVaccinale. Il primo dei due svolge una funzione da "gateway" da e verso il ServerV, mettendo in comunicazione il ClientT ed il ClientS con il ServerV. Invece, il CentroVaccinale è come se svolgesse la stessa funzione del ServerG ma questa volta mettendo in comunicazione il client del cittadino che richiede di vaccinarsi col ServerV. Il terzo livello, infine, è dato dal nodo centrale e dominante dell'architettura ovvero il ServerV stesso: il "cuore" di Vanilla Green Pass.

A questo punto si può stabilire che vengono individuate le seguenti entità all'interno di Vanilla Green Pass:

- Client del cittadino che richiede di vaccinarsi (da ora in poi verrà fatto riferimento a quest'ultimo col nome di ClientCitizen);
- ClientS;
- ClientT;
- CentroVaccinale;
- ServerG;
- ServerV.

Un ClientCitizen comunica solo con un CentroVaccinale. Un CentroVaccinale, invece, comunica sia con più ClientCitizen che con il ServerV. Si noti come si faccia riferimento al ServerV con l'articolo determinativo "il" e non con quello indeterminativo "un" ad indicare che in Vanilla Green Pass si suppone l'esistenza di un solo ServerV, e grazie a questa caratteristica sarà poi possibile effettuare un controllo di tipo centralizzato sui vari Vanilla Green Pass memorizzati mediante file system. Se questo comporta delle semplificazioni da un lato, è anche vero che lo rende un punto debole dell'architettura: un single point of failure.

Un ClientS comunica solo con un ServerG. Un ClientT comunica solo con un ServerG. Un ServerG, invece, comunica sia con più ClientS, sia con più ClientT, che con il ServerV. Si noti anche in questo caso la scelta degli articoli determinativi e indeterminativi. Di conseguenza è possibile avere più istanze di un ServerG in esecuzione, più istanze di un CentroVaccinale in esecuzione, così come di ClientCitizen, ClientS e ClientT, ma non più istanze del ServerV in esecuzione.

Il ServerV comunica sia con più CentriVaccinali, sia con più ServerG. Quindi l'architettura a tre livelli nasce esattamente in questo punto, ovvero il ServerV comunica mediante il ServerG con il ClientS e con il ClientT e mediante il CentroVaccinale con il ClientCitizen.

Di seguito si allega uno schema riassuntivo dell'architettura di rete di Vanilla Green Pass illustrato mediante diagramma delle componenti.

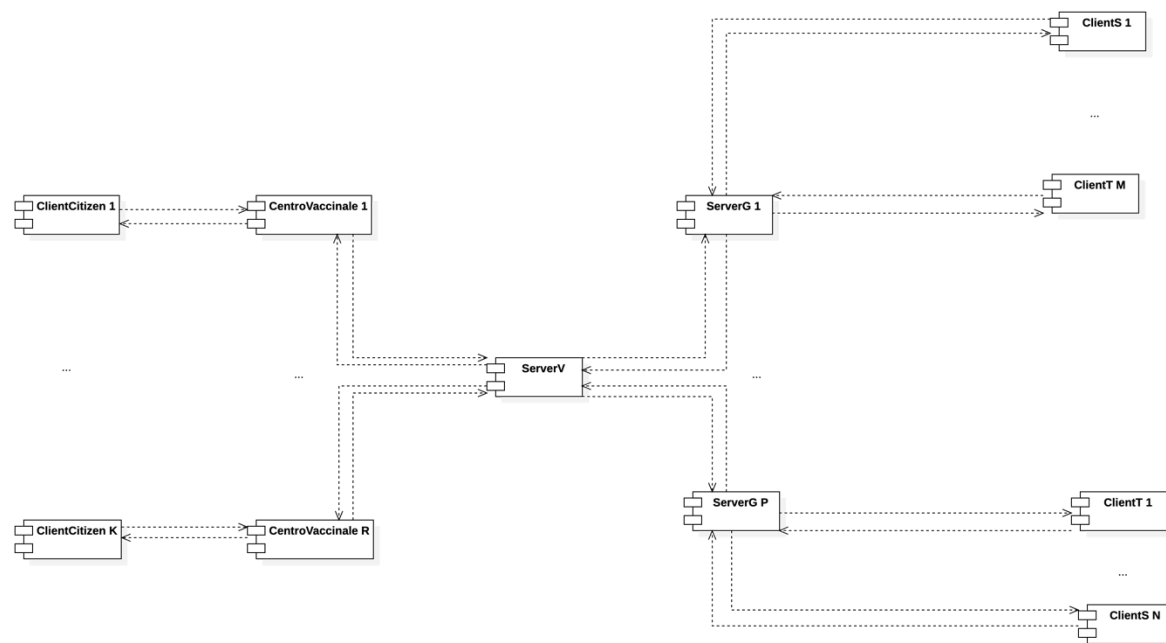


Figura x.y: Diagramma Architettura di Rete Vanilla Green Pass

Un ClientCitizen, un ClientS e un ClientT svolgono soltanto un ruolo di client: ovvero richiedono un servizio sulla rete (tipicamente a un server). Invece, il ServerV svolge solo un ruolo di server: ovvero offre un servizio sulla rete (tipicamente ai client). Infine, un ServerG e un CentroVaccinale svolgono sia un ruolo da server che da client. Questi ultimi svolgono un ruolo da client quando richiedono un servizio al ServerV, mentre svolgono un ruolo da server quando offrono un servizio ai client coi quali hanno stabilito una connessione. ServerG e CentroVaccinale possono anche essere visti come dei gateway (o addirittura degli intermediari tra il ServerV e il client di turno) verso i quali i client indirizzano le proprie richieste e dai quali gli stessi client ricevono delle risposte alle loro richieste.

C'è da sottolineare però che ServerG e CentroVaccinale offrono servizi differenti e non sono la stessa entità a livello di rete. Un ServerG offre un servizio ai ClientS e ai ClientT (rispettivamente verifica validità Vanilla Green Pass e gestione rinnovo/invalidazione Vanilla Green Pass), mentre il CentroVaccinale offre un servizio ai ClientCitizen (richiesta di vaccinazione e ottenimento periodo di validità).

Si noti come nello schema si utilizzino alcune variabili quali: K, M, N, P, R. Tali lettere stanno a significare che possono esservi rispettivamente più istanze di ClientCitizen, più istanze di ClientT, più istanze di ClientS, più istanze di ServerG, più istanze di CentroVaccinale. Inoltre, non necessariamente i valori numerici strettamente positivi assegnati a tali variabili sono uguali tra di loro.

Descrizione e schemi del protocollo applicazione

3 - Descrizione e schemi del protocollo applicazione

Temp text

Dettagli implementativi dei client

4 - Dettagli implementativi dei client

Temp text

Dettagli implementativi dei server

5 - Dettagli implementativi dei server

Temp text

Manuale utente

6 - Manuale utente

6.1 – Istruzioni per la compilazione

Per quanto riguarda la fase di compilazione, è reso disponibile un makefile. Un makefile consiste in linee di testo che definiscono un file (o un gruppo di file) oppure il nome di una regola dipendente dal gruppo di file. I file generati sono contrassegnati come i loro file sorgenti, mentre i file sorgenti sono contrassegnati a seconda dei file inclusi internamente. Dopo che ogni dipendenza è dichiarata, può seguire una serie di linee indentate (da tabulazioni) che definiscono come trasformare i file di ingresso nei file d'uscita, se il primo è stato modificato più di recente rispetto al secondo. Nel caso in cui tali definizioni sono presenti, queste sono riferite a dei script di compilazione e sono passate alla shell per generare i file target.

I singoli eseguibili delle varie entità analizzate nelle sezioni precedenti sono già disponibili. Si precisa che sono stati generati su una macchina con processore Intel e sistema operativo macOS Monterey 12.1. Alternativamente è possibile ricompilare tutti i sorgenti ed effettuare il linking delle relative librerie in maniera facile e veloce tramite il makefile realizzato. Per fare ciò, basta recarsi da shell (o terminale) in “../VanillaGreenPass/src/”, digitare il comando “make” e premere il tasto “Invio” (o Enter). A questo punto verrà avviata l’utility make che provvederà ad effettuare le operazioni dette poc’anzi in base alla struttura del makefile realizzato. Il makefile è consultabile in “../VanillaGreenPass/src/Makefile”. Qualora per qualche motivo il makefile non dovesse funzionare, si consiglia di utilizzare i consueti comandi per la compilazione e il linking da terminale per ogni entità individuata precedentemente.

Infine, non sono richieste particolari librerie esterne. In ogni caso si allegano di seguito le librerie utilizzate:

- stdio.h;
- stdlib.h;
- unistd.h;
- errno.h;
- string.h;
- sys/types.h;
- sys/socket.h;
- arpa/inet.h;
- time.h;
- ctype.h;
- pthread.h;
- stdarg.h;

6.2 – Istruzioni per l’esecuzione

Si prendono ora in considerazione le istruzioni per l’avvio e l’esecuzione di ognuna delle entità del sistema software realizzato. Ogni entità del sistema (tranne che per il ServerV) dispone di un file di configurazione all’interno del quale sono riportate due informazioni: IP e porta. Tali informazioni sono l’IP e la porta dell’entità di rete alla quale è necessario collegarsi per adempiere le proprie funzioni. Di conseguenza ogni entità che dispone di tale file “.conf” vuol dire che ha bisogno di mettersi in collegamento, comunicare con un’altra entità di rete al fine di un corretto funzionamento. Il file di configurazione prende il nome dall’entità che ha la necessità di avere tale file. La struttura generica di un file di configurazione è la seguente:

```
<IP dell’entità di rete alla quale collegarsi>  
<Porta dell’entità di rete alla quale collegarsi>
```

Il file ha anche una terza riga vuota di default che rappresenta l’EOF (end-of-file). Le informazioni sono separate su righe differenti grazie al carattere “new-line”.

Per esempio, il “clientCitizen” avrà nel proprio file “clientCitizen.conf” sulla prima riga l’IP del centro vaccinale al quale collegarsi e sulla seconda riga la porta del centro vaccinale al quale collegarsi (o alternativamente sulla quale il centro vaccinale è in ascolto per un clientCitizen). Quindi fattore essenziale per il suo avvio è impostare tali parametri. Essendovi la possibilità di avere più di un centro vaccinale, si sceglie uno di essi e si impostano i suoi dati in base a quanto detto prima all’interno del file di configurazione. In maniera del tutto analoga all’interno del file di configurazione del centro vaccinale (“centroVaccinale.conf”) e del ServerG (“serverG.conf”), si ritroveranno IP e porta del ServerV al quale collegarsi; mentre all’interno del file di configurazione del ClientT (“clientT.conf”) e del ClientS (“clientS.conf”), si ritroveranno IP e porta del ServerG al quale collegarsi. Per cambiare tali parametri che attualmente sono impostati nella configurazione

```
<IP loopback>  
<porta progressiva>
```

basta andare ad aprire tali file con un qualunque editor di testo e apportare le modifiche opportune. Nella configurazione rilascio il ServerV si suppone sulla porta 20000, il ServerG sulla porta 30000, il CentroVaccinale sulla porta 10000. In base a queste ultime informazioni i file di configurazione .conf sono stati compilati di conseguenza.

Si prendono ora in considerazione gli argomenti di input passati da terminale per ogni entità realizzata seguenti la stringa “./entitàSoftwareDaAvviare”. Si precisa che le virgolette non sono necessarie, denotano soltanto il testo da inserire via terminale prima degli argomenti successivi.

- ClientCitizen e ClientS prevedono un argomento di input da linea di comando che consiste nel codice della tessera sanitaria. Tale codice deve rispettare il formato predefinito ipotizzato, ovvero formato da venti caratteri (lettere, numeri, simboli). In particolare, per ClientCitizen, tale argomento di input rappresenterà il codice della tessera sanitaria del cittadino che sta richiedendo una nuova dose di vaccino; mentre per ClientS rappresenterà il codice della tessera sanitaria associato a un Vanilla Green Pass che un ipotetico esercente (o altra entità del mondo reale) sta richiedendo di verificare.

- ClientT prevede due argomenti di input da linea di comando che corrispondono rispettivamente al codice della tessera sanitaria e al nuovo stato da associare al Vanilla Green Pass collegato a quel codice di tessera sanitaria (se esistente). Anche in questo caso il codice di tessera sanitaria deve rispettare il formato predefinito ipotizzato, ovvero formato da venti caratteri (lettere, numeri, simboli). Invece lo stato deve essere pari a 0 oppure a 1, ad indicare rispettivamente la volontà di invalidare il Vanilla Green Pass associato al codice di tessera sanitaria passato come primo argomento oppure la volontà di riattivare il Vanilla Green Pass associato al codice di tessera sanitaria passato come primo argomento. Qualora un Vanilla Green Pass fosse già attivo e si richiede di riattivarlo, quest'operazione non produce stati d'errore così come l'operazione speculare riguardante l'invalidazione.
- CentroVaccinale, ServerG e ServerV prevedono un solo argomento di input, ovvero la porta sulla quale viene attivato il rispettivo servizio del centro vaccinale, del ServerG, del ServerV. Chiaramente i numeri di porta devono essere consistenti con i requisiti di un numero di porta per il sistema operativo. Si suppone inoltre, che si faccia uso solo di numeri di porta registrati o effimeri. Infine, per un corretto funzionamento i numeri di porta dati in input a queste entità software devono essere coerenti con quelli presenti all'interno dei file di configurazione precedentemente analizzati.

Si allegano qui di seguito alcune catture dello schermo relative all'esecuzione delle varie entità realizzate al fine di poter mostrare come avviarle correttamente e cosa si visualizza durante l'esecuzione. In ordine è possibile visionare l'esecuzione di un ServerV, di un ServerG, di un CentroVaccinale, di un ClientCitizen, di un ClientS, di un ClientT.

```

src — serverV 20000 — 98x27
dennycaruso@Norah-MacBook-Pro src % ./serverV 20000

```

Figura x.y: Esecuzione di ServerV

```

src — serverG 30000 — 98x27
Last login: Sun Jan 23 15:06:08 on ttys003
dennycaruso@Norah-MBP src % ./serverG 30000

```

Figura x.y: Esecuzione di ServerG

```

src — centroVaccinale 10000 — 98x27
dennycaruso@Norah-MacBook-Pro src % ./centroVaccinale 10000

```

Figura x.y: Esecuzione di CentroVaccinale


```

dennycaruso@Norah-MBP src % ./clientS 00000000000000000000
Verifica GreenPass
Numero tessera sanitaria: 00000000000000000000
... A breve verra' mostrato se il GreenPass inserito risulta essere valido...
... Verifica in corso ...
La tessera sanitaria immessa risulta essere associata a un GreenPass attualmente valido.
Arrivederci.
dennycaruso@Norah-MBP src % ./clientT 00000000000000000000 0
Aggiornamento Validita' GreenPass
Numero tessera sanitaria: 00000000000000000000
... A breve verra' inviato il nuovo stato di validita' del Green Pass...
... Aggiornamento in corso ...
L'aggiornamento del Green Pass associato alla tessera sanitaria 00000000000000000000, e' andato a
buon fine.
Arrivederci.
dennycaruso@Norah-MBP src % ./clientS 00000000000000000000
Verifica GreenPass
Numero tessera sanitaria: 00000000000000000000
... A breve verra' mostrato se il GreenPass inserito risulta essere valido...
... Verifica in corso ...
La tessera sanitaria immessa non risulta essere associata a un GreenPass attualmente valido.
Arrivederci.
dennycaruso@Norah-MBP src % █

```

Figura x.y: Esecuzione di ClientS

```
src — zsh — 98x27
dennycaruso@Norah-MBP src % ./clientT 00000000000000000000 0

Aggiornamento Validita' GreenPass
Numero tessera sanitaria: 00000000000000000000

... A breve verra' inviato il nuovo stato di validita' del Green Pass...

... Aggiornamento in corso ...

L'aggiornamento del Green Pass associato alla tessera sanitaria 00000000000000000000, e' andato a
buon fine.
Arrivederci.
dennycaruso@Norah-MBP src % ./clientT 00000000000000000000 1

Aggiornamento Validita' GreenPass
Numero tessera sanitaria: 00000000000000000000

... A breve verra' inviato il nuovo stato di validita' del Green Pass...

... Aggiornamento in corso ...

L'aggiornamento del Green Pass associato alla tessera sanitaria 00000000000000000000, e' andato a
buon fine.
Arrivederci.
dennycaruso@Norah-MBP src %
```

Figura x.y: Esecuzione di ClientT

Come si può notare ServerV, ServerG e CentroVaccinale non presentano output, a differenza di ClientCitizen, ClientS e ClientT. Inoltre, si rende evidente come l'interazione dell'utente è molto banale e facilitata in quanto consiste solo dell'avvio dell'entità di turno considerata.

Sviluppi futuri

7 - Sviluppi futuri

È chiaro che aggiungere peculiarità e dettagli in linguaggio naturale risulta essere molto semplice, un po' meno l'effettiva implementazione di alcuni di essi. Si riportano di seguito alcuni ulteriori aggiornamenti che sarebbe desiderabile implementare nel sistema software realizzato.

- Interfacciamento con l'utenza manageriale al fine di cambiare periodo di validità di un singolo Vanilla Green Pass, per modificare il formato di memorizzazione delle informazioni relative a un Vanilla Green Pass, per effettuare operazioni di ricerca e gestionali all'interno dell'insieme dei Vanilla Green Pass archiviati, gestione scorte vaccini per il centro vaccinale, etc.
- Sostituire il salvataggio dei dati da file memorizzato su file system a un database vero e proprio (relazionale, ad oggetti, etc). Ciò comporta una revisione delle interazioni finora previste, l'interazione e l'inclusione con librerie esterne come per esempio, una fra tante, SQLite.
- Rendere il controllo ServerV maggiormente decentralizzato con l'esecuzione di più istanze del ServerV, eliminando così la problematica del single point of failure (SPOF).
- Risoluzione dei nomi a dominio.
- Risoluzione degli indirizzi IPv6 e/o doppia gestione di indirizzi sia IPv4 che IPv6.
- Realizzazione di un'interfaccia grafica appropriata che sostituisca quella a linea di comando.
- Crittografia dei dati trasmessi da un'entità della rete all'altra e serializzazione secondo un formato proprietario specifico dei dati salvati in maniera persistente su supporto di memorizzazione di massa (file di configurazione, eventuali file temporanei, etc.).

