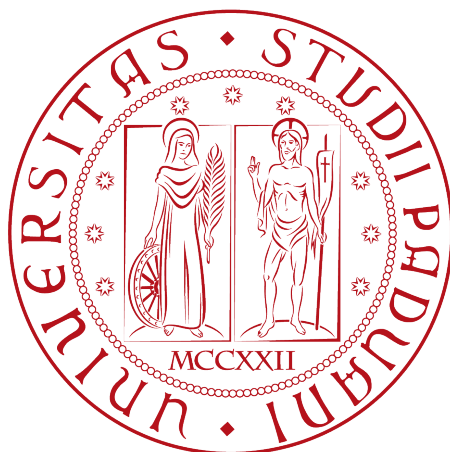


Università degli Studi di Padova

DIPARTIMENTO DI MATEMATICA "TULLIO LEVI-CIVITA"

CORSO DI LAUREA IN INFORMATICA



**Sviluppo soluzioni front-end in ambito
Identity and Access Management e Identity
Governance**

Tesi di laurea

Relatore

Prof. Ombretta Gaggi

Laureando

Dal Medico Simone

Sommario

Il presente documento descrive il lavoro svolto durante il periodo di stage, della durata di circa trecentoventi ore, dal laureando Simone Dal Medico presso l'azienda Monokee. In Conformità con gli obiettivi e la pianificazione dello stage, riportati nella prima parte del documento, le attività svolte hanno in primo luogo richiesto uno studio approfondito delle tecnologie utilizzate, in particolare [Javascript_g](#), il *framework* VueJs, il suo material design component *framework* Vuetify e VeeValidate per la validazione dei [Form_g](#). Lo stage richiedeva lo sviluppo di un *front-end* per la gestione, l'aggiunta e la visualizzazione di Pool e di Agent che compongono la [Self Sovereign Identity \(SSI\)_g](#). Nella tesi sono descritti i risultati delle attività di analisi dei requisiti, progettazione e codifica, con particolare riferimento alla struttura offerta dal *framework* utilizzato. Le attività dello stage hanno portato a conseguire tutti gli obiettivi obbligatori richiesti.

Dedicato a te.

“Fare un altro round quando pensi di non farcela è una cosa che può cambiare tutta la tua vita”

— Sylvester Stallone nel film Rocky

Ringraziamenti

Innanzitutto, vorrei esprimere la mia gratitudine alla Prof. Ombretta Gaggi, relatrice della mia tesi, per l'aiuto e il sostegno fornitomi durante la stesura del lavoro.

Desidero ringraziare con affetto i miei genitori per il sostegno, il grande aiuto e per essermi stati vicini in ogni momento durante gli anni di studio.

Ho desiderio di ringraziare poi i miei amici per tutti i bellissimi anni passati insieme e le mille avventure vissute.

Ringrazio tutti con tutto il mio cuore, se sono riuscito ad arrivare a questo traguardo lo devo anche a voi.

Padova, Dicembre 2022

Dal Medico Simone

Indice

1	Introduzione	1
1.1	L'azienda	1
1.2	L'idea	1
1.3	Organizzazione del testo	2
2	Descrizione dello stage	3
2.1	Contesto e oggetto del progetto	3
2.2	Obiettivi	9
3	Analisi dei requisiti	11
3.1	Attori	11
3.2	Casi d'uso	12
3.3	Tracciamento dei requisiti	47
4	Tecnologie e strumenti	51
4.1	Tecnologie	51
4.1.1	JavaScript	51
4.1.2	VueJS	51
4.1.3	Nuxtjs	51
4.1.4	Vuetify	51
4.1.5	Vee-validate	52
4.1.6	Vuei18n	52
4.1.7	Axios	52
4.2	Strumenti	52
4.2.1	Visual Studio Code	52
4.2.2	Figma	52
4.2.3	YouTrack Administration	52
4.2.4	GitHub	52
4.2.5	Microsoft Teams	53
5	Progettazione e codifica	55
5.1	Progettazione	55
5.1.1	Architettura di Vuejs	55
5.1.2	Architettura dell'applicazione sviluppata	56
5.2	Design Pattern utilizzati	57
5.3	Codifica	58
5.3.1	Layout	59
5.3.2	Components	59
5.4	Accessibilità	67

5.4.1	Lingua	67
5.4.2	Errori	68
5.4.3	Blocco dei Bottoni	68
5.4.4	Colori	69
5.4.5	Altri elementi di accessibilità	69
6	Conclusioni	71
6.1	Consuntivo finale	71
6.2	Raggiungimento degli obiettivi	71
6.3	Raggiungimento degli obiettivi	73
6.4	Conoscenze acquisite	74
	Glossario	75
	Acronimi	77
	Bibliografia	79

Elenco delle figure

2.1	Struttura Monokee	3
2.2	File configurazione <i>Pool</i>	5
2.3	File configurazione <i>Agent</i>	7
3.1	Attori Primari	11
3.2	Diagramma UC1 - Visualizzazione Lista <i>Pool</i>	12
3.3	Diagramma UC3 - Aggiungere nuova <i>Pool</i>	13
3.4	Diagramma UC3.3 - Personalizzazione <i>Agent</i>	14
3.5	Diagramma UC3.4 - Inserire Genesis Txn	15
3.6	Diagramma UC4 - Visualizzazione dati singola <i>Pool</i>	17
3.7	Diagramma UC4.6 - Modificare Personalizzazione <i>Agent</i>	19
3.8	Diagramma UC4.7 - Visualizzazione Tabella	21
3.9	Diagramma UC4.7.3 - Visualizzazione Dati singola riga	22
3.10	Diagramma UC5 - Visualizzazione Lista <i>Agent</i>	24
3.11	Diagramma UC7 - Aggiungere nuovo <i>Agent</i>	25
3.12	Diagramma UC7.1 - Selezionare opzione istanziare nuovo <i>Agent</i>	26
3.13	Diagramma UC7.1.1 - Inserire inbound transport	27
3.14	Diagramma UC7.1.5 - Inserire admin	28
3.15	Diagramma UC7.2 - Selezionare opzione URL <i>Agent</i>	32
3.16	Diagramma UC8 Parte 1 - Visualizzazione dati singolo <i>Agent</i>	34
3.17	Diagramma UC8.1 - Visualizzazione inbound transport	35
3.18	Diagramma UC8.2 - Modificare inbound transport	36
3.19	Diagramma UC8.9 - Visualizzazione admin	38
3.20	Diagramma UC8.10 - Modificare admin	39
3.21	Diagramma UC8 Parte 2 - Visualizzazione dati singolo <i>Agent</i>	42
5.1	Struttura progetto	56
5.2	Struttura MVC e MVVM	58
5.3	layout	59
5.4	Dialog Component	60
5.5	Pool List Component	61
5.6	Input Dialog Component	61
5.7	Add <i>Pool</i> Component - 1	62
5.8	Add <i>Pool</i> Component - 2	63
5.9	Pool - 1	63
5.10	Pool - 2	64
5.11	Agent List Component	65
5.12	Add <i>Agent</i> Component - 1	65

5.13 Add <i>Agent</i> Component - 2	66
5.14 Add <i>Agent</i> Component - 3	66
5.15 Agent - 1	67
5.16 Errore input	68
5.17 Bottone disattivato - attivato	68

Elenco delle tabelle

2.1 Tabella decrizione tag delle <i>Pool</i>	6
2.2 Tabella decrizione tag degli <i>Agent</i>	8
2.3 Tabella degli obiettivi	9
3.1 Tabella del tracciamento dei requisiti funzionali	48
3.2 Tabella del tracciamento dei requisiti di vincolo	49
5.1 Tabella rapporto contrasti	69
6.1 Tabella di confronto tra le ore previste e le ore effettive per ogni attività	72
6.2 Tabella degli obiettivi	73

Convenzioni tipografiche

Riguardo la stesura del testo, relativamente al documento sono state adottate le seguenti convenzioni tipografiche:

- * gli acronimi, le abbreviazioni e i termini ambigui o di uso non comune menzionati vengono definiti nel glossario, situato alla fine del presente documento;
- * per la prima occorrenza dei termini riportati nel glossario viene utilizzata la seguente nomenclatura: *parola_g*;
- * i termini in lingua straniera o facenti parti del gergo tecnico sono evidenziati con il carattere *corsivo*.

Capitolo 1

Introduzione

In questo capitolo viene introdotta l'azienda in cui si è svolto il periodo di stage. Si introduce anche il progetto scelto descrivendone l'idea alla base.

1.1 L'azienda

Monokee è una start-up nata solo negli ultimi anni e che, in poco tempo, si è fatta strada nel campo dell'identità digitale per le sue idee innovative e per la qualità del suo lavoro. Monokee lavora in modo nativo in un ecosistema aziendale eterogeneo, che include, tra gli altri, clienti industriali, bancari e sanitari. L'intento di Monokee è di creare una [Web App_g](#), denominata con lo stesso nome, con lo scopo di portare importanti innovazioni in ambito Identity and Access Management ([IAM_g](#)) e Identity Governance Administration ([IGA_g](#)). Sono entrato in contatto con l'azienda durante lo "StageIT 2022".

1.2 L'idea

Proprio nel campo delle identità digitali nasce il progetto Monokee. Infatti da molti anni il mercato IAM ha come unici veri competitori aziende americane che hanno tutta la volontà di mantenere le identità digitali centralizzate, sia in termini di governance ([IGA](#)) che di accesso ([AM_g](#)). Monokee vuole abbracciare il nuovo paradigma fornendo uno strumento utile per migrare da un modello centralizzato a uno decentralizzato. Per ogni dominio applicativo che gestiscono e proteggono, forniscono una funzionalità di spin-off di identità partendo da profili centralizzati forniti nei repository canonici e trasformandoli in credenziali verificabili. Le credenziali verificabili verranno utilizzate in termini di titolarità secondo la disciplina [IGA](#) e mappate nei protocolli canonici di autenticazione tramite un gateway SP. In aggiunta, molto spesso, le applicazioni per la gestione delle proprie identità digitali sono complicate e difficili da usare per delle persone che non sono esperte del settore. Monokee vuole rendere la [UX_g](#) più confortevole e *user-friendly* cercando di creare un [UI_g](#) semplice e intuitiva per facilitare la gestione e l'Amministrazione della propria identità digitale.

1.3 Organizzazione del testo

Nel **secondo capitolo** si entra maggiormente nei dettagli del progetto Monokee e del progetto di stage descrivendo gli obiettivi richiesti.

Nel **terzo capitolo** viene esposta l'analisi dei requisiti con i rispettivi casi d'uso analizzati.

Nel **quarto capitolo** vengono descritte nel dettaglio le tecnologie utilizzate

Nel **quinto capitolo** vengono descritte le diverse fasi di progettazione e implementazione.

Nel **sesto capitolo** viene riportato un consuntivo delle attività svolte e una valutazione dello stage con alcune riflessioni personali.

Capitolo 2

Descrizione dello stage

In questo capitolo viene presentato nel dettaglio il progetto Monokee insieme agli obiettivi attesi e alla pianificazione delle attività da svolgere.

2.1 Contesto e oggetto del progetto

Come anticipato nel capitolo precedente l'obiettivo di Monokee è quello di permettere ai cittadini che desiderino utilizzare strumenti distribuiti di gestione dell'identità di poterlo fare senza per questo rinunciare all'utilizzo di servizi centralizzati che prevedono una gestione delle identità classica e di poterla gestire ed amministrare nel modo più semplice e intuitivo possibile.

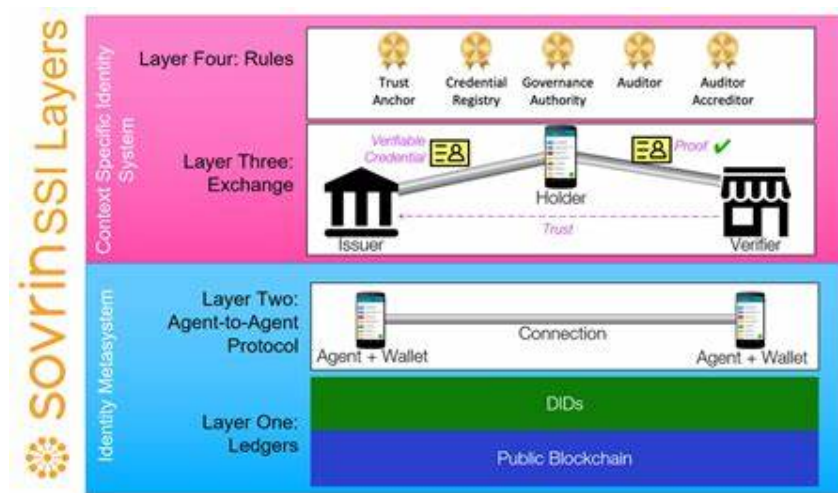


Figura 2.1: Struttura Monokee

La struttura del *framework*, come mostrato in Figura 2.1, è composta da quattro livelli chiamati *Layer* che ricordano molto la suddivisione in livelli del protocollo TCP/IP. Di seguito vengono descritti brevemente i 4 Layer partendo da quello più a basso livello e salendo fino all'ultimo.

Layer 1: Ledgers

Il Layer 1 è il livello più basso della gerarchia ed è costituito da una [BlockChain](#). Quest'ultima è formata da un'insieme di blocchi concatenati l'un l'altro. I nodi validatori, o miner, sono quei server che hanno autorità per scrivere e controllare che tutti facciano le operazioni in modo corretto. Una *Pool* è un insieme di nodi validatori che possono essere contattati per interagire con una particolare [BlockChain](#). Per identificare il set minimo di nodi associati con una *Pool* si utilizzano le transazioni o *Genesis txn* ovvero dei file [Json](#) di configurazione in cui ogni oggetto del file consiste in un nodo validatore che compone la *Pool* stessa. Le primissime txn scritte nella [BlockChain](#) dicono per quali sono i nodi validatori abilitati e tutto il resto della [BlockChain](#) viene derivato poi da queste.

Layer 2: Agent-Agent Protocol

Gli utenti non comunicano direttamente con la [BlockChain](#) ma utilizzano i cosiddetti *Agent*. Ogni *Agent* incapsula un *wallet* che contiene le informazioni dell'utente e le sue varie chiavi di sicurezza ed esso non comunica con nessuno non avendo accesso ad Internet. Attraverso l'*Agent* è possibile interagire con la [BlockChain](#) e con altri *Agent* per scambiare dati cifrati e autenticati in modo tale che possa leggerli solo l'*Agent* di quel particolare *wallet* a cui sono stati inviati. Per fare questo però è necessario che l'*Agent* sappia i parametri della specifica *Pool* prestabilita e l'indirizzo dell'*Agent* con cui si vuole scambiare i dati.

Layer 3: Exchange

Il livello 3 è un livello di protocollo che prende gli *Agent* e risponde alla domanda: che cosa ci posso fare adesso con questa rete di comunicazione? La risposta è scambiare credenziali verificabili, ovvero pacchetti dati ([Json](#)) crittograficamente validabili per essere integri e firmati. Quest'ultimi possono essere utilizzati per veicolare informazioni che nel caso di Monokee servono per l'identità digitale.

Layer 4: Rules

Il livello 4 è un livello di governance ovvero un livello legislativo e stabilisce le norme e le regole per potersi fidare di quello che viene fatto da ogni attore nei livelli inferiori. Stabilisce chi e cosa può emettere, che cosa significa validare, chi è autorizzato a fare e che cosa sulla [BlockChain](#).

In particolare l'oggetto dello stage si focalizza sui Layer 1 e 2. Esso consiste nell'analisi e nell'implementazione del frontend della sezione di [Self Sovereign Identity \(SSI\)](#) e, in particolare, delle pagine di amministrazione delle *Pool* e degli *Agent*. Di seguito vengono descritte brevemente le *Pool* e gli *Agent* e come sono strutturati.

Pool

La Figura 2.2 mostra una porzione del file [Json](#) di configurazione di una *Pool*.

```
{
  "reqSignature": {},
  "txn": {
    "data": {
      "data": {
        "alias": "Node1",
        "blskey":
"4N8aUNHSgjQVgkpm8nhNEfDf6txHznoYREg9kirmJrkivgL4oSEimFF6nsQ6M41QvhM2Z33nves5vf
Sn9n1UwNfJBYtWVnHYMATn76vLuL3zU88KyeAYcHfsih3He6UHcXDxcaecHVz6jhCYz1P2UZn2bD
VruL5wXpehgBfBaLKm3Ba",
        "blskey_pop":
"RahHYiCvoNCiPTTrVtP7nMC5eTYrsUA8WjXbdhNc8debh1agE9bGiJxWBXYNFbnJXoXhWFMvyqhq
hRoq737YQemH5ik9oL7R4NTTCz2LEZhkgLJzB3QRQqJyBNyv7acbdHrAT8nQ9UkLbaVL9NBpnWX
BTw4LEMePaSHEw66RzPNdAX1",
        "client_ip": "185.4.154.131",
        "client_port": 9702,
        "node_ip": "185.4.154.131",
        "node_port": 9701,
        "services": [
          "VALIDATOR"
        ]
      },
      "dest": "Gw6pDLhcBcoQesN72qfotTgFa7cbuqZpkX3Xo6pLhPhv"
    },
    "metadata": {
      "from": "Th7MpTaRZVRYnPiabds81Y"
    },
    "type": "0"
  },
  "txnMetadata": {
    "seqNo": 1,
    "txnId": "fea82e10e894419fe2bea7d96296a6d46f50f93f9eeda954ec461b2ed2950b62"
  },
  "ver": "1"
}
```

Figura 2.2: File configurazione *Pool*

In questa porzione di codice si può osservare in particolare la configurazione di uno dei nodi validatori che, insieme ad altri, costituisce una *Pool*. Ogni nodo perciò è rappresentato da un oggetto all'interno del file [Json](#) in cui vengono definiti specifici tag che di seguito andremo a descrivere brevemente.

Tag	Descrizione
alias	nome del nodo.
blskey	specifica la chiave BLS del nodo.
blskey-pop	specifica la prova di possesso per la chiave BLS.
client-ip	specifica l'ip del client.
client-port	specifica la porta del client.
node-ip	specifica l'ip del nodo.
node-port	specifica la porta del nodo.
services	specifica il tipo di servizio che ricopre il nodo (in questo caso è un nodo validatore).
dest	specifica il DID di Destinazione come stringa con codifica base58.
metadata	al suo interno vengono specificati come provengono i metadati dalla richihesta.
from	specifica il DID dell'autore della transazione come stringa con codifica base58.
type	un "enum" che identifica il tipo di transazione (sempre 0 per i nodi).
txnMetadata	al suo interno vengono definiti i Metadati allegati alla transazione.
seqNo	Un numero di sequenza univoco della transazione su Ledger.
txnId	specifica la chiave di Stato che può essere un indirizzo o dei dati descrittivi.
ver	specifica la Versione di transazione per poter evolvere il contenuto.

Tabella 2.1: Tabella descrizione tag delle *Pool*

Agent

La Figura 2.3 mostra un esempio di un file *aca-py* di istanziazione di uno specifico *Agent*.

```
aca-py start
--inbound-transport http 0.0.0.0 8103
--outbound-transport http
-e http://mkagent.test.ssi.monokee.com:8103
--genesis-url http://pool.test.indy.monokee.com:9000/genesis
--admin 0.0.0.0 8105
--admin-api-key F9KxjsYGtm7kcqgfwbf
--label \"MK Agent (role: Endorser) - DID: D198m2bpgz2Dr4sB8GXA4c\"
--image-url \"https://monokee.com/wp-content/uploads/2021/10/monkee-logo-
sito.png\"
--seed Endorser_MonoKee_Agent_jnudxxtlu
--wallet-name Endorser_MonoKee_Agent_Wallet
--wallet-key monokee_agent_wallet_key_secret_password
--wallet-type indy
--replace-public-did
--auto-provision
--auto-ping-connection
--auto-verify-presentation
--public-invites
```

Figura 2.3: File configurazione *Agent*

Il file *aca-py* soprastante mostra i dati di configurazione di uno specifico *Agent* in cui vengono definiti specifici tag che di seguito andremo a descrivere brevemente.

Tag	Descrizione
inbound-transport	Questo è il protocollo, l'indirizzo e la porta a cui la tua istanza ACA-py sarà raggiungibile per altre istanze ACA-py.
outbound-transport	Questo è il protocollo che la tua istanza ACA-py utilizzerà per comunicare con altre istanze.
e	Questo è l'URL a cui la tua istanza ACA-py sarà disponibile per essere raggiunta da altre istanze ACA-py. Questo URL verrà utilizzato per stabilire le connessioni.
genesis-url	Punta al server web della rete VON dove è possibile trovare il file genesis.

Tag	Descrizione
admin	Questo è l'indirizzo e la porta a cui puoi connetterti con la tua applicazione/controller.
admin-api-key	Questa è la chiave cifrata necessaria per permettere la connessione con la tua applicazione/controller.
label	Il nome assegnato a questa istanza ACA-py che verrà utilizzata nei messaggi.
image-url	Questo è un URL per caricare un'immagine.
seed	Definisce quale DID pubblico registrato nel ledger viene scelto.
wallet-name	specifica il nome del wallet collegato all' <i>Agent</i> .
wallet-key	specifica la chiave criptata del wallet collegato all' <i>Agent</i> .
wallet-type	specifica il tipo di wallet collegato all' <i>Agent</i> (in questo caso è un wallet di tipo indy).
replace-public-did	booleano che se impostato a true aggiornerà il DID nel <i>wallet</i> con il DID che corrisponde al seed fornito.
auto-provision	Booleano che se impostato a true esegue automaticamente il provisioning di un <i>wallet</i> nel caso in cui non esista.
auto-ping-connection	un booleano che se impostato a true invia automaticamente un messaggio ping dopo aver stabilito la connessione per contrassegnarla come "attiva".
auto-verify-presentation	Booleano che se impostato a true verifica automaticamente la presentazione dell' <i>Agent</i> .
public-invites	Un booleano che se impostato a true consente di utilizzare il DID pubblico registrato nel ledger inviando inviti e ricevendo richieste di connessione.

Tabella 2.2: Tabella descrizione tag degli *Agent*

Lo scopo principale dello stage dunque è di creare due *form* distinti per rendere più facile, intuitivo ed agevole la gestione e l'Amministrazione delle *Pool* e degli *Agent* senza dover andare a modificare direttamente i file di configurazione descritti precedentemente e, per far ciò, è necessario familiarizzare e apprendere tecnologie per lo sviluppo *front-end* di applicazioni *web* con i *frameworks* Vuejs e vuetify.

2.2 Obiettivi

La divisione del lavoro in periodi evidenzia una prima parte dedicata allo studio delle tecnologie necessarie allo sviluppo del progetto, in particolar modo a quelle per lo sviluppo del *frontend*. Superata la prima fase, si passa allo studio del progetto di per sé, con conseguente studio della sua architettura, delle tecnologie di *backend* utilizzate fino a quel momento e, analizzando le varie scelte di *frontend* adottate in quel momento. Successivamente si passa all'implementazione vera e propria e quindi alla creazione delle pagine richieste. Nella Tabella 2.3, vengono riportati gli obiettivi attesi, come definiti in fase di pianificazione dello stage. Il codice identificativo prevede la seguente sintassi:

O[Priorità] - [Id]

dove:

- * **O**: identifica che si tratta di un obiettivo.
- * **Priorità**: grado di priorità, può assumere i valori:
 - **A**: obbligatorio, necessario;
 - **B**: desiderabile, non strettamente necessario;
 - **C**: opzionale, non necessario.
- * **Id**: numero progressivo.

Obiettivo	Descrizione
OA01	Acquisizione competenze sulle tecnologie e gli strumenti utilizzati
OA02	Capacità di raggiungere gli obiettivi richiesti in autonomia seguendo il cronoprogramma
OA03	Portare a termine l'implementazione grafica della pagina di amministrazione delle <i>Pool</i>
OA04	Portare a termine l'implementazione grafica della pagina di amministrazione degli <i>Agent</i>
OOB01	realizzare le chiamate API per la visualizzazione e la modifica dei dati
OB02	analisi ed implementazione di dashboard grafiche per visualizzare telemetrie relative al funzionamento della piattaforma
OC01	Apportare un valore aggiunto al gruppo di lavoro durante le fasi di progettazione delle interfacce

Tabella 2.3: Tabella degli obiettivi

Capitolo 3

Analisi dei requisiti

In questo capitolo viene presentata l'analisi dei requisiti che evidenzia i diversi casi d'uso dell'applicativo da sviluppare. I diagrammi dei casi d'uso riportati nelle figure permettono una rappresentazione visiva delle possibili interazioni che l'utente può avere con l'applicativo.

3.1 Attori

La Figura 3.1 riporta il diagramma rappresentante gli attori primari. Si deve però specificare che essendo Monokee una [Web App](#) a pagamento, l'"Utente non Autenticato" si presuppone che abbia già pagato il servizio e che potrà solo effettuare il login e, tale funzionalità è già operativa e non fa parte delle task da implementare durante lo stage. Perciò, da ora in poi, si considereranno tutti diagrammi in cui si presuppone che l'utente sia già autenticato.

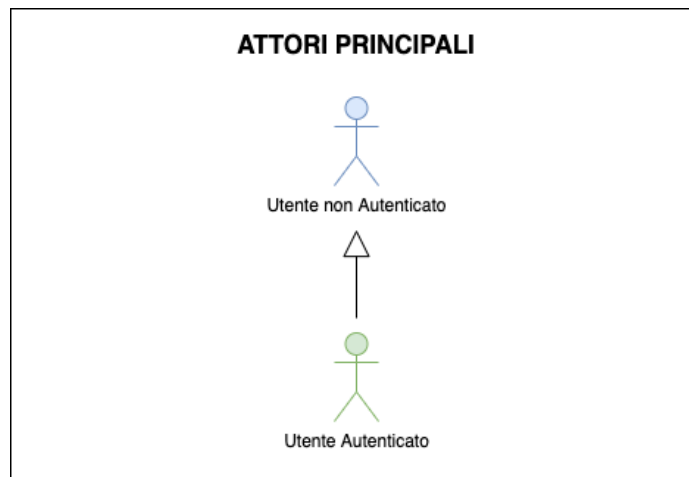


Figura 3.1: Attori Primari

La Figura 3.1 evidenzia la presenza di 2 diversi attori primari:

- * **Utente non Autenticato:** Utente che non risulta Autenticato

- * **Utente Autenticato:** Utente che risulta autenticato ed ha accesso a tutte le funzionalità

3.2 Casi d'uso

Per lo studio dei casi d'uso del prodotto, sono stati creati dei diagrammi. I diagrammi dei casi d'uso (in inglese *Use Case Diagram*) sono diagrammi di tipo **Unified Modeling Language (UML)**_g dedicati alla descrizione delle funzioni o servizi offerti da un sistema, così come sono percepiti e utilizzati dagli attori che interagiscono col sistema stesso [5]. Essendo il progetto finalizzato alla creazione di un applicativo che permetta di svolgere le operazioni in modo semplice, le interazioni da parte dell'utilizzatore sono state opportunamente ridotte allo stretto necessario. Tuttavia, essendo le due pagine da realizzare liste di elementi e form da compilare, il numero di use case, seppur semplici, è abbastanza elevato.

Nel seguito, sono descritti i diversi casi d'uso (*Use Case*) individuati, e, per ogni caso d'uso, si indicano un codice identificativo e un nome, il diagramma **UML** (quando significativo), gli attori, le pre condizioni, le post condizioni ed eventuali estensioni.

UC1 - Visualizzazione Lista *Pool* (Figura 3.2)

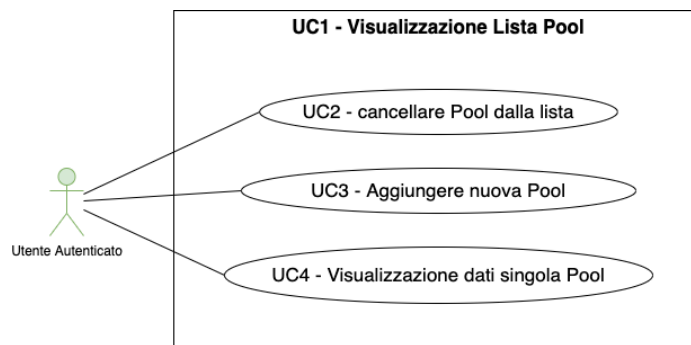


Figura 3.2: Diagramma UC1 - Visualizzazione Lista *Pool*

- * **Attori primari:** Utente Autenticato.
- * **Pre-condizione:** L'utente si è autenticato con username e password e si trova nella pagina contenente la lista delle "*Pool*" salvate.
- * **Flusso Principale degli Eventi:** L'Utente Autenticato accede alla pagina contenente la lista delle *Pool* salvate e consulta le informazioni in essa presenti.
- * **Post-condizione:** L'Utente ha visualizzato la lista di tutte le *Pool* salvate.
- * **Estensioni:**
 - Non sono presenti ordini con l'utente come attore. L'utente viene quindi notificato con un messaggio "*No Pool found...*".

UC2 - cancellare *Pool* dalla lista

- * **Attori primari:** Utente Autenticato.
- * **Pre-condizione:** L'Utente si è autenticato e si trova nella pagina contenente la lista delle *Pool* salvate.
- * **Flusso Principale degli Eventi:** L'Utente Autenticato preme sul bottone "cancel" di un elemento della lista e lo elimina.
- * **Post-condizione:** L'Utente ha cancellato la *Pool* selezionata dalla lista.

UC3 - Aggiungere nuova *Pool* (Figura 3.3)

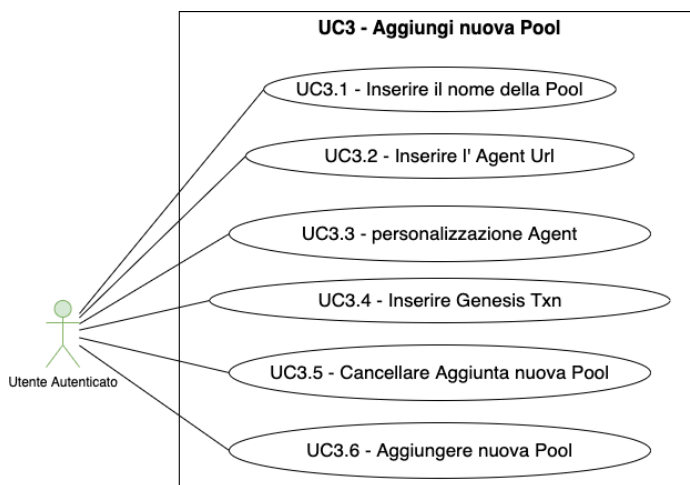


Figura 3.3: Diagramma UC3 - Aggiungere nuova *Pool*

- * **Attori primari:** Utente Autenticato.
- * **Pre-condizione:** L'Utente si è autenticato e si trova nella pagina contenente la lista delle *Pool* salvate.
- * **Flusso Principale degli Eventi:** L'Utente Autenticato preme il bottone "Add *Pool*" e si apre una *dialog* dove, si potrà inserire i dati per l'aggiunta di una nuova *Pool*.
- * **Post-condizione:** L'Utente si trova nella sezione per aggiungere una nuova *Pool* alla lista.

UC3.1 - Inserire il nome della *Pool*

- * **Attori primari:** Utente Autenticato.
- * **Pre-condizione:** L'Utente Autenticato si trova nella sezione per l'aggiunta di una nuova *Pool* alla lista.
- * **Flusso Principale degli Eventi:** l'Utente inserisce il nome della nuova *Pool*.
- * **Post-condizione:** L'Utente ha inserito il nome della nuova *Pool*.

UC3.2 - Inserire l'*Agent* URL

- * **Attori primari:** Utente Autenticato.
- * **Pre-condizione:** L'Utente Autenticato si trova nella sezione per l'aggiunta di una nuova *Pool* alla lista.
- * **Flusso Principale degli Eventi:** L'utente vuole inserire l'*Agent* URL.
- * **Post-condizione:** L'Utente ha inserito l'*Agent* URL.

UC3.3 - Personalizzazione *Agent* (Figura 3.4)

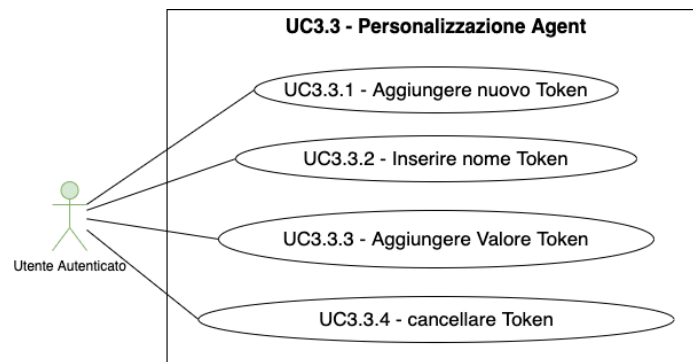


Figura 3.4: Diagramma UC3.3 - Personalizzazione *Agent*

- * **Attori primari:** Utente Autenticato.
- * **Pre-condizione:** L'Utente Autenticato si trova nella sezione per l'aggiunta di una nuova *Pool* alla lista.
- * **Flusso Principale degli Eventi:** l'Utente personalizza l'*Agent* mediante l'Aggiunta di un nuovo token (UC 3.3.1), inserendo il nome (UC3.3.2) e il valore (UC3.3.3).
- * **Post-condizione:** l'Utente ha personalizzato l'*Agent*.

UC3.3.1 - Aggiungere nuovo Token

- * **Attori primari:** Utente Autenticato.
- * **Pre-condizione:** L'Utente Autenticato si trova nella sezione per l'aggiunta di una nuova *Pool* alla lista e vuole personalizzare l'*Agent*.
- * **Flusso Principale degli Eventi:** L'Utente Autenticato preme il bottone "new token" e aggiunge una nuova coppia nome-valore vuoti alla lista dei token.
- * **Post-condizione:** L'Utente ha aggiunto una nuova coppia nome-valore vuoti alla lista dei token.

UC3.3.2 - Inserire nome Token

- * **Attori primari:** Utente Autenticato.
- * **Pre-condizione:** L'Utente Autenticato si trova nella sezione per l'aggiunta di una nuova *Pool* alla lista e ha aggiunto un nuovo token vuoto.
- * **Flusso Principale degli Eventi:** L'Utente Autenticato inserisce il nome del token.
- * **Post-condizione:** L'Utente ha inserito il nome del token.

UC3.3.3 - Aggiungere Valore Token

- * **Attori primari:** Utente Autenticato.
- * **Pre-condizione:** L'Utente Autenticato si trova nella sezione per l'aggiunta di una nuova *Pool* alla lista e ha aggiunto un nuovo token vuoto.
- * **Flusso Principale degli Eventi:** L'Utente Autenticato inserisce il valore del token.
- * **Post-condizione:** L'Utente ha inserito il valore del token.

UC3.3.4 - cancellare Token

- * **Attori primari:** Utente Autenticato.
- * **Pre-condizione:** L'Utente Autenticato si trova nella sezione per l'aggiunta di una nuova *Pool* alla lista e vuole cancellare un token esistente.
- * **Flusso Principale degli Eventi:** L'Utente Autenticato preme il bottone per cancellare il token selezionato dalla lista.
- * **Post-condizione:** L'Utente ha cancellato il token selezionato dalla lista.

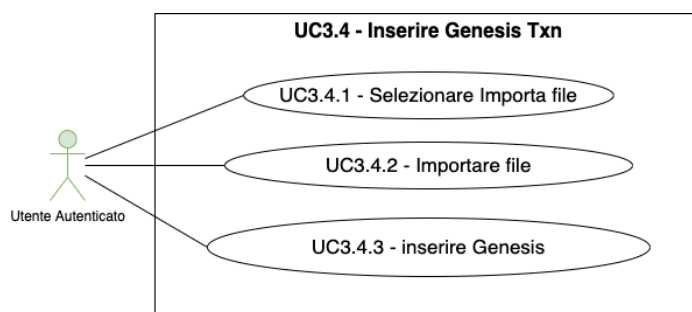
UC3.4 - Inserire Genesis Txn (Figura 3.5)

Figura 3.5: Diagramma UC3.4 - Inserire Genesis Txn

- * **Attori primari:** Utente Autenticato.

- * **Pre-condizione:** L'Utente Autenticato si trova nella sezione per l'aggiunta di una nuova *Pool* alla lista e vuole aggiungere il Genesis Txn.
- * **Flusso Principale degli Eventi:** L'Utente Autenticato vuole inserire il Genesis Txn o selezionando di importare un file (UC3.4.1) e selezionando il file da importare (UC3.4.2) o inserendo manualmente il Genesis Txn (UC3.4.3).
- * **Post-condizione:** L'Utente ha inserito il Genesis Txn.

UC3.4.1 - Selezionare Importa file

- * **Attori primari:** Utente Autenticato.
- * **Pre-condizione:** L'Utente Autenticato si trova nella sezione per l'aggiunta di una nuova *Pool* alla lista e vuole aggiungere il Genesis Txn importando un file.
- * **Flusso Principale degli Eventi:** L'Utente Autenticato seleziona l'opzione di inserire il Genesis Txn mediante l'importazione di un file.
- * **Post-condizione:** L'Utente ha selezionato l'opzione di inserire il Genesis Txn mediante l'importazione di un file.

UC3.4.2 - Importare file

- * **Attori primari:** Utente Autenticato.
- * **Pre-condizione:** L'Utente Autenticato si trova nella sezione per l'aggiunta di una nuova *Pool* alla lista e ha selezionato l'opzione di caricare il Genesis Txn importando un file..
- * **Flusso Principale degli Eventi:** L'Utente Autenticato seleziona il file .txt o .json da importare.
- * **Post-condizione:** L'Utente ha importato il file selezionato e può visualizzare il testo del file importato.

UC3.4.3 - inserire Genesis

- * **Attori primari:** Utente Autenticato.
- * **Pre-condizione:** L'Utente Autenticato si trova nella sezione per l'aggiunta di una nuova *Pool* alla lista e ha deciso di inserire il Genesis Txn manualmente.
- * **Flusso Principale degli Eventi:** L'Utente inserisce manualmente il testo del Genesis Txn.
- * **Post-condizione:** L'Utente ha inserito manualmente il Genesis Txn.

UC3.5 - Cancellare Aggiunta nuova *Pool*

- * **Attori primari:** Utente Autenticato.
- * **Pre-condizione:** L'Utente Autenticato si trova nella sezione per l'aggiunta di una nuova *Pool* alla lista e ha deciso di annullarne l'inserimento.

- * **Flusso Principale degli Eventi:** L'Utente Autenticato preme il bottone "cancel" per annullare l'aggiunta di una nuova *Pool* nella lista con conseguente annullamento dell'operazione e la chiusura della *dialog*.
- * **Post-condizione:** L'Utente ha annullato l'aggiunta di una nuova *Pool* alla lista e si ritrova nuovamente nella pagina dove può visualizzare la lista di *Pool* salvate (UC1).

UC3.6 - Aggiungere nuova *Pool*

- * **Attori primari:** Utente Autenticato.
- * **Pre-condizione:** L'Utente Autenticato si trova nella sezione per l'aggiunta di una nuova *Pool* alla lista e ha deciso di confermare l'inserimento.
- * **Flusso Principale degli Eventi:** L'Utente Autenticato preme il bottone "Add" per aggiungere la nuova *Pool* alla lista con la conseguente chiusura della *dialog*.
- * **Post-condizione:** L'Utente ha confermato l'aggiunta di una nuova *Pool* alla lista e si ritrova nuovamente nella pagina dove può visualizzare la lista di *Pool* salvate (UC1) compresa quella appena inserita.

UC4 - Visualizzazione dati singola *Pool* (Figura 3.6)

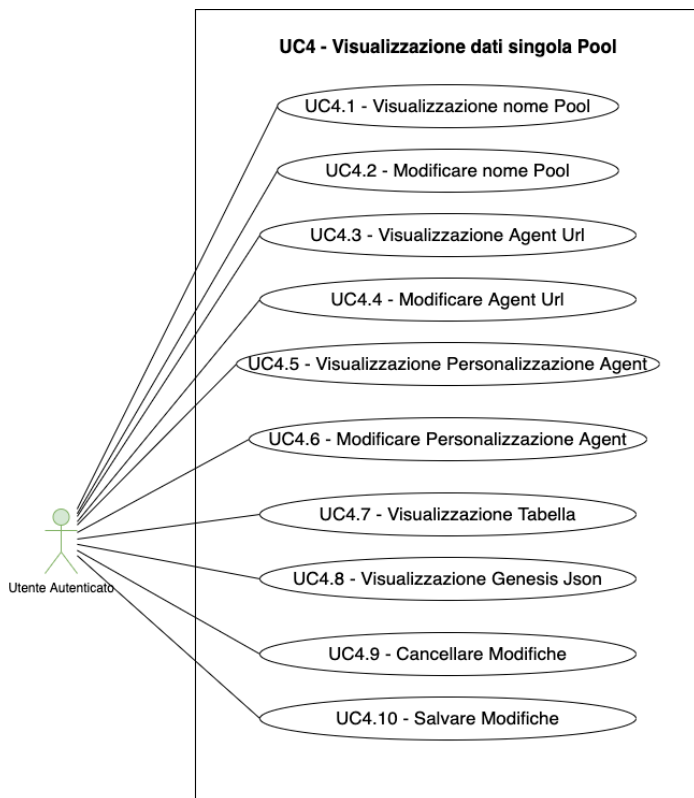


Figura 3.6: Diagramma UC4 - Visualizzazione dati singola *Pool*

- * **Attori primari:** Utente Autenticato.
- * **Pre-condizione:** L'Utente Autenticato si trova nella pagina contenente la lista delle *Pool* salvate e vuole visualizzare i dati di uno degli elementi della lista.
- * **Flusso Principale degli Eventi:** L'Utente Autenticato preme una delle *card* che formano la lista e si apre una *dialog* dove al suo interno ci sono i dati della *Pool* selezionata.
- * **Post-condizione:** L'Utente si trova nella sezione per visualizzare i dati della *Pool* selezionata.

UC4.1 - Visualizzazione nome *Pool*

- * **Attori primari:** Utente Autenticato.
- * **Pre-condizione:** L'Utente Autenticato si trova nella sezione per la visualizzazione dei dati della *Pool* selezionata e vuole visualizzarne il nome.
- * **Flusso Principale degli Eventi:** L'Utente Autenticato visualizza il nome della *Pool*.
- * **Post-condizione:** L'Utente ha visualizzato il nome della *Pool*.

UC4.2 - Modificare nome *Pool*

- * **Attori primari:** Utente Autenticato.
- * **Pre-condizione:** L'Utente Autenticato si trova nella sezione per la visualizzazione dei dati della *Pool* selezionata e vuole modificarne il nome.
- * **Flusso Principale degli Eventi:** L'Utente Autenticato modifica il nome della *Pool*.
- * **Post-condizione:** L'Utente ha modificato il nome della *Pool*.

UC4.3 - Visualizzazione *Agent* URL

- * **Attori primari:** Utente Autenticato.
- * **Pre-condizione:** L'Utente Autenticato si trova nella sezione per la visualizzazione dei dati della *Pool* selezionata e vuole visualizzarne l'*Agent* URL.
- * **Flusso Principale degli Eventi:** L'Utente Autenticato visualizza l'*Agent* URL della *Pool*.
- * **Post-condizione:** L'Utente ha visualizzato l'*Agent* URL della *Pool*.

UC4.4 - Modificare *Agent* URL

- * **Attori primari:** Utente Autenticato.
- * **Pre-condizione:** L'Utente Autenticato si trova nella sezione per la visualizzazione dei dati della *Pool* selezionata e vuole modificarne l'*Agent* URL.

- * **Flusso Principale degli Eventi:** L'Utente Autenticato modifica l'*Agent* URL della *Pool*.
- * **Post-condizione:** L'Utente ha modificato l'*Agent* URL della *Pool*.

UC4.5 - Visualizzazione Personalizzazione *Agent*

- * **Attori primari:** Utente Autenticato.
- * **Pre-condizione:** L'Utente Autenticato si trova nella sezione per la visualizzazione dei dati della *Pool* selezionata e vuole visualizzarne le varie chiavi nome-valore dei token di personalizzazione dell'*Agent*.
- * **Flusso Principale degli Eventi:** L'Utente Autenticato visualizza i vari token di personalizzazione dell'*Agent*.
- * **Post-condizione:** L'Utente ha visualizzato le varie chiavi nome-valore dei token di personalizzazione dell'*Agent*.

UC4.6 - Modificare Personalizzazione *Agent* (Figura 3.7)

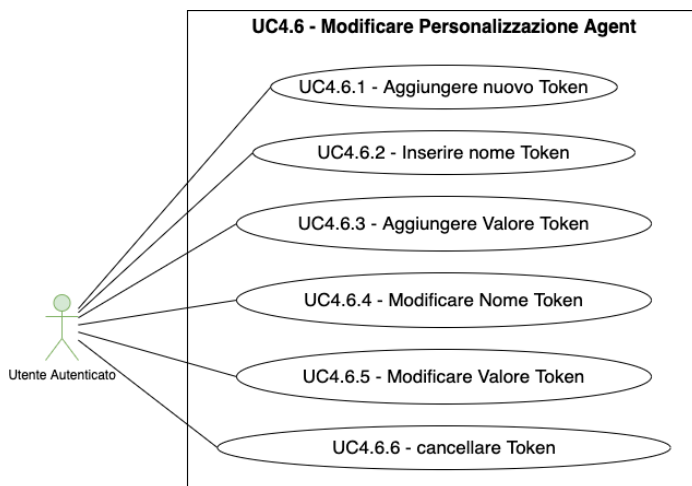


Figura 3.7: Diagramma UC4.6 - Modificare Personalizzazione *Agent*

- * **Attori primari:** Utente Autenticato.
- * **Pre-condizione:** L'Utente Autenticato si trova nella sezione per la visualizzazione dei dati della *Pool* selezionata e vuole modificarne le varie chiavi nome-valore dei token di personalizzazione dell'*Agent*.
- * **Flusso Principale degli Eventi:** L'Utente Autenticato visualizza i vari token di personalizzazione dell'*Agent*.
- * **Post-condizione:** L'Utente ha modificato le varie chiavi nome-valore dei token di personalizzazione dell'*Agent*.

UC4.6.1 - Aggiungere nuovo Token

- * **Attori primari:** Utente Autenticato.
- * **Pre-condizione:** L'Utente Autenticato si trova nella sezione per la visualizzazione dei dati della *Pool* selezionata e vuole aggiungere una nuova coppia chiave-valore vuota alla lista dei token di personalizzazione degli *Agent*.
- * **Flusso Principale degli Eventi:** L'Utente Autenticato preme il bottone "new token" e aggiunge una nuova coppia nome-valore vuoti alla lista dei token.
- * **Post-condizione:** L'Utente ha aggiunto una nuova coppia nome-valore vuoti alla lista dei token.

UC4.6.2 - Inserire nome Token

- * **Attori primari:** Utente Autenticato.
- * **Pre-condizione:** L'Utente Autenticato si trova nella sezione per la visualizzazione dei dati della *Pool* selezionata e vuole aggiungere il nome al token.
- * **Flusso Principale degli Eventi:** L'Utente Autenticato inserisce il nome del token.
- * **Post-condizione:** L'Utente ha inserito il nome del token.

UC4.6.3 - Aggiungere Valore Token

- * **Attori primari:** Utente Autenticato.
- * **Pre-condizione:** L'Utente Autenticato si trova nella sezione per la visualizzazione dei dati della *Pool* selezionata e vuole aggiungere il valore al token.
- * **Flusso Principale degli Eventi:** L'Utente Autenticato inserisce il valore del token.
- * **Post-condizione:** L'Utente ha inserito il valore del token.

UC4.6.4 - Modificare Nome Token

- * **Attori primari:** Utente Autenticato.
- * **Pre-condizione:** L'Utente Autenticato si trova nella sezione per la visualizzazione dei dati della *Pool* selezionata e vuole modificare il nome al token.
- * **Flusso Principale degli Eventi:** L'Utente Autenticato modificare il nome del token.
- * **Post-condizione:** L'Utente ha modificato il nome del token.

UC4.6.5 - Modificare Valore Token

- * **Attori primari:** Utente Autenticato.
- * **Pre-condizione:** L'Utente Autenticato si trova nella sezione per la visualizzazione dei dati della *Pool* selezionata e vuole modificare il valore al token.
- * **Flusso Principale degli Eventi:** L'Utente Autenticato modificare il valore del token.
- * **Post-condizione:** L'Utente ha modificato il valore del token.

UC4.6.6 - cancellare Token

- * **Attori primari:** Utente Autenticato.
- * **Pre-condizione:** L'Utente Autenticato si trova nella sezione per la visualizzazione dei dati della *Pool* selezionata e vuole cancellare un token esistente.
- * **Flusso Principale degli Eventi:** L'Utente Autenticato preme il bottone per cancellare il token selezionato dalla lista.
- * **Post-condizione:** L'Utente ha cancellato il token selezionato dalla lista.

UC4.7 - Visualizzazione Tabella (Figura 3.8)

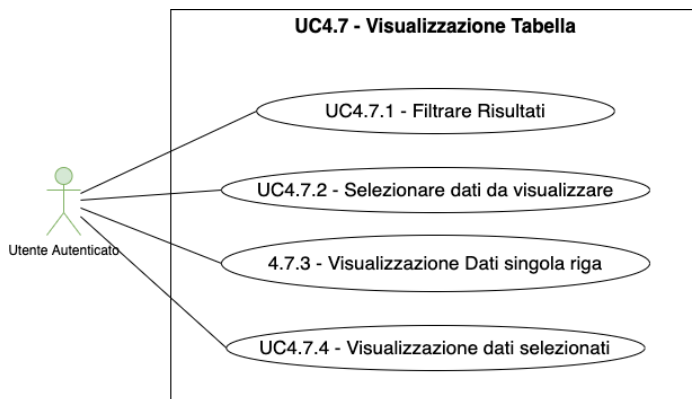


Figura 3.8: Diagramma UC4.7 - Visualizzazione Tabella

- * **Attori primari:** Utente Autenticato.
- * **Pre-condizione:** L'Utente Autenticato si trova nella sezione per la visualizzazione dei dati della *Pool* selezionata e vuole visualizzare i dati all'interno della tabella.
- * **Flusso Principale degli Eventi:** L'Utente Autenticato sceglie quali valori visualizzare all'interno della tabella (UC4.7.2) e li visualizza.
- * **Post-condizione:** L'Utente ha visualizzato i dati della tabella che ha selezionato.

UC4.7.1 - Filtrare Risultati

- * **Attori primari:** Utente Autenticato.
- * **Pre-condizione:** L'Utente Autenticato si trova nella sezione per la visualizzazione dei dati della *Pool* selezionata e vuole filtrare i dati della tabella.
- * **Flusso Principale degli Eventi:** L'Utente Autenticato inserisce la *keyword* da ricercare tra i dati della tabella.
- * **Post-condizione:** L'Utente visualizza i dati filtrati nella tabella.

UC4.7.2 - Selezionare dati da visualizzare

- * **Attori primari:** Utente Autenticato.
- * **Pre-condizione:** L'Utente Autenticato si trova nella sezione per la visualizzazione dei dati della *Pool* selezionata e vuole selezionare i dati da visualizzare nella tabella.
- * **Flusso Principale degli Eventi:** L'Utente Autenticato seleziona i dati da visualizzare nella tabella.
- * **Post-condizione:** L'Utente ha selezionato i dati selezionati nella tabella.

4.7.3 - Visualizzazione Dati singola riga (Figura 3.9)

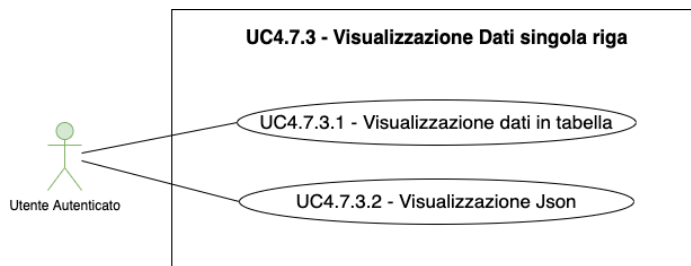


Figura 3.9: Diagramma UC4.7.3 - Visualizzazione Dati singola riga

- * **Attori primari:** Utente Autenticato.
- * **Pre-condizione:** L'Utente Autenticato si trova nella sezione per la visualizzazione dei dati della *Pool* selezionata e vuole visualizzare tutti i dati di una singola riga.
- * **Flusso Principale degli Eventi:** L'Utente Autenticato preme il pulsante *eye* per aprire la pagina di visualizzazione di tutti i dati della riga selezionata.
- * **Post-condizione:** L'Utente si trova nella pagina di visualizzazione dei dati della riga selezionata e ne può visualizzare il contenuto.

UC4.7.3.1 - Visualizzazione dati in tabella

- * **Attori primari:** Utente Autenticato.
- * **Pre-condizione:** L'Utente Autenticato si trova nella pagina per la visualizzazione di tutti i dati della riga selezionata della tabella e vuole visualizzarli in forma tabellare.
- * **Flusso Principale degli Eventi:** L'Utente Autenticato visualizza tutti i dati della riga selezionata in forma tabellare.
- * **Post-condizione:** L'Utente ha visualizzato i dati della riga selezionata in forma tabellare.

UC4.7.3.2 - Visualizzazione Json

- * **Attori primari:** Utente Autenticato.
- * **Pre-condizione:** L'Utente Autenticato si trova nella pagina per la visualizzazione di tutti i dati della riga selezionata della tabella e vuole visualizzarli nel formato [Json](#).
- * **Flusso Principale degli Eventi:** L'Utente Autenticato visualizza i dati della riga della tabella nel formato [Json](#).
- * **Post-condizione:** L'Utente ha visualizzato i dati della riga selezionata in formato [Json](#).

UC4.7.4 - Visualizzazione dati selezionati

- * **Attori primari:** Utente Autenticato.
- * **Pre-condizione:** L'Utente Autenticato si trova nella sezione per la visualizzazione dei dati della *Pool* selezionata e visualizzare i dati selezionati della tabella.
- * **Flusso Principale degli Eventi:** L'Utente Autenticato visualizza i dati selezionati della tabella.
- * **Post-condizione:** L'Utente Autenticato ha visualizzato i dati selezionati della tabella.

UC4.8 - Visualizzazione Genesis Json

- * **Attori primari:** Utente Autenticato.
- * **Pre-condizione:** L'Utente Autenticato si trova nella sezione per la visualizzazione dei dati della *Pool* selezionata e vuole visualizzarli nel formato [Json](#).
- * **Flusso Principale degli Eventi:** L'Utente Autenticato visualizza i dati della *Pool* selezionata nel formato [Json](#).
- * **Post-condizione:** L'Utente ha visualizzato i dati nel formato [Json](#).

UC4.9 - Cancellare Modifiche

- * **Attori primari:** Utente Autenticato.
- * **Pre-condizione:** L'Utente Autenticato si trova nella sezione per la visualizzazione dei dati della *Pool* selezionata e vuole cancellare le modifiche effettuate.
- * **Flusso Principale degli Eventi:** L'Utente Autenticato preme il bottone *cancel* e cancella tutte le modifiche con la conseguente chiusura della *dialog* per la visualizzazione dei dati della *Pool* selezionata.
- * **Post-condizione:** L'Utente ha cancellato le modifiche apportate e si ritrova nuovamente nella pagina con la lista delle *Pool* salvate (UC1).

UC4.10 - Salvare Modifiche

- * **Attori primari:** Utente Autenticato.
- * **Pre-condizione:** L'Utente Autenticato si trova nella sezione per la visualizzazione dei dati della *Pool* selezionata e vuole confermare le modifiche effettuate.
- * **Flusso Principale degli Eventi:** L'Utente Autenticato preme il bottone *save* e salva tutte le modifiche con la conseguente chiusura della *dialog* per la visualizzazione dei dati della *Pool* selezionata.
- * **Post-condizione:** L'Utente ha salvato le modifiche apportate e si ritrova nuovamente nella pagina con la lista delle *Pool* salvate (UC1).

UC5 - Visualizzazione lista *Agent* (Figura 3.10)

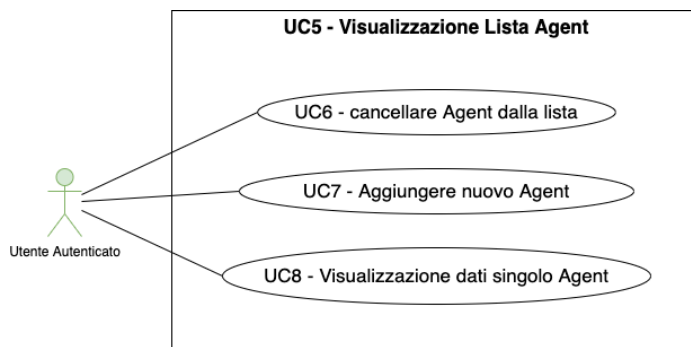


Figura 3.10: Diagramma UC5 - Visualizzazione Lista *Agent*

- * **Attori primari:** Utente Autenticato.
- * **Pre-condizione:** L'Utente Autenticato si trova nella pagina contenente la lista degli *Agent* salvati.
- * **Flusso Principale degli Eventi:** ' Utente Autenticato accede alla pagina contenente la lista degli *Agent* salvati e consulta le informazioni in essa presenti.
- * **Post-condizione:** L'utente ha visualizzato la lista degli *Agent* salvati.

UC6 - cancellare *Agent* dalla lista

- * **Attori primari:** Utente Autenticato.
- * **Pre-condizione:** L'Utente si è autenticato e si trova nella pagina contenente la lista degli *Agent* salvati.
- * **Flusso Principale degli Eventi:** L'Utente Autenticato preme sul bottone "cancel" di un elemento della lista e lo elimina.
- * **Post-condizione:** L' Utente ha cancellato l'*Agent* selezionato dalla lista.

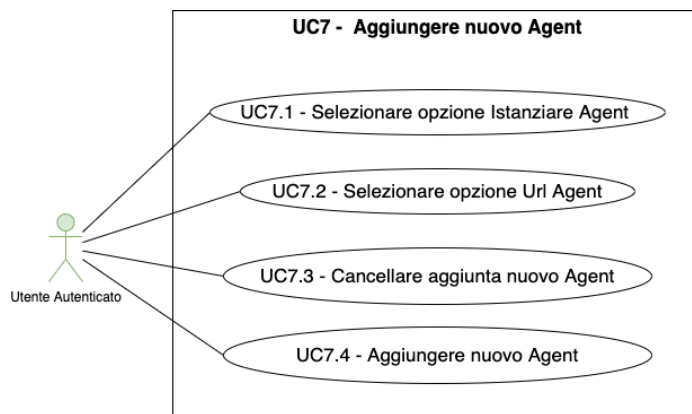
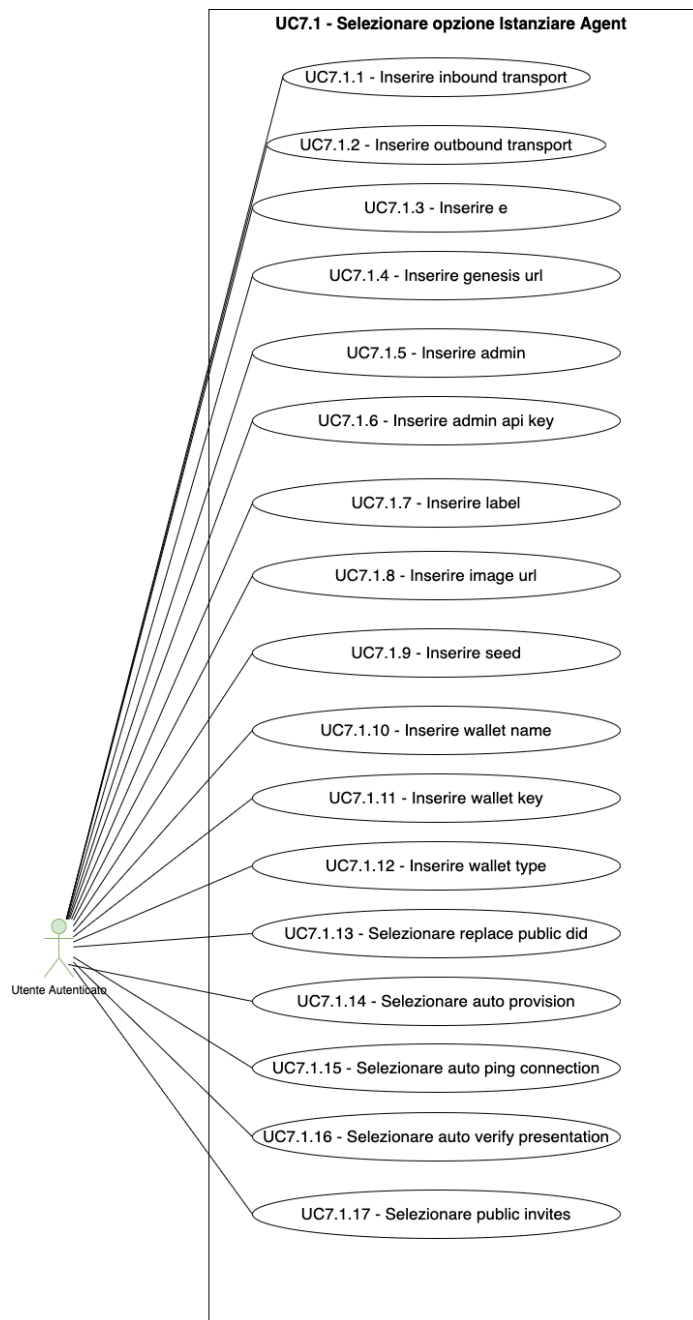
UC7 - Aggiungere nuovo *Agent* (Figura 3.11)

Figura 3.11: Diagramma UC7 - Aggiungere nuovo *Agent*

- * **Attori primari:** Utente Autenticato.
- * **Pre-condizione:** L' Utente si è autenticato e si trova nella pagina contenente la lista degli *Agent* salvati.
- * **Flusso Principale degli Eventi:** L' Utente Autenticato preme il bottone "Add *Agent*" e si apre una *dialog* dove si potrà inserire i dati per l' aggiunta di un nuovo *Agent*.
- * **Post-condizione:** L'Utente si trova nella sezione per l'aggiunta di un nuovo *Agent*.

UC7.1 - Selezionare opzione istanziare nuovo *Agent* (Figura 3.12)**Figura 3.12:** Diagramma UC7.1 - Selezionare opzione istanziare nuovo *Agent*

- * **Attori primari:** Utente Autenticato.
- * **Pre-condizione:** L'Utente Autenticato di trova nella sezione per l'aggiunta di un nuovo *Agent*.

- * **Flusso Principale degli Eventi:** L'Utente vuole istanziare un nuovo *Agent* e seleziona l'opzione "instance *Agent*".
- * **Post-condizione:** L'Utente ha selezionato l'opzione "instance *Agent*" e ha davanti il form per istanziare il nuovo *Agent*.

UC7.1.1 - Inserire inbound transport (Figura 3.13)

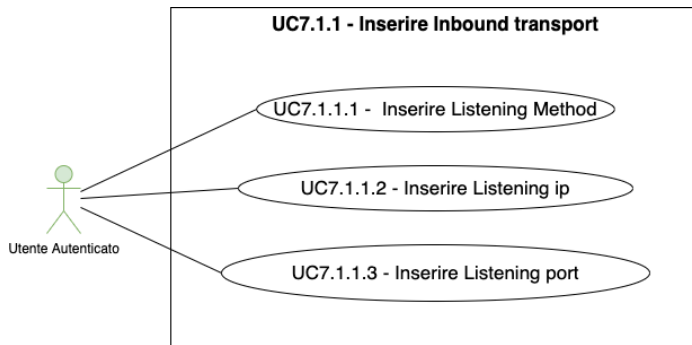


Figura 3.13: Diagramma UC7.1.1 - Inserire inbound transport

- * **Attori primari:** Utente Autenticato.
- * **Pre-condizione:** L'Utente si trova nella sezione per l'aggiunta di un nuovo *Agent* e ha selezionato di volerlo istanziare.
- * **Flusso Principale degli Eventi:** L'Utente inserisce i dati dell'inbound transport.
- * **Post-condizione:** L'Utente ha inserito i dati dell'inbound transport.

UC7.1.1.1 - Inserire Listening Method

- * **Attori primari:** Utente Autenticato.
- * **Pre-condizione:** L'Utente si trova nella sezione per l'aggiunta di un nuovo *Agent* e ha selezionato di volerlo istanziare.
- * **Flusso Principale degli Eventi:** L'Utente inserisce il listening method.
- * **Post-condizione:** L'Utente ha inserito il listening method.

UC7.1.1.2 - Inserire Listening ip

- * **Attori primari:** Utente Autenticato.
- * **Pre-condizione:** L'Utente si trova nella sezione per l'aggiunta di un nuovo *Agent* e ha selezionato di volerlo istanziare.
- * **Flusso Principale degli Eventi:** L'Utente inserisce il listening ip.
- * **Post-condizione:** L'Utente ha inserito il listening ip.

UC7.1.1.2 - Inserire Listening ip

- * **Attori primari:** Utente Autenticato.
- * **Pre-condizione:** L'Utente si trova nella sezione per l'aggiunta di un nuovo *Agent* e ha selezionato di volerlo istanziare.
- * **Flusso Principale degli Eventi:** L'Utente inserisce il listening port.
- * **Post-condizione:** L'Utente ha inserito il listening port.

UC7.1.2 - Inserire outbound transport

- * **Attori primari:** Utente Autenticato.
- * **Pre-condizione:** L'Utente si trova nella sezione per l'aggiunta di un nuovo *Agent* e ha selezionato di volerlo istanziare.
- * **Flusso Principale degli Eventi:** L'Utente inserisce l'outbound transport.
- * **Post-condizione:** L'Utente ha inserito l'outbound transport.

UC7.1.3 - Inserire e

- * **Attori primari:** Utente Autenticato.
- * **Pre-condizione:** L'Utente si trova nella sezione per l'aggiunta di un nuovo *Agent* e ha selezionato di volerlo istanziare.
- * **Flusso Principale degli Eventi:** L'Utente inserisce e.
- * **Post-condizione:** L'Utente ha inserito e.

UC7.1.4 - Inserire genesis URL

- * **Attori primari:** Utente Autenticato.
- * **Pre-condizione:** L'Utente si trova nella sezione per l'aggiunta di un nuovo *Agent* e ha selezionato di volerlo istanziare.
- * **Flusso Principale degli Eventi:** L'Utente inserisce il genesis URL.
- * **Post-condizione:** L'Utente ha inserito il genesis URL.

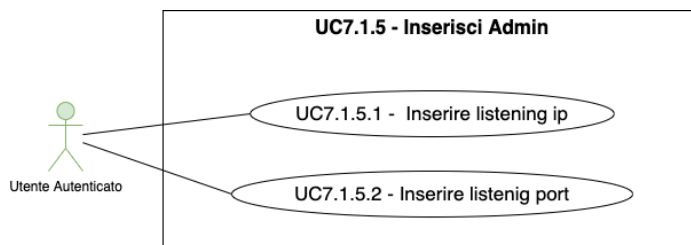
UC7.1.5 - Inserire admin (Figura 3.14)

Figura 3.14: Diagramma UC7.1.5 - Inserire admin

- * **Attori primari:** Utente Autenticato.
- * **Pre-condizione:** L'Utente si trova nella sezione per l'aggiunta di un nuovo *Agent* e ha selezionato di volerlo istanziare.
- * **Flusso Principale degli Eventi:** L'Utente inserisce l'admin.
- * **Post-condizione:** L'Utente ha inserito l'admin.

UC7.1.5.1 - Inserire listening ip

- * **Attori primari:** Utente Autenticato.
- * **Pre-condizione:** L'Utente si trova nella sezione per l'aggiunta di un nuovo *Agent*, ha selezionato di volerlo istanziare e vuole inserire i dati dell'admin.
- * **Flusso Principale degli Eventi:** L'Utente inserisce il listening ip.
- * **Post-condizione:** L'utente ha inserito il listening ip.

UC7.1.5.2 - Inserire listening port

- * **Attori primari:** Utente Autenticato.
- * **Pre-condizione:** L'Utente si trova nella sezione per l'aggiunta di un nuovo *Agent*, ha selezionato di volerlo istanziare e vuole inserire i dati dell'admin.
- * **Flusso Principale degli Eventi:** L'Utente inserisce il listening port.
- * **Post-condizione:** L'utente ha inserito il listening port.

UC7.1.6 - Inserire admin api key

- * **Attori primari:** Utente Autenticato.
- * **Pre-condizione:** L'Utente si trova nella sezione per l'aggiunta di un nuovo *Agent* e ha selezionato di volerlo istanziare.
- * **Flusso Principale degli Eventi:** L'Utente inserisce l'admin api key.
- * **Post-condizione:** L'Utente ha inserito l'admin api key.

UC7.1.7 - Inserire label

- * **Attori primari:** Utente Autenticato.
- * **Pre-condizione:** L'Utente si trova nella sezione per l'aggiunta di un nuovo *Agent* e ha selezionato di volerlo istanziare.
- * **Flusso Principale degli Eventi:** L'Utente inserisce la label.
- * **Post-condizione:** L'Utente ha inserito la label.

UC7.1.8 - Inserire image URL

- * **Attori primari:** Utente Autenticato.
- * **Pre-condizione:** L'Utente si trova nella sezione per l'aggiunta di un nuovo *Agent* e ha selezionato di volerlo istanziare.
- * **Flusso Principale degli Eventi:** L'Utente inserisce l'image URL.
- * **Post-condizione:** L'Utente ha inserito l'image URL.

UC7.1.9 - inserire seed

- * **Attori primari:** Utente Autenticato.
- * **Pre-condizione:** L'Utente si trova nella sezione per l'aggiunta di un nuovo *Agent* e ha selezionato di volerlo istanziare.
- * **Flusso Principale degli Eventi:** L'Utente inserisce il seed.
- * **Post-condizione:** L'Utente ha inserito il seed.

UC7.1.10 - Inserire wallet name

- * **Attori primari:** Utente Autenticato.
- * **Pre-condizione:** L'Utente si trova nella sezione per l'aggiunta di un nuovo *Agent* e ha selezionato di volerlo istanziare.
- * **Flusso Principale degli Eventi:** L'Utente inserisce il wallet name.
- * **Post-condizione:** L'Utente ha inserito il wallet name.

UC7.1.11 - Inserire il wallet key

- * **Attori primari:** Utente Autenticato.
- * **Pre-condizione:** L'Utente si trova nella sezione per l'aggiunta di un nuovo *Agent* e ha selezionato di volerlo istanziare.
- * **Flusso Principale degli Eventi:** L'Utente inserisce il wallet key.
- * **Post-condizione:** L'Utente ha inserito il wallet key.

UC7.1.12 - Inserire il wallet type

- * **Attori primari:** Utente Autenticato.
- * **Pre-condizione:** L'Utente si trova nella sezione per l'aggiunta di un nuovo *Agent* e ha selezionato di volerlo istanziare.
- * **Flusso Principale degli Eventi:** L'Utente inserisce il wallet type.
- * **Post-condizione:** L'Utente ha inserito il wallet type.

UC7.1.13 - Selezionare replace public did

- * **Attori primari:** Utente Autenticato.
- * **Pre-condizione:** L'Utente si trova nella sezione per l'aggiunta di un nuovo *Agent* e ha selezionato di volerlo istanziare.
- * **Flusso Principale degli Eventi:** L'Utente seleziona se attivare o disattivare il replace public did.
- * **Post-condizione:** L'Utente ha selezionato se attivare o disattivare il replace public did.

UC7.1.14 - Selezionare auto provision

- * **Attori primari:** Utente Autenticato.
- * **Pre-condizione:** L'Utente si trova nella sezione per l'aggiunta di un nuovo *Agent* e ha selezionato di volerlo istanziare.
- * **Flusso Principale degli Eventi:** L'Utente seleziona se attivare o disattivare l'auto provision.
- * **Post-condizione:** L'Utente ha selezionato se attivare o disattivare l'auto provision.

UC7.1.15 - Selezionare auto ping connection

- * **Attori primari:** Utente Autenticato.
- * **Pre-condizione:** L'Utente si trova nella sezione per l'aggiunta di un nuovo *Agent* e ha selezionato di volerlo istanziare.
- * **Flusso Principale degli Eventi:** L'Utente seleziona se attivare o disattivare l'auto ping connection.
- * **Post-condizione:** L'Utente ha selezionato se attivare o disattivare l'auto ping connection.

UC7.1.16 - Selezionare auto verify presentation

- * **Attori primari:** Utente Autenticato.
- * **Pre-condizione:** L'Utente si trova nella sezione per l'aggiunta di un nuovo *Agent* e ha selezionato di volerlo istanziare.
- * **Flusso Principale degli Eventi:** L'Utente seleziona se attivare o disattivare l'auto verify presentation.
- * **Post-condizione:** L'Utente ha selezionato se attivare o disattivare l'auto verify presentation.

UC7.1.17 - Selezionare public invites

- * **Attori primari:** Utente Autenticato.
- * **Pre-condizione:** L'Utente si trova nella sezione per l'aggiunta di un nuovo *Agent* e ha selezionato di volerlo istanziare.
- * **Flusso Principale degli Eventi:** L'Utente seleziona se attivare o disattivare il public invites presentation.
- * **Post-condizione:** L'Utente ha selezionato se attivare o disattivare il public invites.

UC7.2 - Selezionare opzione URL *Agent* (Figura 3.15)

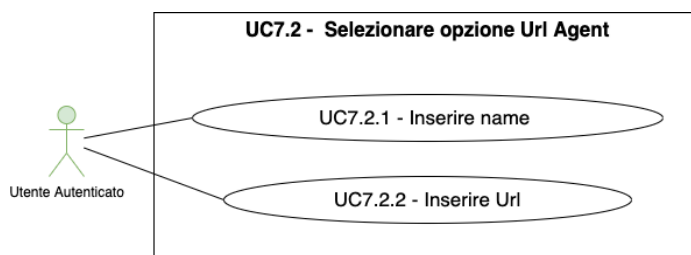


Figura 3.15: Diagramma UC7.2 - Selezionare opzione URL *Agent*

- * **Attori primari:** Utente Autenticato.
- * **Pre-condizione:** L'Utente Autenticato si trova nella sezione per l'aggiunta di un nuovo *Agent*.
- * **Flusso Principale degli Eventi:** L'Utente vuole aggiungere un nuovo *Agent* mediante URL e seleziona l'opzione "URL *Agent*".
- * **Post-condizione:** L'Utente ha selezionato l'opzione "URL *Agent*" e ha davanti il form per aggiungere il nuovo *Agent* mediante URL.

UC7.2.1 - Inserire name

- * **Attori primari:** Utente Autenticato.
- * **Pre-condizione:** L'Utente si trova nella sezione per l'aggiunta di un nuovo *Agent* e ha selezionato di volerlo utilizzare l'URL *Agent*.
- * **Flusso Principale degli Eventi:** L'Utente inserisce il name.
- * **Post-condizione:** L'Utente ha inserito il name.

UC7.2.2 - Inserire URL

- * **Attori primari:** Utente Autenticato.
- * **Pre-condizione:** L'Utente si trova nella sezione per l'aggiunta di un nuovo *Agent* e ha selezionato di volerlo utilizzare l'URL *Agent*.

- * **Flusso Principale degli Eventi:** L'Utente inserisce l'URL.

- * **Post-condizione:** L'Utente ha inserito l'URL.

UC7.3 - Cancellare aggiunta nuovo *Agent*

- * **Attori primari:** Utente Autenticato.

- * **Pre-condizione:** L'Utente Autenticato si trova nella sezione per l'aggiunta di un nuovo *Agent* e ne vuole annullare l'aggiunta.

- * **Flusso Principale degli Eventi:** L'Utente preme il bottone "Cancel" per annullare l'aggiunta di un nuovo *Agent* con la conseguente chiusura della *dialog*.

- * **Post-condizione:** L'Utente ha annullato l'aggiunta di un nuovo *Agent* e si trova nuovamente nella pagina con la lista degli *Agent* salvati.

UC7.4 - Aggiungere nuovo *Agent*

- * **Attori primari:** Utente Autenticato.

- * **Pre-condizione:** L'Utente Autenticato si trova nella sezione per l'aggiunta di un nuovo *Agent* e ne vuole confermare l'aggiunta.

- * **Flusso Principale degli Eventi:** L'Utente preme il bottone "Add" per annullare l'aggiunta di un nuovo *Agent* con la conseguente chiusura della *dialog*.

- * **Post-condizione:** L'Utente ha aggiunto con successo un nuovo *Agent* e si trova nuovamente nella pagina con la lista degli *Agent* salvati dove in ultima posizione si troverà il nuovo *Agent* appena aggiunto.

UC8 - Visualizzazione dati singolo *Agent* (Figure 3.16 e 3.21)

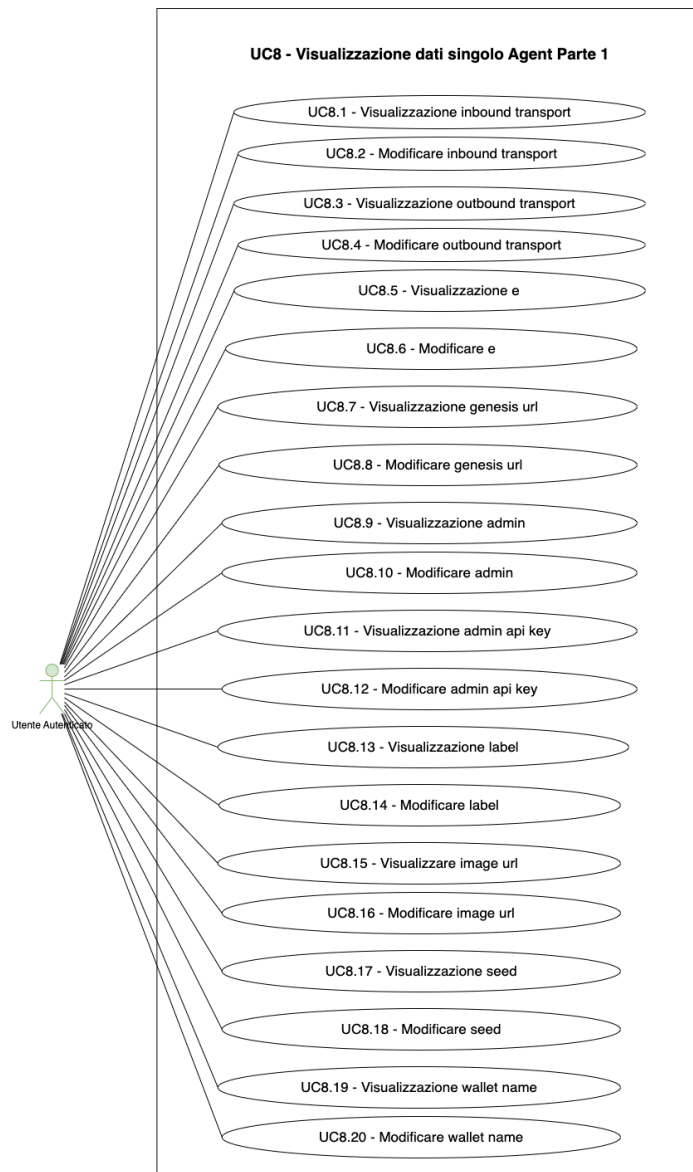


Figura 3.16: Diagramma UC8 Parte 1 - Visualizzazione dati singolo *Agent*

- * **Attori primari:** Utente Autenticato.
- * **Pre-condizione:** L'Utente Autenticato si trova nella pagina contenente la lista degli *Agent* salvati.
- * **Flusso Principale degli Eventi:** L' Utente Autenticato clicca su uno degli *Agent* della lista e si apre una *dialog* contenente i dati di quell'*Agent*.

- * **Post-condizione:** L'utente si trova nella sezione per la visualizzazione dei dati dell'*Agent* selezionato.

UC8.1 - Visualizzazione inbound transport (Figura 3.17)

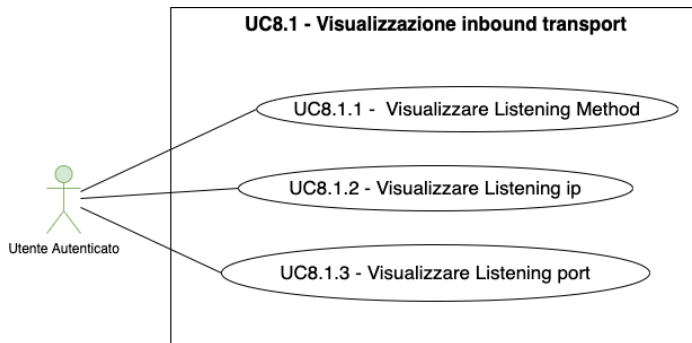


Figura 3.17: Diagramma UC8.1 - Visualizzazione inbound transport

- * **Attori primari:** Utente Autenticato.
- * **Pre-condizione:** L'Utente si trova nella sezione per la visualizzazione dei dati di un singolo *Agent* aggiunto mediante modalità "instance *Agent*".
- * **Flusso Principale degli Eventi:** L'Utente visualizza l'inbound transport.
- * **Post-condizione:** L'Utente ha visualizzato l'inbound transport.

UC8.1.1 - Visualizzare Listening Method

- * **Attori primari:** Utente Autenticato.
- * **Pre-condizione:** L'Utente si trova nella sezione per la visualizzazione dei dati di un singolo *Agent* aggiunto mediante modalità "instance *Agent*".
- * **Flusso Principale degli Eventi:** L'Utente visualizza il Listening method dell'inbound transport.
- * **Post-condizione:** L'Utente ha visualizzato il Listening method dell'inbound transport.

UC8.1.2 - Visualizzare Listening ip

- * **Attori primari:** Utente Autenticato.
- * **Pre-condizione:** L'Utente si trova nella sezione per la visualizzazione dei dati di un singolo *Agent* aggiunto mediante modalità "instance *Agent*".
- * **Flusso Principale degli Eventi:** L'Utente visualizza il Listening ip dell'inbound transport.
- * **Post-condizione:** L'Utente ha visualizzato il Listening ip dell'inbound transport.

UC8.1.3 - Visualizzare Listening port

- * **Attori primari:** Utente Autenticato.
- * **Pre-condizione:** L'Utente si trova nella sezione per la visualizzazione dei dati di un singolo *Agent* aggiunto mediante modalità "instance *Agent*".
- * **Flusso Principale degli Eventi:** L'Utente visualizza il Listening port dell'inbound transport.
- * **Post-condizione:** L'Utente ha visualizzato il Listening port dell'inbound transport.

UC8.2 - Modificare inbound transport (Figura 3.18)

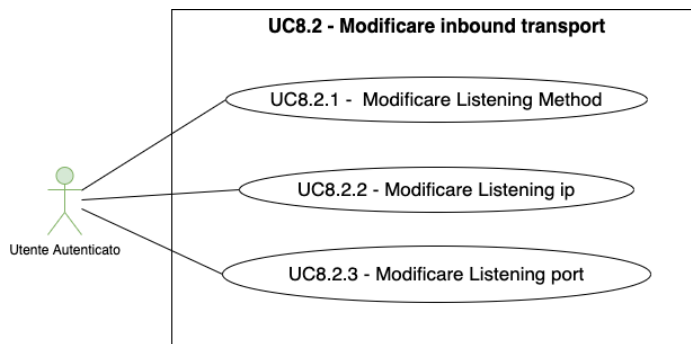


Figura 3.18: Diagramma UC8.2 - Modificare inbound transport

- * **Attori primari:** Utente Autenticato.
- * **Pre-condizione:** L'Utente si trova nella sezione per la visualizzazione dei dati di un singolo *Agent* aggiunto mediante modalità "instance *Agent*".
- * **Flusso Principale degli Eventi:** L'Utente modifica l'inbound transport.
- * **Post-condizione:** L'Utente ha modificato l'inbound transport.

UC8.2.1 - Modificare Listening Method

- * **Attori primari:** Utente Autenticato.
- * **Pre-condizione:** L'Utente si trova nella sezione per la visualizzazione dei dati di un singolo *Agent* aggiunto mediante modalità "instance *Agent*".
- * **Flusso Principale degli Eventi:** L'Utente modifica il Listening method dell'inbound transport.
- * **Post-condizione:** L'Utente ha modificato il Listening method dell'inbound transport.

UC8.2.2 - Modificare Listening ip

- * **Attori primari:** Utente Autenticato.
- * **Pre-condizione:** L'Utente si trova nella sezione per la visualizzazione dei dati di un singolo *Agent* aggiunto mediante modalità "instance *Agent*".
- * **Flusso Principale degli Eventi:** L'Utente modifica il Listening ip dell'inbound transport.
- * **Post-condizione:** L'Utente ha modificato il Listening ip dell'inbound transport.

UC8.2.3 - Modificare Listening port

- * **Attori primari:** Utente Autenticato.
- * **Pre-condizione:** L'Utente si trova nella sezione per la visualizzazione dei dati di un singolo *Agent* aggiunto mediante modalità "instance *Agent*".
- * **Flusso Principale degli Eventi:** L'Utente modifica il Listening port dell'inbound transport.
- * **Post-condizione:** L'Utente ha modificato il Listening port dell'inbound transport.

UC8.3 - Visualizzazione outbound transport

- * **Attori primari:** Utente Autenticato.
- * **Pre-condizione:** L'Utente si trova nella sezione per la visualizzazione dei dati di un singolo *Agent* aggiunto mediante modalità "instance *Agent*".
- * **Flusso Principale degli Eventi:** L'Utente visualizza l'outbound transport.
- * **Post-condizione:** L'Utente ha visualizzato l'outbound transport.

UC8.4 - Modificare outbound transport

- * **Attori primari:** Utente Autenticato.
- * **Pre-condizione:** L'Utente si trova nella sezione per la visualizzazione dei dati di un singolo *Agent* aggiunto mediante modalità "instance *Agent*".
- * **Flusso Principale degli Eventi:** L'Utente modifica l'outbound transport.
- * **Post-condizione:** L'Utente ha modificato l'outbound transport.

UC8.5 - Visualizzazione e

- * **Attori primari:** Utente Autenticato.
- * **Pre-condizione:** L'Utente si trova nella sezione per la visualizzazione dei dati di un singolo *Agent* aggiunto mediante modalità "instance *Agent*".
- * **Flusso Principale degli Eventi:** L'Utente visualizza l'e.
- * **Post-condizione:** L'Utente ha visualizzato l'e.

UC8.6 - Modificare e

- * **Attori primari:** Utente Autenticato.
- * **Pre-condizione:** L'Utente si trova nella sezione per la visualizzazione dei dati di un singolo *Agent* aggiunto mediante modalità "instance *Agent*".
- * **Flusso Principale degli Eventi:** L'Utente modifica l'e.
- * **Post-condizione:** L'Utente ha modificato l'e.

UC8.7 - Visualizzazione genesis URL

- * **Attori primari:** Utente Autenticato.
- * **Pre-condizione:** L'Utente si trova nella sezione per la visualizzazione dei dati di un singolo *Agent* aggiunto mediante modalità "instance *Agent*".
- * **Flusso Principale degli Eventi:** L'Utente visualizza il genesis URL.
- * **Post-condizione:** L'Utente ha visualizzato il genesis URL.

UC8.8 - Modificare genesis URL

- * **Attori primari:** Utente Autenticato.
- * **Pre-condizione:** L'Utente si trova nella sezione per la visualizzazione dei dati di un singolo *Agent* aggiunto mediante modalità "instance *Agent*".
- * **Flusso Principale degli Eventi:** L'Utente modifica il genesis URL.
- * **Post-condizione:** L'Utente ha modificato il genesis URL.

UC8.9 - Visualizzazione admin (Figura 3.19)

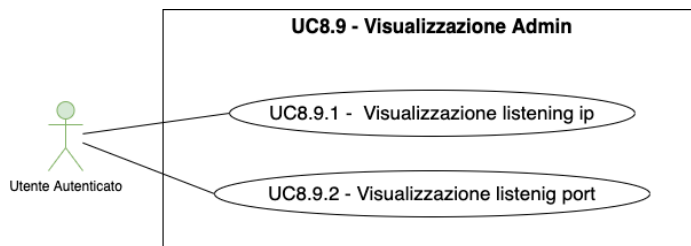


Figura 3.19: Diagramma UC8.9 - Visualizzazione admin

- * **Attori primari:** Utente Autenticato.
- * **Pre-condizione:** L'Utente si trova nella sezione per la visualizzazione dei dati di un singolo *Agent* aggiunto mediante modalità "instance *Agent*".
- * **Flusso Principale degli Eventi:** L'Utente visualizza l'admin.
- * **Post-condizione:** L'Utente ha visualizzato l'admin.

UC8.9.1 - Visualizzazione listening ip

- * **Attori primari:** Utente Autenticato.
- * **Pre-condizione:** L'Utente si trova nella sezione per la visualizzazione dei dati di un singolo *Agent* aggiunto mediante modalità "instance *Agent*".
- * **Flusso Principale degli Eventi:** L'Utente visualizza il Listening ip dell'admin.
- * **Post-condizione:** L'Utente ha visualizzato il Listening ip dell'admin.

UC8.9.2 - Visualizzazione listening port

- * **Attori primari:** Utente Autenticato.
- * **Pre-condizione:** L'Utente si trova nella sezione per la visualizzazione dei dati di un singolo *Agent* aggiunto mediante modalità "instance *Agent*".
- * **Flusso Principale degli Eventi:** L'Utente visualizza il Listening port dell'admin.
- * **Post-condizione:** L'Utente ha visualizzato il Listening port dell'admin.

UC8.10 - Modificare admin (Figura 3.20)

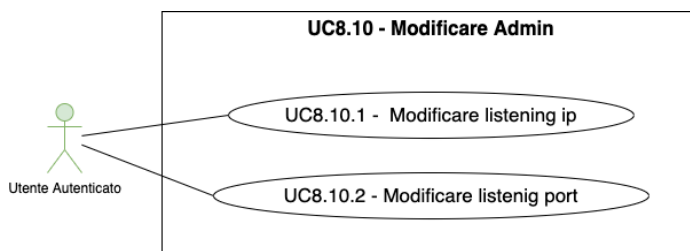


Figura 3.20: Diagramma UC8.10 - Modificare admin

- * **Attori primari:** Utente Autenticato.
- * **Pre-condizione:** L'Utente si trova nella sezione per la visualizzazione dei dati di un singolo *Agent* aggiunto mediante modalità "instance *Agent*".
- * **Flusso Principale degli Eventi:** L'Utente modifica l'admin.
- * **Post-condizione:** L'Utente ha modificato l'admin.

UC8.10.1 - Modificare listening ip

- * **Attori primari:** Utente Autenticato.
- * **Pre-condizione:** L'Utente si trova nella sezione per la visualizzazione dei dati di un singolo *Agent* aggiunto mediante modalità "instance *Agent*".
- * **Flusso Principale degli Eventi:** L'Utente modifica il Listening ip dell'admin.
- * **Post-condizione:** L'Utente ha modificato il Listening ip dell'admin.

UC8.10.2 - Modificare listening port

- * **Attori primari:** Utente Autenticato.
- * **Pre-condizione:** L'Utente si trova nella sezione per la visualizzazione dei dati di un singolo *Agent* aggiunto mediante modalità "instance *Agent*".
- * **Flusso Principale degli Eventi:** L'Utente modifica il Listening port dell'admin.
- * **Post-condizione:** L'Utente ha modificato il Listening port dell'admin.

UC8.11 - Visualizzazione admin api key

- * **Attori primari:** Utente Autenticato.
- * **Pre-condizione:** L'Utente si trova nella sezione per la visualizzazione dei dati di un singolo *Agent* aggiunto mediante modalità "instance *Agent*".
- * **Flusso Principale degli Eventi:** L'Utente visualizza l'admin api key.
- * **Post-condizione:** L'Utente ha visualizzato l'admin api key.

UC8.12 - Modificare admin api key

- * **Attori primari:** Utente Autenticato.
- * **Pre-condizione:** L'Utente si trova nella sezione per la visualizzazione dei dati di un singolo *Agent* aggiunto mediante modalità "instance *Agent*".
- * **Flusso Principale degli Eventi:** L'Utente modifica l'admin api key.
- * **Post-condizione:** L'Utente ha modificato l'admin api key.

UC8.13 - Visualizzazione label

- * **Attori primari:** Utente Autenticato.
- * **Pre-condizione:** L'Utente si trova nella sezione per la visualizzazione dei dati di un singolo *Agent* aggiunto mediante modalità "instance *Agent*".
- * **Flusso Principale degli Eventi:** L'Utente visualizza la label.
- * **Post-condizione:** L'Utente ha visualizzato la label.

UC8.14 - Modificare label

- * **Attori primari:** Utente Autenticato.
- * **Pre-condizione:** L'Utente si trova nella sezione per la visualizzazione dei dati di un singolo *Agent* aggiunto mediante modalità "instance *Agent*".
- * **Flusso Principale degli Eventi:** L'Utente modifica la label.
- * **Post-condizione:** L'Utente ha modificato la label.

UC8.15 - Visualizzare image URL

- * **Attori primari:** Utente Autenticato.
- * **Pre-condizione:** L'Utente si trova nella sezione per la visualizzazione dei dati di un singolo *Agent* aggiunto mediante modalità "instance *Agent*".
- * **Flusso Principale degli Eventi:** L'Utente visualizza l'immagine URL.
- * **Post-condizione:** L'Utente ha visualizzato l'immagine URL.

UC8.16 - Modificare image URL

- * **Attori primari:** Utente Autenticato.
- * **Pre-condizione:** L'Utente si trova nella sezione per la visualizzazione dei dati di un singolo *Agent* aggiunto mediante modalità "instance *Agent*".
- * **Flusso Principale degli Eventi:** L'Utente modifica l'immagine URL.
- * **Post-condizione:** L'Utente ha modificato l'immagine URL.

UC8.17 - Visualizzazione seed

- * **Attori primari:** Utente Autenticato.
- * **Pre-condizione:** L'Utente si trova nella sezione per la visualizzazione dei dati di un singolo *Agent* aggiunto mediante modalità "instance *Agent*".
- * **Flusso Principale degli Eventi:** L'Utente visualizza il seed.
- * **Post-condizione:** L'Utente ha visualizzato il seed.

UC8.18 - Modificare seed

- * **Attori primari:** Utente Autenticato.
- * **Pre-condizione:** L'Utente si trova nella sezione per la visualizzazione dei dati di un singolo *Agent* aggiunto mediante modalità "instance *Agent*".
- * **Flusso Principale degli Eventi:** L'Utente modifica il seed.
- * **Post-condizione:** L'Utente ha modificato il seed.

UC8.19 - Visualizzazione wallet name

- * **Attori primari:** Utente Autenticato.
- * **Pre-condizione:** L'Utente si trova nella sezione per la visualizzazione dei dati di un singolo *Agent* aggiunto mediante modalità "instance *Agent*".
- * **Flusso Principale degli Eventi:** L'Utente visualizza il wallet name.
- * **Post-condizione:** L'Utente ha visualizzato il wallet name.

UC8.20 - Modificare wallet name

- * **Attori primari:** Utente Autenticato.
- * **Pre-condizione:** L'Utente si trova nella sezione per la visualizzazione dei dati di un singolo *Agent* aggiunto mediante modalità "instance *Agent*".
- * **Flusso Principale degli Eventi:** L'Utente modifica il wallet name.
- * **Post-condizione:** L'Utente ha modificato il wallet name.

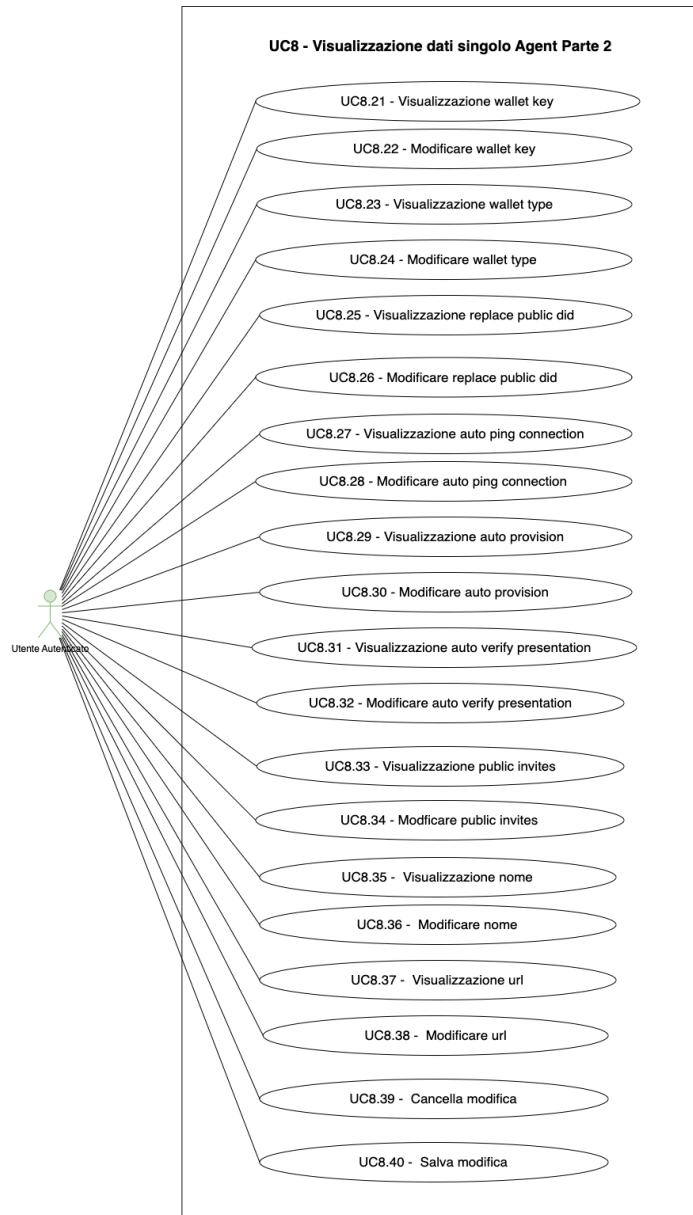


Figura 3.21: Diagramma UC8 Parte 2 - Visualizzazione dati singolo *Agent*

UC8.21 - Visualizzazione wallet key

- * **Attori primari:** Utente Autenticato.
- * **Pre-condizione:** L'Utente si trova nella sezione per la visualizzazione dei dati di un singolo *Agent* aggiunto mediante modalità "instance *Agent*".
- * **Flusso Principale degli Eventi:** L'Utente visualizza il wallet key.
- * **Post-condizione:** L'Utente ha visualizzato il wallet key.

UC8.22 - Modificare wallet key

- * **Attori primari:** Utente Autenticato.
- * **Pre-condizione:** L'Utente si trova nella sezione per la visualizzazione dei dati di un singolo *Agent* aggiunto mediante modalità "instance *Agent*".
- * **Flusso Principale degli Eventi:** L'Utente modifica il wallet key.
- * **Post-condizione:** L'Utente ha modificato il wallet key.

UC8.23 - Visualizzazione wallet type

- * **Attori primari:** Utente Autenticato.
- * **Pre-condizione:** L'Utente si trova nella sezione per la visualizzazione dei dati di un singolo *Agent* aggiunto mediante modalità "instance *Agent*".
- * **Flusso Principale degli Eventi:** L'Utente visualizza il wallet type.
- * **Post-condizione:** L'Utente ha visualizzato il wallet type.

UC8.24 - Modificare wallet type

- * **Attori primari:** Utente Autenticato.
- * **Pre-condizione:** L'Utente si trova nella sezione per la visualizzazione dei dati di un singolo *Agent* aggiunto mediante modalità "instance *Agent*".
- * **Flusso Principale degli Eventi:** L'Utente modifica il wallet type.
- * **Post-condizione:** L'Utente ha modificato il wallet type.

UC8.25 - Visualizzazione replace public did

- * **Attori primari:** Utente Autenticato.
- * **Pre-condizione:** L'Utente si trova nella sezione per la visualizzazione dei dati di un singolo *Agent* aggiunto mediante modalità "instance *Agent*".
- * **Flusso Principale degli Eventi:** L'Utente visualizza il replace public did.
- * **Post-condizione:** L'Utente ha visualizzato il replace public did.

UC8.26 - Modificare replace public did

- * **Attori primari:** Utente Autenticato.
- * **Pre-condizione:** L'Utente si trova nella sezione per la visualizzazione dei dati di un singolo *Agent* aggiunto mediante modalità "instance *Agent*".
- * **Flusso Principale degli Eventi:** L'Utente modifica il replace public did.
- * **Post-condizione:** L'Utente ha modificato il replace public did.

UC8.27 - Visualizzazione auto ping connection

- * **Attori primari:** Utente Autenticato.
- * **Pre-condizione:** L'Utente si trova nella sezione per la visualizzazione dei dati di un singolo *Agent* aggiunto mediante modalità "instance *Agent*".
- * **Flusso Principale degli Eventi:** L'Utente visualizza l'auto ping connection.
- * **Post-condizione:** L'Utente ha visualizzato l'auto ping connection.

UC8.28 - Modificare auto ping connection

- * **Attori primari:** Utente Autenticato.
- * **Pre-condizione:** L'Utente si trova nella sezione per la visualizzazione dei dati di un singolo *Agent* aggiunto mediante modalità "instance *Agent*".
- * **Flusso Principale degli Eventi:** L'Utente modifica l'auto ping connection.
- * **Post-condizione:** L'Utente ha modificato l'auto ping connection.

UC8.29 - Visualizzazione auto provision

- * **Attori primari:** Utente Autenticato.
- * **Pre-condizione:** L'Utente si trova nella sezione per la visualizzazione dei dati di un singolo *Agent* aggiunto mediante modalità "instance *Agent*".
- * **Flusso Principale degli Eventi:** L'Utente visualizza l'auto provision.
- * **Post-condizione:** L'Utente ha visualizzato l'auto provision.

UC8.30 - Modificare auto provision

- * **Attori primari:** Utente Autenticato.
- * **Pre-condizione:** L'Utente si trova nella sezione per la visualizzazione dei dati di un singolo *Agent* aggiunto mediante modalità "instance *Agent*".
- * **Flusso Principale degli Eventi:** L'Utente modifica l'auto provision.
- * **Post-condizione:** L'Utente ha modificato l'auto provision.

UC8.31 - Visualizzazione auto verify presentation

- * **Attori primari:** Utente Autenticato.
- * **Pre-condizione:** L'Utente si trova nella sezione per la visualizzazione dei dati di un singolo *Agent* aggiunto mediante modalità "instance *Agent*".
- * **Flusso Principale degli Eventi:** L'Utente visualizza l'auto verify presentation.
- * **Post-condizione:** L'Utente ha visualizzato l'auto verify presentation.

UC8.32 - Modificare auto verify presentation

- * **Attori primari:** Utente Autenticato.
- * **Pre-condizione:** L'Utente si trova nella sezione per la visualizzazione dei dati di un singolo *Agent* aggiunto mediante modalità "instance *Agent*".
- * **Flusso Principale degli Eventi:** L'Utente modifica l'auto verify presentation.
- * **Post-condizione:** L'Utente ha modificato l'auto verify presentation.

UC8.33 - Visualizzazione public invites

- * **Attori primari:** Utente Autenticato.
- * **Pre-condizione:** L'Utente si trova nella sezione per la visualizzazione dei dati di un singolo *Agent* aggiunto mediante modalità "instance *Agent*".
- * **Flusso Principale degli Eventi:** L'Utente visualizza il public invites.
- * **Post-condizione:** L'Utente ha visualizzato il public invites.

UC8.34 - Modificare public invites

- * **Attori primari:** Utente Autenticato.
- * **Pre-condizione:** L'Utente si trova nella sezione per la visualizzazione dei dati di un singolo *Agent* aggiunto mediante modalità "instance *Agent*".
- * **Flusso Principale degli Eventi:** L'Utente modifica il public invites.
- * **Post-condizione:** L'Utente ha modificato il public invites.

UC8.35 - Visualizzazione nome

- * **Attori primari:** Utente Autenticato.
- * **Pre-condizione:** L'Utente si trova nella sezione per la visualizzazione dei dati di un singolo *Agent* aggiunto mediante modalità "URL *Agent*".
- * **Flusso Principale degli Eventi:** L'Utente visualizza il nome.
- * **Post-condizione:** L'Utente ha visualizzato il nome.

UC8.36 - Modificare nome

- * **Attori primari:** Utente Autenticato.
- * **Pre-condizione:** L'Utente si trova nella sezione per la visualizzazione dei dati di un singolo *Agent* aggiunto mediante modalità "URL *Agent*".
- * **Flusso Principale degli Eventi:** L'Utente modifica il nome.
- * **Post-condizione:** L'Utente ha modificato il nome.

UC8.37 - Visualizzazione URL

- * **Attori primari:** Utente Autenticato.
- * **Pre-condizione:** L'Utente si trova nella sezione per la visualizzazione dei dati di un singolo *Agent* aggiunto mediante modalità "URL *Agent*".
- * **Flusso Principale degli Eventi:** L'Utente visualizza l'URL.
- * **Post-condizione:** L'Utente ha visualizzato l'URL.

UC8.38 - Modificare URL

- * **Attori primari:** Utente Autenticato.
- * **Pre-condizione:** L'Utente si trova nella sezione per la visualizzazione dei dati di un singolo *Agent* aggiunto mediante modalità "URL *Agent*".
- * **Flusso Principale degli Eventi:** L'Utente modifica l'URL.
- * **Post-condizione:** L'Utente ha modificato l'URL.

UC8.39 - Cancella modifica

- * **Attori primari:** Utente Autenticato.
- * **Pre-condizione:** L'Utente si trova nella sezione per la visualizzazione dei dati di un singolo *Agent*.
- * **Flusso Principale degli Eventi:** L'Utente preme il bottone "Cancel" per annullare le modifiche apportate all'*Agent Agent* con la conseguente chiusura della *dialog*.
- * **Post-condizione:** L'Utente ha annullato le modifiche apportate all'*Agent* e si trova nuovamente nella pagina con la lista degli *Agent* salvati.

UC8.40 - Salva modifica

- * **Attori primari:** Utente Autenticato.
- * **Pre-condizione:** L'Utente si trova nella sezione per la visualizzazione dei dati di un singolo *Agent*.
- * **Flusso Principale degli Eventi:** L'Utente preme il bottone "Save" per salvare le modifiche apportate all'*Agent Agent* con la conseguente chiusura della *dialog*.
- * **Post-condizione:** L'Utente ha salvato le modifiche apportate all'*Agent* e si trova nuovamente nella pagina con la lista degli *Agent* salvati.

3.3 Tracciamento dei requisiti

Ogni requisito viene definito nel seguente modo:

- * Codice identificativo.
- * Descrizione.
- * Caso d'uso.

Il codice identificativo prevede la seguente sintassi:

R [Tipologia] - [Identificativo]

dove:

- * **R**: identifica che si tratta di un requisito.
- * **Priorità**: grado di priorità, può assumere i valori:
 - **A**: obbligatorio, necessario;
 - **B**: desiderabile, non strettamente necessario;
 - **C**: opzionale, non necessario.
- * **Tipologia**: tipo di requisito, può assumere i valori:
 - **F**: funzionale;
 - **P**: prestazionale;
 - **Q**: qualitativo;
 - **V**: vincolo.
- * **ID**: numero progressivo.

Nelle tabelle 3.1 e 3.2 sono riassunti i requisiti e il loro tracciamento con gli use case delineati in fase di analisi e riportati nella Sezione 3.2.

Si fa notare che tutti i vincoli individuati sono funzionali o di vincolo e di priorità "A" o "B" .

Tabella 3.1: Tabella del tracciamento dei requisiti funzionali

Requisito	Descrizione	Use Case
RAF-1	L'interfaccia deve permettere la visualizzazione della lista delle <i>Pool</i>	UC1
RAF-2	L'interfaccia deve permettere di cancellare una <i>Pool</i> dalla lista	UC2
RAF-3	L'interfaccia deve permettere l'aggiunta di una nuova <i>Pool</i>	UC3
RAF-4	L'interfaccia deve permettere l'inserimento dei dati per l'aggiunta di una <i>Pool</i>	UC3.1, UC3.2, UC3.3, UC3.4, UC3.4.3
RAF-5	L'interfaccia deve permettere l'inserimento dei valori del nuovo token per la personalizzazione dell' <i>Agent</i>	UC3.3.2, UC3.3.3, UC4.6.2, UC4.6.3
RAF-6	L'interfaccia deve permettere l'annullamento di aggiunta di una nuova <i>Pool</i>	UC3.5
RAF-7	L'interfaccia deve permettere di confermare l'aggiunta di una nuova <i>Pool</i>	UC3.6
RAF-8	L'interfaccia deve permettere di visualizzare i dati di ogni singola <i>Pool</i>	UC4
RAF-9	L'interfaccia deve permettere di visualizzare il nome, l' <i>Agent</i> URL e i vari token di personalizzazione della <i>Pool</i> selezionata	UC4.1, UC4.3, UC4.5
RAF-10	L'interfaccia deve permettere di modificare il nome, l' <i>Agent</i> URL e i vari token di personalizzazione della <i>Pool</i> selezionata	UC4.2, UC4.4, UC4.6
RAF-11	L'interfaccia deve permettere di Aggiungere altri token per la personalizzazione dell' <i>Agent</i>	UC3.3.1, UC4.6.1,
RAF-12	L'interfaccia deve permettere di modificare i vari token per la personalizzazione dell' <i>Agent</i>	UC4.6.4, UC4.6.5
RAF-13	L'interfaccia deve permettere di cancellare i vari token inseriti per la personalizzazione dell' <i>Agent</i>	UC3.3.4, UC4.6.6
RAF-14	L'interfaccia deve permettere di visualizzare il Genesis Txn della <i>Pool</i> selezionata in formato Json	UC4.8
RAF-15	L'interfaccia deve permettere di cancellare le modifiche effettuate alla <i>Pool</i> selezionata	UC4.9
RAF-16	L'interfaccia deve permettere di salvare le modifiche effettuate alla <i>Pool</i> selezionata	UC4.10
RBF-1	L'interfaccia mette a disposizione la funzionalità di poter visualizzare il Genesis Txn delle <i>Pool</i> selezionata in formato tabellare	UC4.7
RBF-2	L'interfaccia mette a disposizione la funzionalità di importare un file <i>.txt</i> o <i>.json</i> per l'inserimento del Genesis Txn	UC3.4.1, UC3.4.2
RBF-3	L'interfaccia mette a disposizione la funzionalità di ricerca per filtrare i dati della tabella	UC4.7.1
RBF-4	L'interfaccia mette a disposizione la funzionalità di selezione dei dati da visualizzare nella tabella	UC-4.7.2
RBF-5	L'interfaccia mette a disposizione la funzionalità di aprire una nuova pagina per visualizzare tutti i dati della riga della tabella selezionata sia in formato tabellare che in formato Json	UC4.7.3, UC4.7.3.1, UC4.7.3.2

Tabella 3.2: Tabella del tracciamento dei requisiti di vincolo

Requisito	Descrizione
RAV-1	Lo sviluppo del front-end deve avvenire tramite il <i>framework</i> Vuejs
RAV-2	Lo sviluppo delle componenti che compongono le pagine deve avvenire tramite il <i>material design framework</i> Vuetify
RAV-3	Il sistema deve tradurre automaticamente le parole nella lingua preselezionata dal browser (se non presente quella di default sarà l'inglese) attraverso il <i>plugin</i> vuei18n
RAV-4	Il sistema deve validare i campi obbligatori se vuoti tramite la libreria vee-validate
RAV-5	Il sistema deve essere accessibile
RAV-6	Lo sviluppo del front-end deve essere consistente a quello già esistente all'interno della Web App
RBV-1	Il sistema deve gestire le richieste all' API attraverso la libreria axios

Capitolo 4

Tecnologie e strumenti

In questo capitolo vengono descritte le tecnologie e gli strumenti utilizzati all'interno dello stage.

4.1 Tecnologie

Di seguito viene data una panoramica delle tecnologie utilizzate

4.1.1 JavaScript

JavaScript è un linguaggio di programmazione multi paradigma orientato agli eventi, comunemente utilizzato nella programmazione *Web* lato *client*, per la creazione di effetti dinamici interattivi. I file con estensione ".js" sono scritti con il linguaggio [Javascript](#) [9].

4.1.2 VueJS

Vue.js è un *framework* JavaScript open-source in configurazione Model-View-ViewModel per la creazione di applicazioni web dinamiche grazie a una serie di funzionalità e strumenti forniti dallo stesso. L'architettura modulare consente di strutturare al meglio un'applicazione e di avere un elevato riutilizzo del codice. Permette inoltre un'elevata manutenibilità in quanto ogni componente è adibito ad un'unica funzione [21].

4.1.3 Nuxtjs

Nuxtjs è un *framework* frontend basato su Vuejs che offre ottime funzionalità di sviluppo come il rendering lato server, percorsi generati automaticamente, una migliore gestione dei meta tag e un miglioramento [SEO](#)_g creando così una notevole riduzione dei tempi di interattività [15].

4.1.4 Vuetify

Vuetify è un *material design framework* basato su vuejs che ha come obiettivo quello di fornire agli sviluppatori gli strumenti necessari per creare esperienze utente ricche e coinvolgenti. Vuetify mette a disposizione una serie di [UI components](#) e altre risorse per aiutare lo sviluppatore a creare un *frontend* completo, accessibile ed armonioso [23].

4.1.5 Vee-validate

VeeValidate è una libreria di validazione per Vuejs. Ha molte regole di validazione pronte all'uso e supporto anche per quelle personalizzate. È basato su modelli, quindi è simile e ha familiarità con l'API di validazione HTML5. Puoi validare input HTML5 e componenti Vue personalizzati [19].

4.1.6 Vuei18n

Vuei18n è il plugin di internazionalizzazione di Vue.js e permette di effettuare la traduzione della pagina in base alla lingua predefinita del browser [22].

4.1.7 Axios

Axios è una libreria HTTP che viene eseguita sia su un client che su un server, che effettua una richiesta API, esegue il compito di produrre il risultato e specifica concetti più semplici da leggere ed esegue il debug [2].

4.2 Strumenti

Di seguito viene data una panoramica degli strumenti utilizzati

4.2.1 Visual Studio Code

Visual Studio Code è un editor di codice sorgente, particolarmente comodo nell'utilizzo in quanto include il supporto per *debugging*, un controllo per *Git* integrato, *syntax highlighting*, *IntelliSense*, *snippet* e *refactoring* del codice [20].

4.2.2 Figma

Figma è un editor di grafica vettoriale e uno strumento di prototipazione. È un'applicazione di progettazione dell'interfaccia utente e dell'esperienza utente, con particolare attenzione alla collaborazione in tempo reale [6].

4.2.3 YouTrack Administration

YouTrack è uno strumento per gestire progetti, configurare *agile boards*, mantenere basi di conoscenza, utilizzare report e diagrammi di Gantt, dashboard e monitoraggio del tempo. Tiene traccia di attività e problemi, supporta le richieste dei clienti, pianifica sprint e rilasci e crea flussi di lavoro [24].

4.2.4 GitHub

GitHub è un servizio di hosting per progetti software e viene principalmente utilizzato dagli sviluppatori, che caricano il codice sorgente dei loro programmi e lo rendono scaricabile dagli utenti. Questi ultimi possono interagire con lo sviluppatore tramite un sistema di issue tracking, pull request e commenti che permette di migliorare il codice del repository risolvendo bug o aggiungendo funzionalità. Inoltre Github elabora dettagliate pagine che riassumono come gli sviluppatori lavorano sulle varie versioni dei repository [7].

4.2.5 Microsoft Teams

Microsoft Teams è una piattaforma di comunicazione e collaborazione unificata che combina chat di lavoro persistente, teleconferenza, condivisione di contenuti incluso lo scambio e il lavoro simultaneo sui file e integrazione delle applicazioni [12].

Capitolo 5

Progettazione e codifica

In questo capitolo vengono descritte le fasi di progettazione e codifica relativi all'applicativo sviluppato.

5.1 Progettazione

Si presenta ora l'architettura del *framework* Vuejs, dell'applicativo sviluppato e i *design pattern* utilizzati.

5.1.1 Architettura di Vuejs

Come mostrato in Figura 5.1, in Vuejs il progetto è composto principalmente da file con estensione *.vue*. Essi sono l'unità principale e vengono salvati in diverse *folder* in base al loro ruolo. L'Architettura principale di Vuejs è suddivisa in:

- * **components:** I components sono istanze Vue riutilizzabili con elementi HTML personalizzati. I componenti possono essere riutilizzati tutte le volte che si vuole o utilizzati in un altro componente, rendendolo un componente figlio. Una pagina web è composta da diversi componenti.
- * **pages:** Le pages sono costruite partendo dai vari componenti e sono la pagina effettiva dove l'utente visualizza od interagisce con il contenuto.
- * **layouts:** I layouts sono le parti dell'interfaccia che non cambiano mai e che si ripetono in più pagine. Un esempio classico di layout è il menu che rimane sempre uguale per tutte l'interfacce.

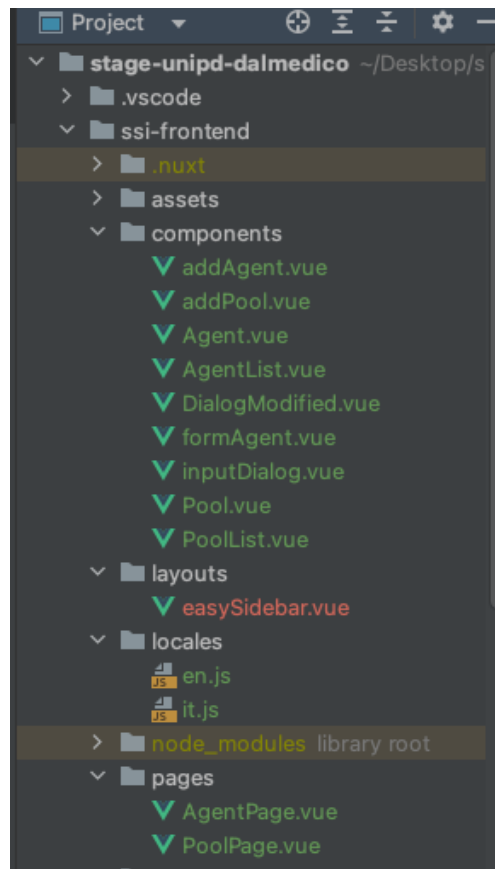


Figura 5.1: Struttura progetto

Ogni file con estensione `.vue` è formato da 3 *tag* principali rendendo molto facile, veloce ed intuitiva l'implementazione, la modifica e la manutenzione. Essi sono dunque suddivisi in:

- * **<template>**: Dove al suo interno troviamo tutti i vari tag HTML che compongono la struttura del file.
- * **<script>**: Dove al suo interno troviamo tutto il codice, i metodi e le variabili `Javascript` che servono per rendere la pagina dinamica e far interagire l'Utente.
- * **<style>**: Dove al suo interno troviamo il codice CSS per modificare l'aspetto e la grafica della pagina.

Questa struttura dei file rende molto facile, veloce ed intuitiva l'implementazione, la modifica e la manutenzione.

5.1.2 Architettura dell'applicazione sviluppata

L'architettura dell'applicazione è stata progettata seguendo le *best practise* già consolidate nell'ambiente di Vuejs [3]. L'utilizzo di questo *framework* ha semplificato la creazione di un codice altamente modulare e mantenibile, la sua architettura infatti guida lo sviluppatore nella creazione di un programma che rispetti tali caratteristiche.

Come si può vedere nella Figura 5.1, tutti i *component* sono situati nell'apposita cartella *components* e lo stesso vale anche per le *pages* e i *layouts*. Per lo spostamento tra le pagine, è stato scelto di sfruttare il *routing*, una tecnica che permette di modificare quello che l'utente vede senza cambiare pagina, ma mostrando o nascondendo parti che corrispondono a determinati *component* per permettere lo spostamento rapido tra diverse *view*. All'interno della cartella *layouts* si trova il file `easySidebar.vue` dove nel quale viene definito il menu e l'header che verranno utilizzati in tutto l'applicativo. Nella cartella *components* invece vi sono tutti i componenti che verranno utilizzati all'interno dell'applicativo. I vari componenti sono:

- * **PoolList.vue**: in questo file si trova il codice per la realizzazione della lista di *card* dove ognuna di esse rappresenta una delle *Pool* salvate.
- * **DialogModified.vue**: L'applicativo è stato ideato in modo tale da usare le *dialog* come elemento base per contenere le interazioni principali tra utente e [Web App](#). Questo file contiene il codice per la creazione di una *Dialog* standard dove al suo interno verranno inseriti i vari componenti in base alle necessità.
- * **addPool**: in questo file è contenuto il codice per l'aggiunta di una nuova *Pool* alla lista di quelle già salvate.
- * **Pool**: in questo file è contenuto tutto il codice per la visualizzazione dei dati della *Pool* selezionata dalla lista.
- * **inputDialog**: in questo file vi è una parte del Form input che rimane uguale sia per l'aggiunta e sia per la visualizzazione di una *Pool*.
- * **AgentList**: in questo file vi è contenuto il codice per l'implementazione dell'interfaccia dove si può visualizzare la lista di *card* degli *Agent* già salvati.
- * **addAgent**: in questo file è contenuto il codice per l'aggiunta di una nuova *Agent* alla lista di quelle già salvate.
- * **Agent**: in questo file è contenuto tutto il codice per la visualizzazione dei dati della *Agent* selezionata dalla lista.

Infine, nella cartella *pages* troviamo due file, rispettivamente:

- * **PoolPage**: la pagina principale della sezione delle *Pool*.
- * **AgentPage**: la pagina principale della sezione degli *Agent*.

5.2 Design Pattern utilizzati

MVC Pattern

Il classico [Model View Controller \(MVC\)](#)_g consiste in un'architettura costituita da tre parti distinte [13]:

- * **Model**: interfacce e classi senza logica;
- * **View**: user interface;
- * **Controller**: uniscono i due layer precedenti.

Vuejs intrinsecamente sfrutta un *design pattern* leggermente variato e semplificato che si accosta ad un **Model View ViewModel (MVVM)**_g [14] che, come mostrato in Figura 5.2, permette un *data binding* a due direzioni tra "View" e "ViewModel".

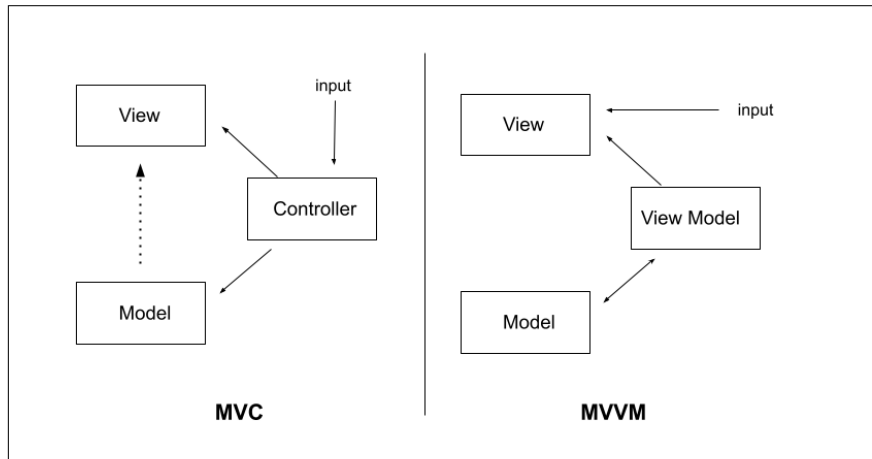


Figura 5.2: Struttura MVC e MVVM

Decorator

Il *design pattern* Decorator permette l'aggiunta di nuove funzionalità ad oggetti o classi già esistenti, senza modificare il codice sorgente. Questo è reso possibile da una classe "decoratore" alla quale viene passato come parametro l'oggetto originale [4]. Questo *pattern* differisce dalla classica ereditarietà, che non permette la modifica, aggiunta o rimozione delle funzionalità a *run-time*.

5.3 Codifica

Si presentano ora nello specifico i diversi file descrivendo i vari componenti e motivando le scelte effettuate.

Si vuole far notare che per il design e la UX di base si è utilizzato quello già in uso nel progetto Monokee e che, prima di fare la codifica di ogni singolo componente, si doveva passare per l'approvazione dei responsabili del progetto presentando i disegni (effettuati con *Figma*), commentando e giustificando le scelte effettuate.

5.3.1 Layout

Come anticipato nella sezione 5.1.1, la cartella *layouts* contiene tutte quelle parti dell'interfaccia che rimangono statici all'interno dell'intera applicazione.

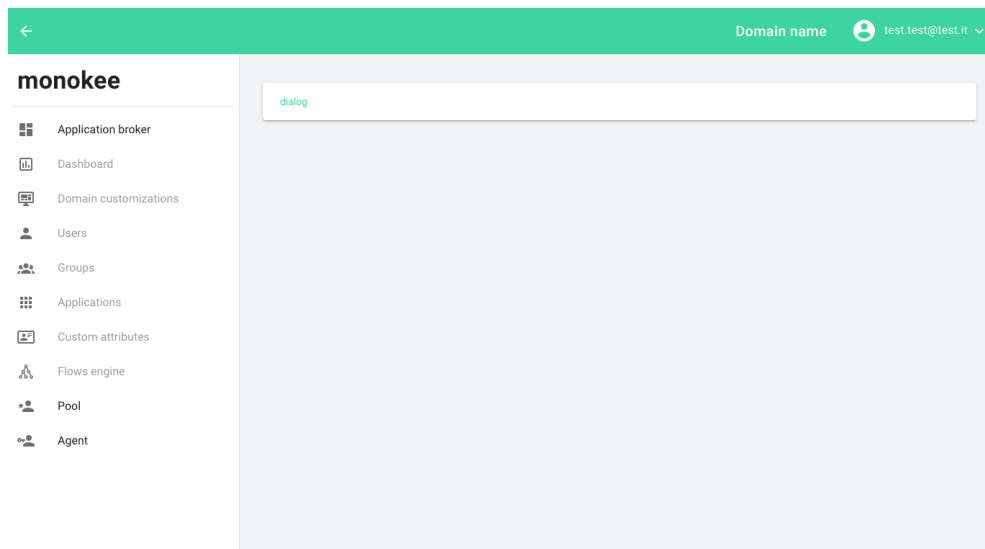


Figura 5.3: layout

In questo progetto all'interno della cartella *layouts* è presente un unico file denominato *easySideBar.vue*. Come si vede in Figura 5.3, è composto da un menu verticale a sinistra espandibile per la navigazione tra le pagine (dove vi sono stati inserite alcune voci statiche con il solo scopo di far capire come si presenterebbe l'applicazione conclusa) e da un header con il *Domain name* e un ulteriore *drop down* menu a tendina per le varie impostazioni e per il logout.

5.3.2 Components

Nella cartella "Components" sono contenuti tutti i componenti dell'applicativo, ognuno di essi svolge un particolare ruolo nella la formazione del sistema completo. Analizziamo ora nel dettaglio ogni *component* presente.

5.3.2.1 Dialog Modified

Il *component* "Dialog Modified" da origine a una dialog standard all'interno dell'applicativo dove al suo interno, attraverso diversi *tag "slot"* di *vuejs* vengono inseriti i vari *component* necessari nelle diverse circostanze.

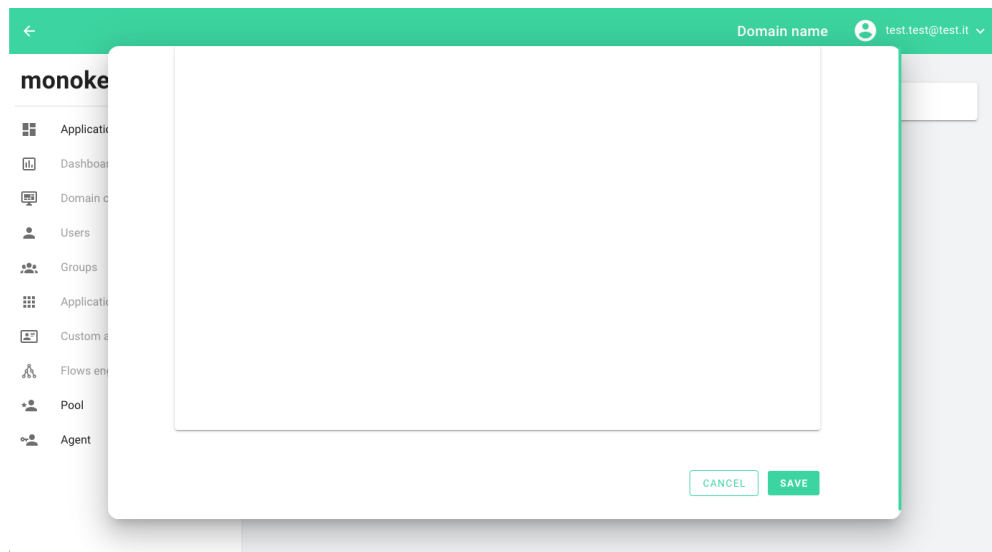


Figura 5.4: Dialog Component

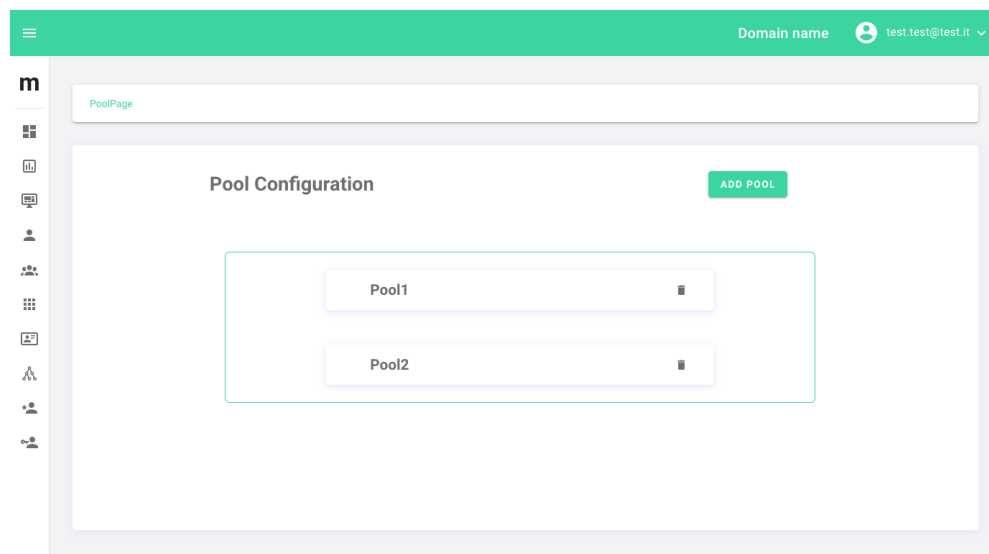
Come si vede dalla Figura 5.4 la *dialog* ha una dimensione fissata all'80% dello schermo per permettere così di visualizzare i componenti e i dati al suo interno senza stressare la vista ma allo stesso tempo facendo capire che non è una nuova pagina. all'interno del file *DialogModified.vue* sono presenti due *slot*:

- * **activator**: dove verranno inseriti i *components* e le azioni per far aprire la dialog
- * **content**: dove verranno inseriti i *components* che verranno visualizzati all'interno della *dialog*

La scelta di creare un *component* Dialog Modified è avvenuta in quanto le *dialog* sono uno degli elementi principali all'interno del progetto Monokee e quindi, essendo utilizzati in larga scala si è deciso di "standardizzare" tale componente.

5.3.2.2 Pool List

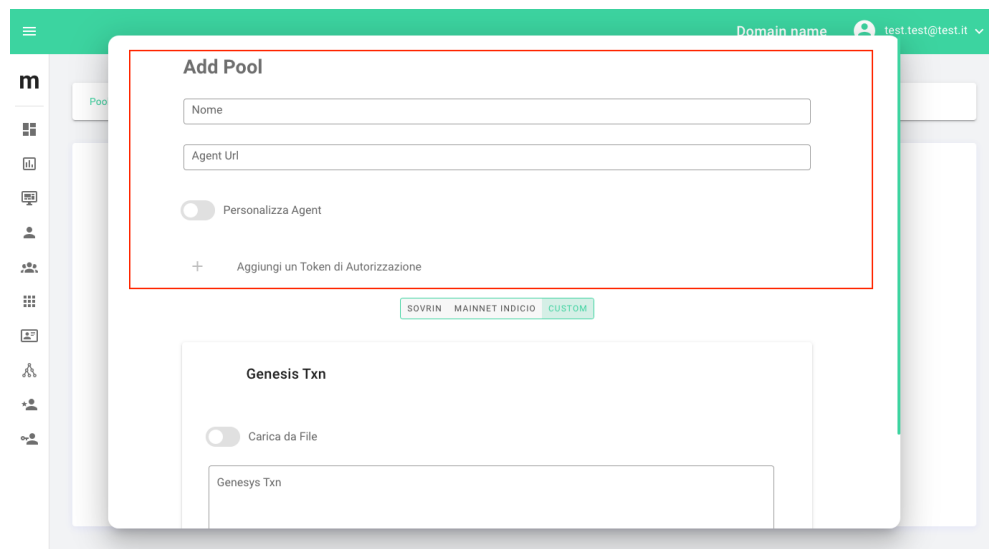
Il *component* "Pool List" dà origine a una lista di *card* dove ognuna di esse rappresenta una delle *Pool* salvate.

**Figura 5.5:** Pool List Component

Come si vede dalla Figura 5.5 la lista di *card* è in verticale e si trova una sola *card* per riga. In ogni *card* si può visualizzare il nome della *Pool* e un'icona a forma di cestino per la sua cancellazione dalla lista.

5.3.2.3 Input Dialog

Il *component* "Input Dialog" dà origine a una serie di *text field* per la compilazione, la visualizzazione o la modifica dei dati.

**Figura 5.6:** Input Dialog Component

Come si vede dalla Figura 5.6 all'interno del rettangolo rosso il *component* presenta una serie di *text field* che servono per l'inserimento o la visualizzazione rispettivamente

del nome e dell' *Agent* url. In aggiunta c'è anche uno *switch* per selezionare se si vuole oppure no personalizzare l'Agent. In caso affermativo si attiverà il *button* per l'aggiunta di due nuove *text field* per l'inserimento del nome e del valore del nuovo Token. Il sistema permette di creare quanti token di personalizzazione si vuole e anche di rimuoverli cliccando l'icona a forma di cestino presente per ogni token alla fine della propria riga. La scelta di creare un *component* apposito per questa lista di input si spiega in quanto essa sia presente, seppur per motivi diversi, sia nella *dialog* per l'aggiunta di una nuova *Pool* sia in quella per la visualizzazione delle *Pool* salvate. Essendo che entrambe le *dialog* presentano questa lista di input si è deciso dunque di creare un *component* unico.

5.3.2.4 Add Pool

Il *component* "Add Pool" dà origine a un *button* con titolo "Add Pool" che, se cliccato, apre una *dialog* con tutti i *component* per l'inserimento dei dati necessari per la creazione di una nuova *Pool*

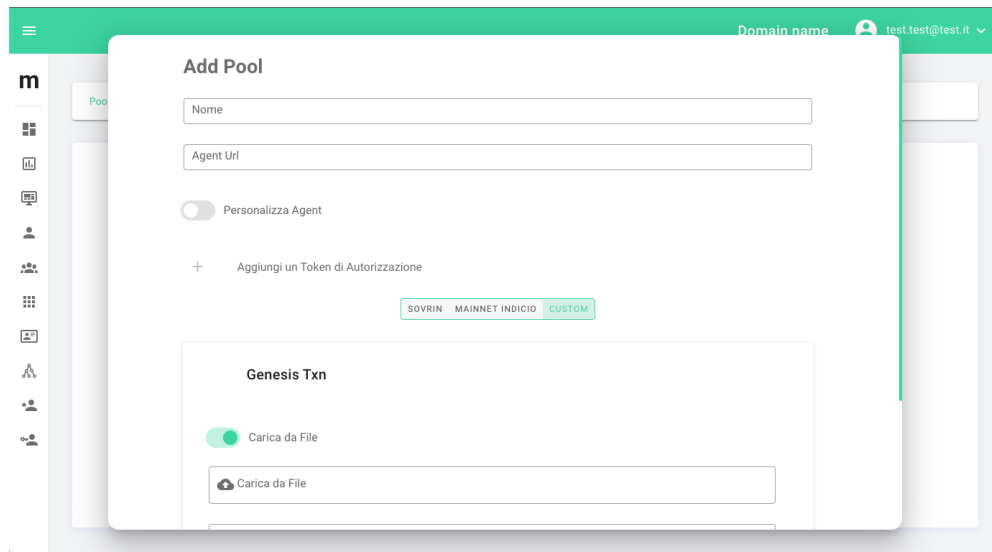


Figura 5.7: Add Pool Component - 1

Come mostrato in Figura 5.7 all'interno della *dialog* troviamo un form da compilare in cui in primo luogo si può notare il *component* "Input Dialog" e per la sua descrizione si rimanda alla sezione 5.3.2.3.

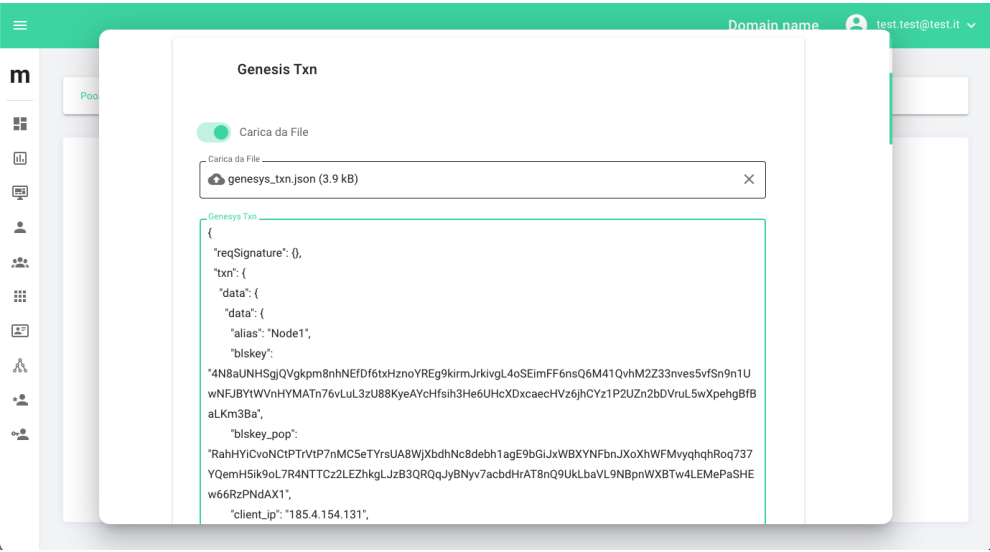


Figura 5.8: Add Pool Component - 2

Il resto del form (Figura 5.8) è strutturato all'interno di una card con titolo "Genesis Txn" dentro al quale si trova uno *switch* per poter decidere se riempire il *text field* importando un file *.txt* o *.json* dal dispositivo o compilarlo a mano. Da far notare che se si sceglie di importare un file comparirà, prima del *text field*, un *file input* dove, una volta importato il file, permetterà di visualizzare il nome del file e la sua dimensione.

5.3.2.5 Pool

Il *component* "Pool" da origine a ciascuna *card* della lista (per la descrizione delle *card* si rimanda alla sezione 5.5). Una volta cliccata una delle *card* si aprirà una *dialog* per la visualizzazione o la modifica dei dati della *Pool* salvata.

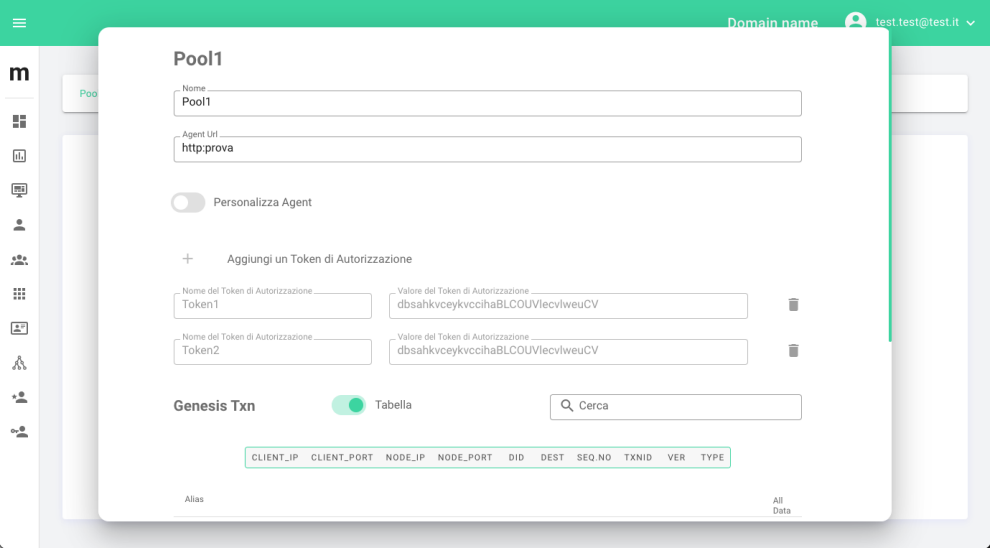


Figura 5.9: Pool - 1

Come mostrato in Figura 5.9 all'interno della *dialog* troviamo un form già compilato con i dati della *Pool* selezionata. In primo luogo in questo form si può notare il *component* "Input Dialog" e per la sua descrizione si rimanda alla sezione 5.3.2.3.

The screenshot shows a web application interface with a sidebar on the left and a main content area. A modal dialog titled "Genesis Txn" is open. At the top of the dialog, there are two input fields for authorization tokens. Below these, there is a toggle switch labeled "Tabella" which is currently turned on. A search bar with the placeholder "Cerca" is also present. The main part of the dialog is a table with the following columns: CLIENT_IP, CLIENT_PORT, NODE_IP, NODE_PORT, DID, DEST, SEQ_NO, TXNID, VER, TYPE. The table contains four rows of data for Node1, Node2, Node3, and Node4. At the bottom of the dialog, there are "CANCEL" and "SAVE" buttons.

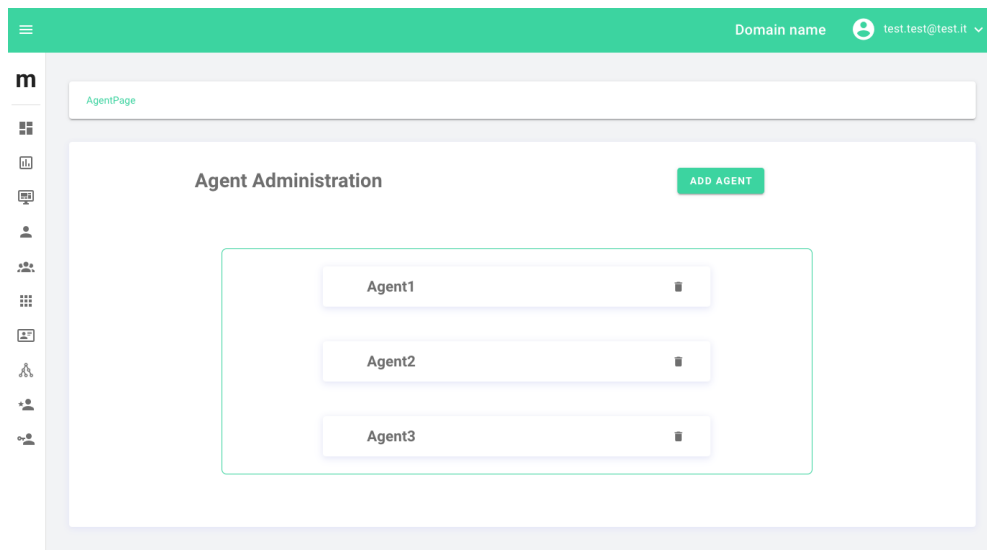
	CLIENT_IP	CLIENT_PORT	NODE_IP	NODE_PORT	DID	DEST	SEQ_NO	TXNID	VER	TYPE
Alias	client_ip				DID				ver	type
Node1	185.4.154.131				Th7MpTaRZVRmPiabds81Y				1	0
Node2	185.4.154.131				EbP4aYNeTHL6q385GuVpRV				1	0
Node3	185.4.154.131				4cU41vWW82ArfxJxHkzXPG				1	0
Node4	185.4.154.131				TWwCRQRZ2ZHMJFn9TzLp7W				1	0

Figura 5.10: Pool - 2

Il resto del form (Figura 5.10) è strutturato all'interno di una card con il titolo "Genesis Txn" dentro al quale si trova uno *switch* per poter decidere se visualizzare i dati in forma tabellare (l'opzione di default) o in formato *Json*. Se si sceglie di visualizzare i dati in formato *Json* comparirà una *text area* in sola lettura in cui al suo interno sarà possibile vedere i dati nel loro formato naturale. Se si sceglie di visualizzare i dati in formato tabellare comparirà in alto a destra all'interno della *card* una barra di ricerca per filtrare i dati all'interno della tabella. Successivamente si può notare una serie di *toggle button* per selezionare quali dati visualizzare all'interno della tabella. Questa scelta è stata effettuata in quanto il grande numero e la dimensione dei dati avrebbe reso difficile la lettura della tabella obbligando l'utente a fare degli swipe a destra e sinistra all'interno della tabella stessa. Alla fine di ogni riga della tabella si trova un *icon* a forma di occhio che se cliccato permetterà l'apertura di una nuova finestra in cui sarà possibile visualizzare tutti i dati della singola riga. Anche in questo caso si è deciso di dare la possibilità, attraverso uno *switch*, di decidere se visualizzare i dati in forma tabellare o in formato *Json*. La scelta di permettere la visualizzazione dei dati in due formati differenti non è casuale ed è stata decisa in quanto, parlando con i responsabili del progetto monokee, è uscito che facendo il supporto tecnico ai clienti è capitato di avere la necessità sia vedere i dati in formato tabellare per una maggior velocità di lettura ma anche di avere la possibilità di avere a disposizione il codice *Json* per poterlo copiare e incollare in altre applicazioni.

5.3.2.6 Agent List

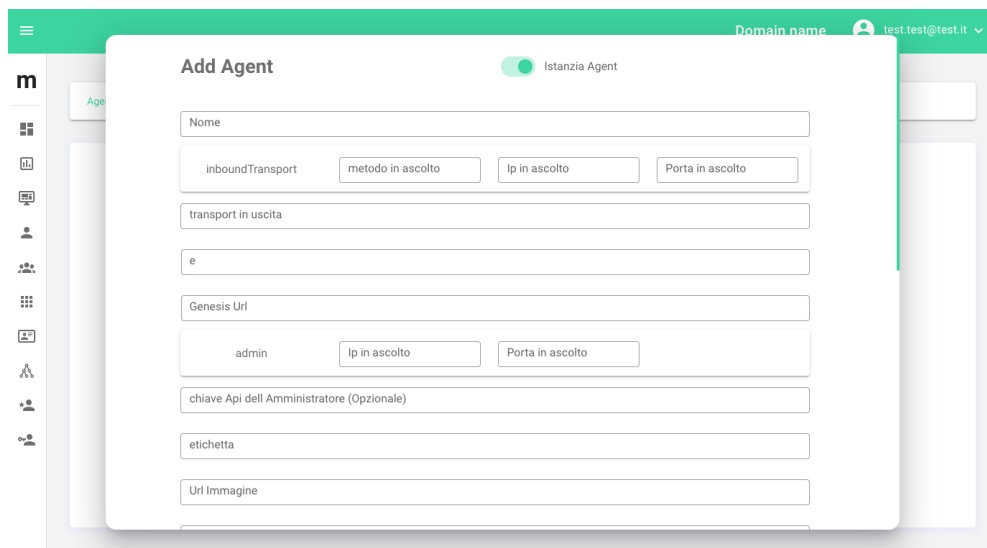
Il *component* "Agent List", come il "Pool List", da origine a una lista di *card* dove ognuna di esse rappresenta uno degli *Agent* salvati.

**Figura 5.11:** Agent List Component

Come si vede dalla Figura 5.11 la lista di *card* è in verticale e si trova una sola card per riga. In ogni *card* si può visualizzare il nome dell' *Agent* e un'icona a forma di cestino per la sua cancellazione dalla lista.

5.3.2.7 Add Agent

Il *component* "Add Agent" dà origine a un *button* con titolo "Add Agent" che, se cliccato, apre una dialog con tutti i *component* per l'inserimento dei dati necessari per la creazione di un nuovo *Agent*.

**Figura 5.12:** Add Agent Component - 1

Nome Portafoglio

Chiave Portafoglio

☒ Sostituisci il DID pubblico

☒ Fornitura automatica

☒ Connessione con ping automatico

☒ Verifica automatica della presentazione

☐ Inviti pubblici

CANCEL ADD

Figura 5.13: Add *Agent* Component - 2

Come mostrato nelle Figure 5.12 e 5.13 all'interno della *dialog* troviamo un form da compilare che è strutturato all'interno di una card con titolo "Add Agent" dentro al quale si trova uno *switch* per poter decidere se istanziare il nuovo *Agent* inserendo i dati manualmente o importando un *Agent* già esistente inserendo il nome e il suo URL (Figura 5.14).

Add Agent

☐ Url Agent

Nome

Url

CANCEL ADD

Figura 5.14: Add *Agent* Component - 3

Da far notare che nel caso si decidesse di istanziare l'*Agent* manualmente il form è costituito da *Text Field* per i dati da inserire e da *switch* per le impostazioni booleane per la configurazione. Per facilitare l'inserimento dei dati dei campi *inbound Transport*

e dell' *admin* si è deciso di creare una *card* che contenesse al suo interno, sulla stessa riga, i vari *Text Field* dei campi che costituiscono i due tag (Figura 5.12).

5.3.2.8 Agent

Il *component* "Agent" da origine a ciascuna *card* della lista (per la descrizione delle *card* si rimanda alla sezione 5.3.2.6). Una volta cliccata una delle *card* si aprirà una *dialog* per la visualizzazione o la modifica dei dati dell' *Agent* salvato.

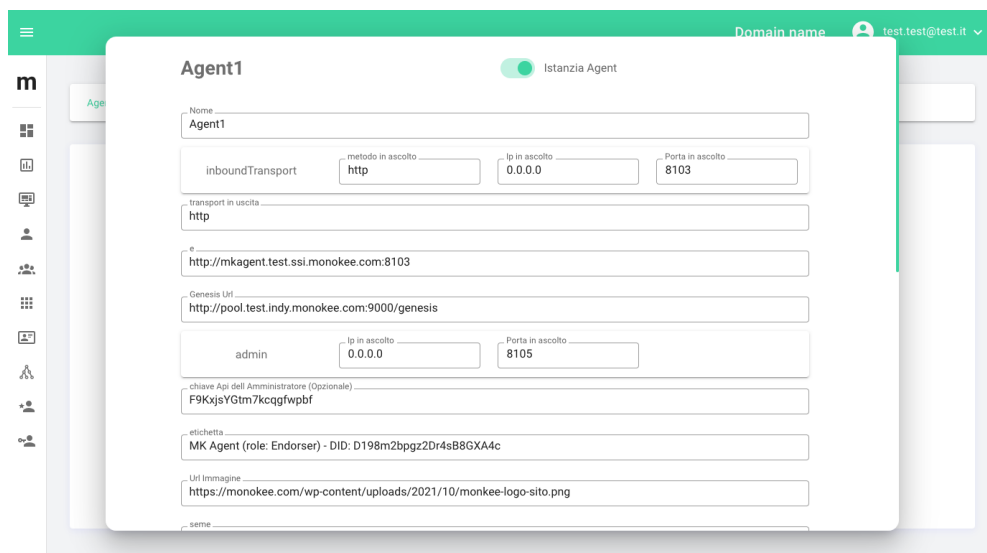


Figura 5.15: Agent - 1

Come mostrato in Figura 5.15 all'interno della *dialog* troviamo un form, identico a quello dell'add *Agent* (per la sua descrizione si rimanda alla sezione 5.3.2.7), già compilato con i dati dell' *Agent* selezionato così da poterne visualizzare o modificare il contenuto.

5.4 Accessibilità

In questa sezione vengono trattati gli aspetti di accessibilità che sono stati presi in considerazione durante lo sviluppo. Si fa notare che le interfacce in esame fanno parte di un progetto molto più grande e già in sviluppo e che quindi si è dovuto seguire le indicazioni date durante l'analisi e la progettazione.

5.4.1 Lingua

Per rendere il sito più accessibile si è scelto di utilizzare "vuei18n" per le traduzioni multilingua. Questo plugin permette, dopo la creazione dei vari "dizionari" nelle lingue desiderate, di andare a tradurre la pagina in base alla lingua predefinita del browser. Durante lo stage sono stati creati due file all'interno della cartella *locales* chiamati "en.js" e "it.js" che sono rispettivamente i "dizionari" per la lingua inglese e quella italiana. Si fa notare che in mancanza del "dizionario" della lingua selezionata nel browser quella di default è l'inglese.

5.4.2 Errori

Per rendere il sito più accessibile si è deciso di utilizzare *vee-validate* per contrassegnare i campi vuoti ma obbligatori all'interno dei vari form. *vee-validate* permette di evidenziare l'input lasciato vuoto colorando il bordo di rosso e inserendo un messaggio esplicativo del motivo dell'errore.

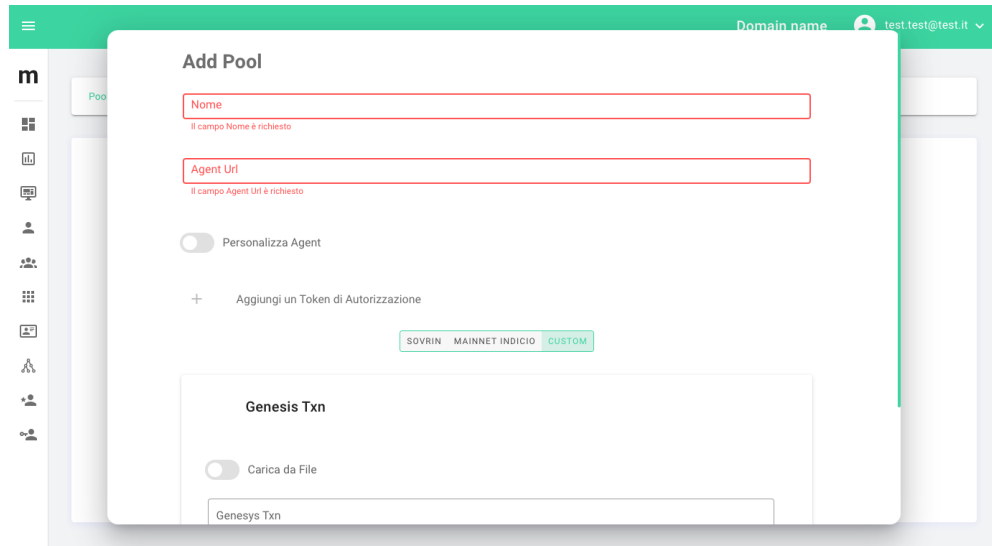


Figura 5.16: Errore input

Si fa notare che, come mostrato in Figura 5.16 per ogni input il testo del messaggio di errore è standard ma in cui viene specificato il suo nome.

5.4.3 Blocco dei Bottoni

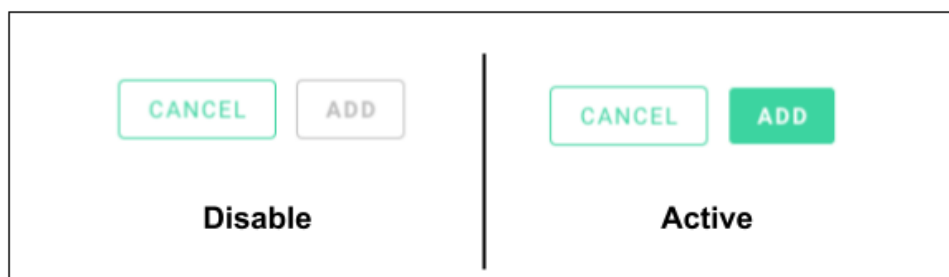


Figura 5.17: Bottone disattivato - attivato

Come mostrato in Figura 5.17 per evitare di aggiungere delle *Pool* o degli *Agent* errati o non completi si è deciso di disattivare il *button* "Add" all'interno delle *dialog* finchè tutti i campi obbligatori non siano completati.

5.4.4 Colori

Per quanto riguarda i colori il discorso è un po più complesso in quanto i colori sono stati una scelta aziendale decisa dai responsabili del progetto ed essi, come mostrato in tabella 5.1, non passano pienamente i *test Web Content Accessibility Guidelines*_g [18] che richiede un rapporto di contrasti di almeno 4.5:1 per il testo normale e 3:1 per il testo in grassetto.

Colore Testo	Colore Sfondo	Contrasto
#FFFFFF	#3CD4A0	1.89:1
#3CD4A0	#FFFFFF	1.89:1
#9E9E9E	#FFFFFF	2.68:1
#6E6E6E	#FFFFFF	5.1:1
#212121	#FFFFFF	16.07:1

Tabella 5.1: Tabella rapporto contrasti

5.4.5 Altri elementi di accessibilità

Per aumentare ulteriormente i livelli di accessibilità, si sono evitati testi lampeggianti, barrati ed in generale *font* troppo elaborati, in modo da agevolare la lettura.

Capitolo 6

Conclusioni

In questo capitolo si dà un consuntivo delle attività, viene analizzato il raggiungimento degli obiettivi prefissati ad inizio stage e vengono esposte delle riflessioni sul percorso di stage.

6.1 Consuntivo finale

In questa sezione si analizza, settimana per settimana, la conformità delle tempistiche preventivate) con quanto realmente accaduto nel corso del tirocinio.

- * **Settimana 1:** settimana svolta rispettando a pieno la pianificazione.
- * **Settimana 2:** settimana svolta rispettando la pianificazione.
- * **Primo ciclo: settimane 3 e 4:** La realizzazione dell'interfaccia per la visualizzazione delle *Pool* ha richiesto molto più tempo del previsto proteandosi anche nella prima settimana del ciclo successivo. Questo ritardo non ha permesso l'implementazione delle chiamate [API](#) con la libreria axios.
- * **Secondo ciclo: settimane 5 e 6:** Il grosso ritardo del ciclo precedente ha portato a un ulteriore ritardo sulla pianificazione non permettendo nuovamente l'implementazione delle chiamate [API](#) per gli *Agent* attraverso la libreria axios
- * **Terzo ciclo: settimane 7 e 8:** i ritardi accumulati nei cicli precedenti hanno reso obbligatoria una ripianificazione in cui si è deciso insieme ai responsabili del progetto di concentrarsi solamente sugli obiettivi obbligatori.

Si deve far notare inoltre che la scelta di concentrarsi solamente sugli obiettivi obbligatori non è dovuta solo ai ritardi accumulati durante l'implementazione ma anche dal fatto che il responsabile del *backend* non è riuscito a realizzarlo in tempo e, perciò, non si è potuto nemmeno effettuare le chiamate [API](#). In accordo con i responsabili si è dunque deciso di focalizzarsi solamente sugli obiettivi obbligatori investendoci più tempo e migliorandoli rispetto alla pianificazione iniziale.

6.2 Raggiungimento degli obiettivi

Nella Tabella [6.1](#) vengono esposte le ore previste per ogni attività in relazione con le ore effettivamente dedicate ad ognuna di esse.

Descrizione dell'attività	Ore previste	Ore effettive
Formazione ed onboarding su stack e tecnologie utilizzate	40	40
Studio di fattibilità ed analisi dei requisiti	30	30
Stesura documentazione relativa ad analisi e progettazione	8	9
Studio e progettazione funzionalità	24	24
Meeting di allineamento interno	10	10
Implementazione delle funzionalità identificate durante la fase di progettazione ed utilizzo delle librerie associate	107	160
Scrittura dei <i>service</i> (su <i>front-end</i>) di chiamata al <i>back-end</i>	60	0
Validazione delle componenti	20	22
Stesura Documentazione	10	12
Collaudo	2	2
Stesura documentazione finale	7	9
Incontro di presentazione della piattaforma con gli <i>stakeholders</i>	1	1
<i>Live</i> demo di tutto il lavoro di stage	1	1
Totale ore	320	320

Tabella 6.1: Tabella di confronto tra le ore previste e le ore effettive per ogni attività

Come indicato dalla tabella, le ore totali coincidono con quelle previste, ci sono però state delle differenze rispetto a quanto pianificato per le motivazioni descritte sopra. Il periodo iniziale di studio si è rivelato importante per apprendere al meglio il *framework* Vue.js e le sue librerie e *plugin* che, prima di questo momento, non avevo mai utilizzato. Sebbene il periodo di stage non sia proseguito in maniera lineare rispetto alla pianificazione è stato molto soddisfacente e il prodotto finito è risultato all'altezza delle aspettative dell'azienda che si è dimostrata soddisfatta del lavoro svolto.

6.3 Raggiungimento degli obiettivi

Gli obiettivi obbligatori prefissati a inizio stage risultano totalmente raggiunti come mostrato nella Tabella 6.2.

Obiettivo	Descrizione	Soddisfatto
OA01	Acquisizione competenze sulle tecnologie e gli strumenti utilizzati	sì
OA02	Capacità di raggiungere gli obiettivi richiesti in autonomia e rispettando le richieste	sì
OA03	Portare a termine l'implementazione grafica della pagina di amministrazione delle <i>Pool</i>	sì
OA04	Portare a termine l'implementazione grafica della pagina di amministrazione degli <i>Agent</i>	sì
OB01	realizzare le chiamate API per la visualizzazione e la modifica dei dati	no
OB02	analisi ed implementazione di dashboard grafiche per visualizzare telemetrie relative al funzionamento della piattaforma	no
OC01	Apportare un valore aggiunto al gruppo di lavoro durante le fasi di progettazione delle interfacce	sì

Tabella 6.2: Tabella degli obiettivi

6.4 Conoscenze acquisite

Il percorso di stage mi ha permesso di fare notevoli progressi sulle conoscenze relative allo sviluppo del *frontend* di applicativi *web*. Il periodo di studio iniziale con l'aiuto del responsabile di progetto mi ha permesso di apprendere Vuejs nel migliore dei modi e quindi di non riscontrare particolari problemi nella fase di sviluppo se non nella parte di interfaccia per la visualizzazione delle *Pool*. Non si può nascondere il dispiacere che, per i diversi motivi descritti precedentemente, non si è riusciti ad apprendere e ad utilizzare tutte le tecnologie pianificate ad inizio stage.

Vorrei aggiungere inoltre che lo stage è stata un'esperienza molto positiva, che mi ha sicuramente permesso di comprendere meglio il mondo del lavoro nel campo informatico. Poter lavorare finalmente su un progetto scelto da me e che tratta argomenti di mio maggiore interesse, mi ha permesso di apprezzare ancora di più questa attività. Ho ritenuto particolarmente stimolante il dover approfondire e studiare argomenti solo tramite documentazione online, diversamente da come avviene tipicamente nel percorso di studi. Ho potuto finalmente capire cosa significa lavorare in una realtà aziendale nel campo dell'Informatica, grazie anche all'interazione con, non solo altri stagisti, ma anche numerosi dipendenti dell'azienda Athesys. In conclusione credo questo percorso di stage sia un fondamentale componente del mio percorso di studio.

Glossario

Agile boards Una scheda agile è la rappresentazione visiva di un processo di sviluppo agile. La bacheca ti consente di visualizzare lo stato corrente di ogni attività, raggruppare le attività correlate, stimare i tuoi sforzi e tenere traccia dei progressi verso il completamento di ciascuna attività. Le colonne sul tabellone rappresentano il ciclo di sviluppo. Le corsie rappresentano un livello secondario della struttura organizzativa. Le impostazioni della scheda ti consentono di configurare la scheda in modo che rifletta il processo di sviluppo utilizzato dal tuo team. [1] . 52

AM La gestione degli accessi è il processo di identificazione, tracciamento, controllo e gestione dell'accesso degli utenti autorizzati o specifici a un sistema o ad un'applicazione. 1, 77

API in informatica con il termine *Application Programming Interface API* (ing. interfaccia di programmazione di un'applicazione) si indica ogni insieme di procedure disponibili al programmatore, di solito raggruppate a formare un set di strumenti specifici per l'espletamento di un determinato compito all'interno di un certo programma. La finalità è ottenere un'astrazione, di solito tra l'hardware e il programmatore o tra software a basso e quello ad alto livello semplificando così il lavoro di programmazione. 9, 49, 52, 71, 73, 77

BlockChain Un insieme di tecnologie, in cui il registro è strutturato come una catena di blocchi contenenti le transazioni e il consenso è distribuito su tutti i nodi della rete. Tutti i nodi possono partecipare al processo di validazione delle transazioni da includere nel registro. 4

Form indica la parte di interfaccia utente di un'applicazione web che consente all'utente client di inserire e inviare al web server/application server uno o più dati liberamente digitati dallo stesso sulla tastiera attraverso l'uso di componenti grafici detti widget presenti sull'interfaccia stessa. v

IAM si intendono i sistemi integrati di tecnologie, criteri e procedure in grado di consentire alle organizzazioni di facilitare - e al tempo stesso controllare - gli accessi degli utenti ad applicazioni e dati critici, proteggendo contestualmente i dati personali da accessi non autorizzati.. 1, 77

IGA si intendono i sistemi integrati di tecnologie, criteri e procedure in grado di consentire alle organizzazioni di facilitare, e al tempo stesso controllare, gli accessi degli utenti ad applicazioni e dati critici, proteggendo contestualmente i dati personali da accessi non autorizzati.. 1, 77

Javascript linguaggio di programmazione multi paradigma orientato agli eventi, comunemente utilizzato nella programmazione Web lato client (esteso poi anche al lato server). [v](#), [51](#), [56](#)

Json è un formato di file standard (con estensione *.json*) e un formato di interscambio di dati che utilizza testo leggibile dall'uomo per archiviare e trasmettere oggetti di dati costituiti da coppie di attributi-valori e array (o altri valori serializzabili). È un formato di dati comune con usi diversi nello scambio elettronico di dati, compreso quello di applicazioni web con server [\[10\]](#). . [4](#), [5](#), [23](#), [48](#), [64](#), [77](#)

MVC L'MVC è un *design pattern* che porta alla creazione di un'architettura divisa in Model, View e Controller [\[13\]](#). [77](#)

MVVM L'MVVM è un design pattern che porta alla creazione di un'architettura divisa in Model, View e ViewModel [\[14\]](#). [77](#)

Self Sovereign Identity La Self Sovereign Identity (SSI) è un modello di identità digitale decentralizzato basato su tecnologia blockchain e distributed ledger. Questo modello consente all'utente di essere possessore dei propri dati. [v](#), [5](#)

SEO tradotto significa ottimizzazione per i motori di ricerca. È una disciplina informatica che ha l'obiettivo incrementare la visibilità e il posizionamento di un sito web all'interno dei risultati organici dei motori di ricerca. [\[site:SEO\]](#). [51](#), [77](#)

UI Letteralmente interfaccia utente, rappresenta tutto quello che è visibile all'utente finale. [1](#), [51](#), [77](#)

UML in ingegneria del software *UML*, *Unified Modeling Language* (ing. linguaggio di modellazione unificato) è un linguaggio di modellazione e specifica basato sul paradigma object-oriented. L'*UML* svolge un'importantissima funzione di "lingua franca" nella comunità della progettazione e programmazione a oggetti. Gran parte della letteratura di settore usa tale linguaggio per descrivere soluzioni analitiche e progettuali in modo sintetico e comprensibile a un vasto pubblico. [77](#)

UX Letteralmente Esperienza Utente, utilizzato per definire la relazione tra una persona e un prodotto, un servizio, un sistema.. [1](#), [58](#), [77](#)

Web Content Accessibility Guidelines Le *Web Content Accessibility Guidelines* (WCAG) indicano come rendere i contenuti su internet più accessibili per le persone con disabilità. Esistono tre livelli di conformità A, AA e AAA [\[18\]](#). [69](#)

WebApp Una applicazione web (web application in inglese, abbreviato web app), in informatica ed in particolare nella programmazione web, indica genericamente tutte le applicazioni distribuite ovvero applicazioni accessibili/fruibili via web per mezzo di un network o attraverso internet, cioè in una architettura tipica di tipo client-server, offrendo determinati servizi all'utente client.. [1](#), [11](#), [49](#), [57](#)

Acronimi

AM [Access Management](#). 75

API [Application Program Interface](#). 75

IAM [Identity and Access Management](#). 75

IGA [Identity Governance Administration](#). 75

Json [JavaScript Object Notation](#). 76

MVC [Model View Controller](#). 57, 76

MVVM [Model View ViewModel](#). 58, 76

SEO [Search Engine Optimization](#). 76

UI [User Interface](#). 76

UML [Unified Modeling Language](#). 12, 76

UX [User Experience](#). 76

Bibliografia

Riferimenti bibliografici

- [8] Daniel T. Jones James P. Womack. *Lean Thinking, Second Editon*. Simon & Schuster, Inc., 2010.

Siti web consultati

- [1] *Agile Board*. URL: <https://www.jetbrains.com/help/youtrack/cloud/Agile-Board-Glossary.html>.
- [2] *Axios*. URL: <https://www.cloudsigma.com/configuring-a-rest-api-with-axios-in-vue-js-a-tutorial/>.
- [3] *best practice*. URL: <https://learnvue.co/tutorials/vue-best-practices>.
- [4] *Decorator Design Pattern*. URL: <https://it.wikipedia.org/wiki/Decorator>.
- [5] *Diagrammi dei casi d'uso*. URL: https://it.wikipedia.org/wiki/Use_Case_Diagram.
- [6] *Figma*. URL: [https://en.wikipedia.org/wiki/Figma_\(software\)](https://en.wikipedia.org/wiki/Figma_(software)).
- [7] *GitHub*. URL: <https://it.wikipedia.org/wiki/GitHub>.
- [9] *Javascript*. URL: <https://it.wikipedia.org/wiki/JavaScript>.
- [10] *Json*. URL: <https://en.wikipedia.org/wiki/JSON>.
- [11] *Manifesto Agile*. URL: <http://agilemanifesto.org/iso/it/>.
- [12] *Micorsoft Teams*. URL: https://it.wikipedia.org/wiki/Microsoft_Teams.
- [13] *MVC Pattern*. URL: <https://www.laramind.com/blog/definizione-di-mvc-cose-un-framework-mvc/>.
- [14] *MVVM Pattern*. URL: <https://it.wikipedia.org/wiki/Model-view-viewmodel>.
- [15] *Nuxt.js*. URL: <https://en.wikipedia.org/wiki/Nuxt.js>.
- [16] *Pool configuration*. URL: <https://hyperledger-indy.readthedocs.io/projects/node/en/latest/transactions.html>.
- [17] *Screen Reader*. URL: https://it.wikipedia.org/wiki/Screen_reader.
- [18] *Test WCAG AA*. URL: <https://www.w3.org/WAI/WCAG2AA-Conformance>.

- [19] *Vee-validate*. URL: <https://vee-validate.logaretm.com/v4/>.
- [20] *Visual studio code*. URL: https://en.wikipedia.org/wiki/Visual_Studio_Code.
- [21] *Vue.js*. URL: <https://vuejs.org/guide/introduction.html>.
- [22] *Vuei18n*. URL: <https://kazupon.github.io/vue-i18n/introduction.html>.
- [23] *Vuetify*. URL: <https://vuetifyjs.com/en/introduction/why-vuetify/#getting-started>.
- [24] *YouTrack Administration*. URL: <https://www.jetbrains.com/help/youtrack-server/introduction-to-youtrack-server.html>.