
PRANVERAAPP SRL

Doc. Nome: prv-app-login

Titolo: prv-app-login Architecture

P.A.S

Author:

Elion Haxhi
PranveraApp

Date:

Signature:

Storico del Documento

Versione	Data	Autore	Commenti
1.0.0	05/03/2021	1. Elion Haxhi	Primo Rilascio
1.0.1	17/02/0223	2. Elion Haxhi	Correzione testo

Tabella dei contenuti

1. DOCUMENTATION ROADMAP	2
1.1. COME È ORGANIZZATO IL DOCUMENTO	3
1.2. STAKEHOLDER COINVOLTI	3
2. DOCUMENT VISION	3
2.1. PROBLEM BACKGROUND	3
2.1.1. System Overview	3
2.1.2. Goals and Context	3
3. FUNCTIONAL REQUIRMENTS	4
3.1. LOGIN	4
3.2. CHECKTOKEN	4
4. NON FUNCTIONAL REQUIREMENT	5
4.1. SCALABILITA	ERRORE. IL SEGNALIBRO NON È DEFINITO.
4.2. SICUREZZA	5
5. USE CASES	6
5.1. USE CASE – LOGIN	6
5.2. USE CASE – CHECKTOKEN	7
6. ACTIVITY DIAGRAM	7
7. HIGH LEVEL SOLUTION	9
7.1. ASSUMPTION	10
7.2. RISK AND MITIGATION	10
7.3. SEQUENCE DIAGRAM	10
7.3.1. Sequence Login	10
7.3.2. CheckToken	10
7.4. CLASS DIAGRAM	11
7.5. COMPONENT DIAGRAM	12
7.6. DEPLOYMENT DIAGRAM	13

1. Documentation Roadmap

Questa wiki serve per illustrare l'architettura del'applicativo prv-app-login. E' una app accessibile con i browser comuni come chrome , firefox, internet explorer.

1.1. Come è organizzato il documento

- Documentation Roadmap: fornisce informazioni riguardanti questo documento
- Document Vision: fornisce informazioni del architettura del applicativo , descrive i requisiti del business e la composizione generica del impianto.
- Functional Requirements: Descrive i servizi chiesti dal business
- Non Functional Requirements: Descrive le varie problematiche richieste dal business
- Use Cases: descrive I requisiti funzionali mediante gli use cases
- Activity diagram: descrive i comportamenti del sistema per soddisfare i requisiti
- Refinement: descrive gli aggiornamenti dei use case
- High level solution: descrive la soluzione mediante i diagrammi UML

1.2. StakeHolder coinvolti

1.	Project Manager
2.	Software Architects
3.	Software Developers
4.	Business Experts

2. Document Vision

2.1. Problem Background

2.1.1. System Overview

C'e' la necessità di creare un app con la responsabilità di autenticare vari utenti che vogliono utilizzare gli applicativi del PranveraApp factory. Tutti gli applicativi comunicano via protocollo https.

2.1.2. Goals and Context

L'applicativo prv-app-login deve essere responsabile per autenticare utenti che accedono mediante username a password. L'app deve verificare che l'utente sia un utente reale se si allora prv-app-login invia un token come parametro al l'applicativo a cui vuole accedere. Quindi prv-app-login deve fare due cose:

- Staccare i token jwt
- Validare i rispettivi token jwt

La figura 1 ci illustra ad alto livello la struttura architetturale di prv-app-login e i suoi partecipanti.

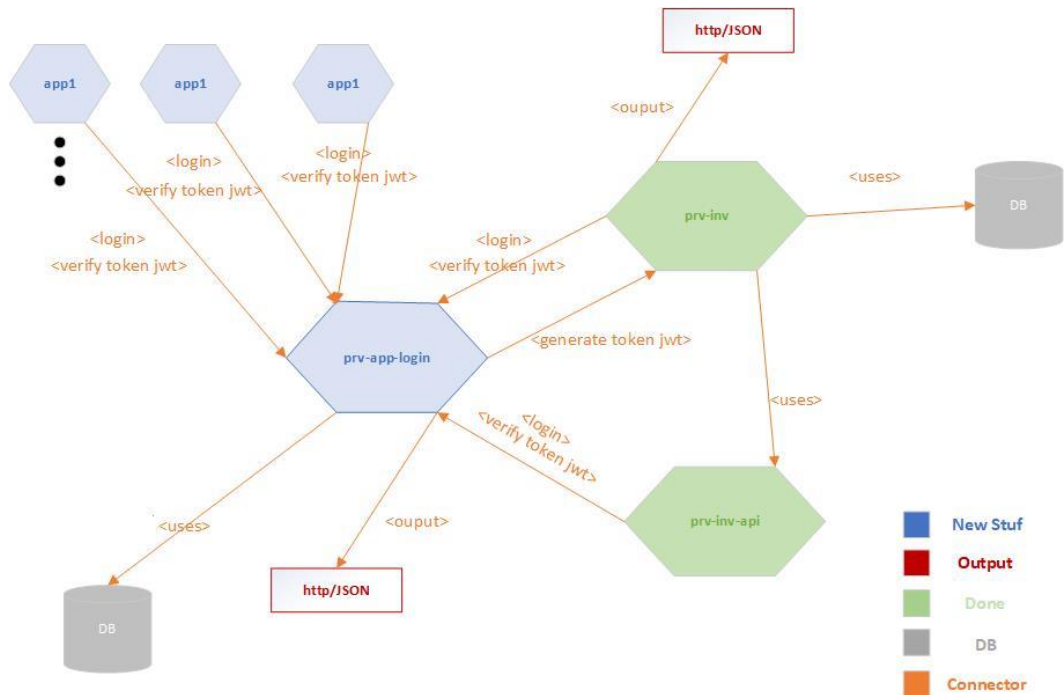


figura 1

3. Functional Requirments

3.1. Login

Quando un utente vuole autenticarsi su una app che fa parte del insieme PranveraApp allora l'applicativo prv-app-login deve entrare in scena è l'utente si troverà davanti a una form dove potrà inserire le sue credenziali. Se l'autenticazione va a buon fine allora mediante un meccanismo di redirect l'utente potrà usare la sua app di interesse.

3.2. CheckToken

Se un utente autenticato precedentemente chiude il browser e in un secondo momento lo riapre e cerca di usare un app del insieme dei applicativi PranveraApp allora l'accesso sarà automatico non ci sarà bisogno di autenticarsi di nuovo. Se esiste il cookie allora entrerà in gioco la procedura di autenticazione in modo "silent background". Ovvero l'app ricaverà le credenziali di accesso dal token e successivamente gli aggiungerà nel

SpringContext per le app con framework spring invece per quelle di tipo JavaEE vera usato una chiamata al database per ottenere il profilo utente.

4. Non Functional Requirement

4.1. Scalabilità

L'applicazione vera utilizzata con un alta frequenza da parte delle altre app. Quindi è richiesto un alto livello di scalabilità.

4.2. Sicurezza

Le chiamate di autenticazione devono essere fatte sotto un canale https. Non sarà possibile vedere il plain text della password la verifica sarà fatta sulla password cifrata mediante un algoritmo di cifratura.

5. Use Cases

La figura 2 riporta tutti gli use case necessarie richiesti dal business.

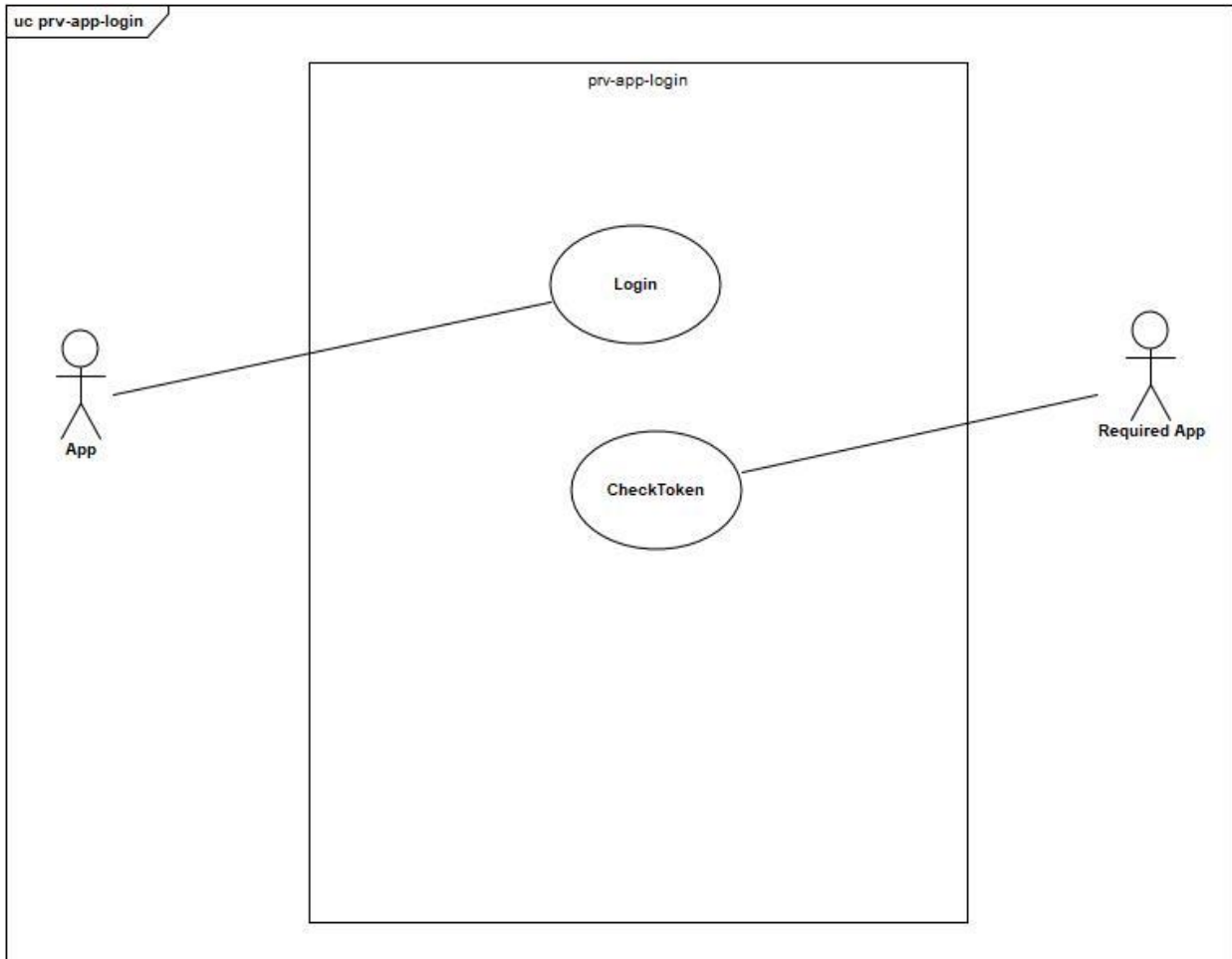


figura 2

5.1. Use Case – Login

Il login da la possibilità di accedere a servizi protetti di altre app.

1. L'utente vuole accedere la app1 quindi l'applicazione(app1) come primo passo tenta di ottenere il cookie corrispondente dalla request, se il cookie non esiste allora la app1 invoca prv-app-login.
2. L'utente inserisce lo username e la password e fa la submit della request.
3. La prv-app-login fa le verifiche per autenticare l'utente, se le verifiche sono ok allora la prv-app-login genera un token lo setta nel cookie e fa il redirect verso la app1.

5.2. Use Case – CheckToken

Dopo il login di successo l'applicativo avrà una session di durata 2 ore. Le app verranno invocate sempre con un parametro jwt token.

1. Per le app che espongono solo servizi API, si farà un check del token autenticazione chiamando il servizio esposto da prv-app-login se la risposta è vera allora le API daranno servizio se invece la risposta è falsa un popup di authorization/authentication si evidenzierà.

6. Activity Diagram

Nella figura 3 vediamo il flusso principale che di login.

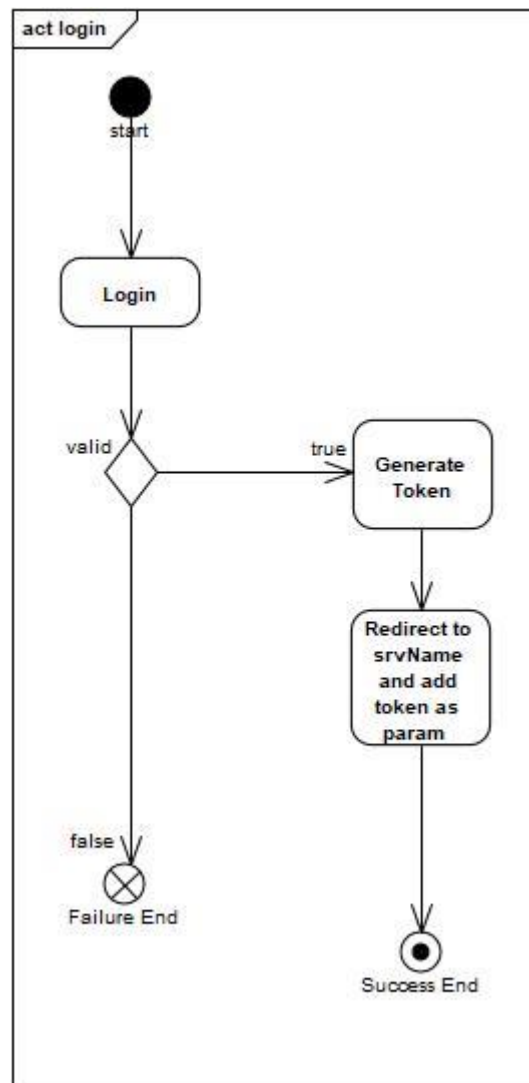


figura 3

Nella figura 3.1 vediamo la procedura di checkToken. Questo endpoint dovrà essere chiamato dalle app alle quali è stato richiesto servizio e perciò eseguiranno la procedura di verifica token.

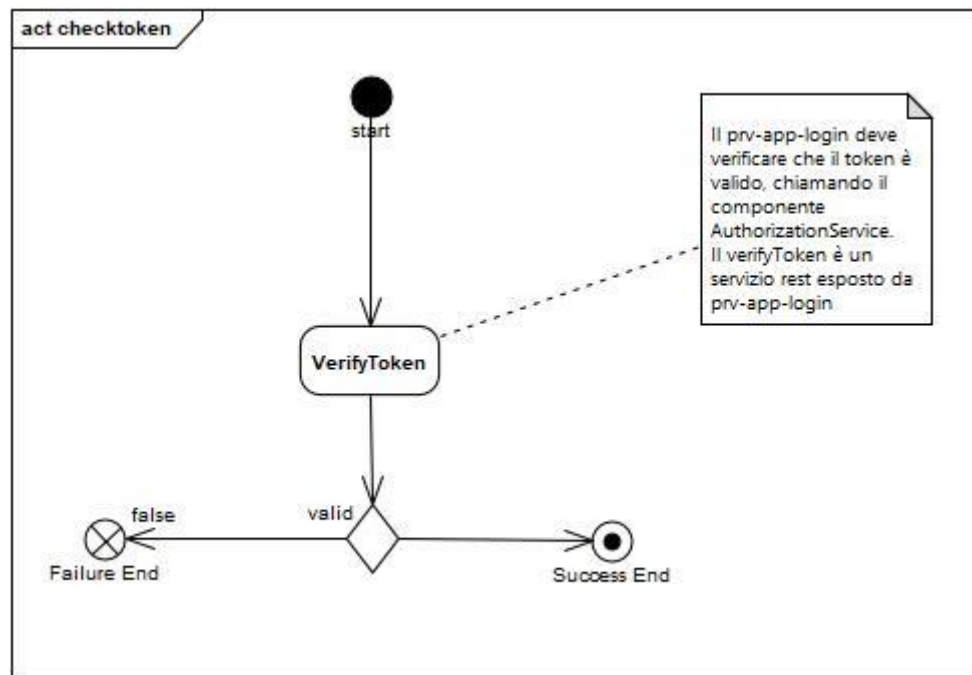


figura 3.1

7. High level solution

7.1. Assumption

- Tutte le app devono utilizzare il pattern front controller per la verifica del cookie.

7.2. Risk and Mitigation

Risk	Mitigation
I dati sensibili del utente viaggiano in chiaro.	Tutte le chiamate verso la prv-app-login devono essere fatte sotto il protocollo https.

7.3. Sequence Diagram

7.3.1. Sequence Login

Nella figura 5 viene illustrata il diagramma della procedura di login.

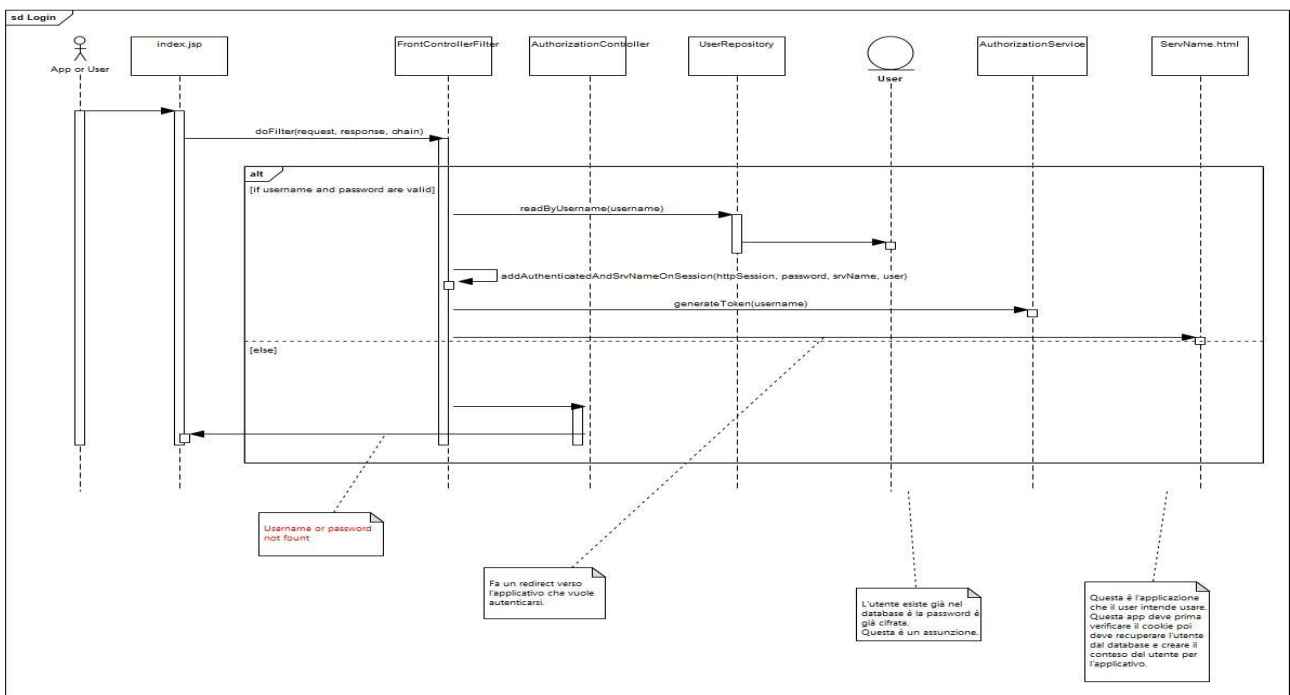


figura 5

7.3.2. CheckToken

Nella figura 6 viene illustrata il diagramma della procedura di checkToken.

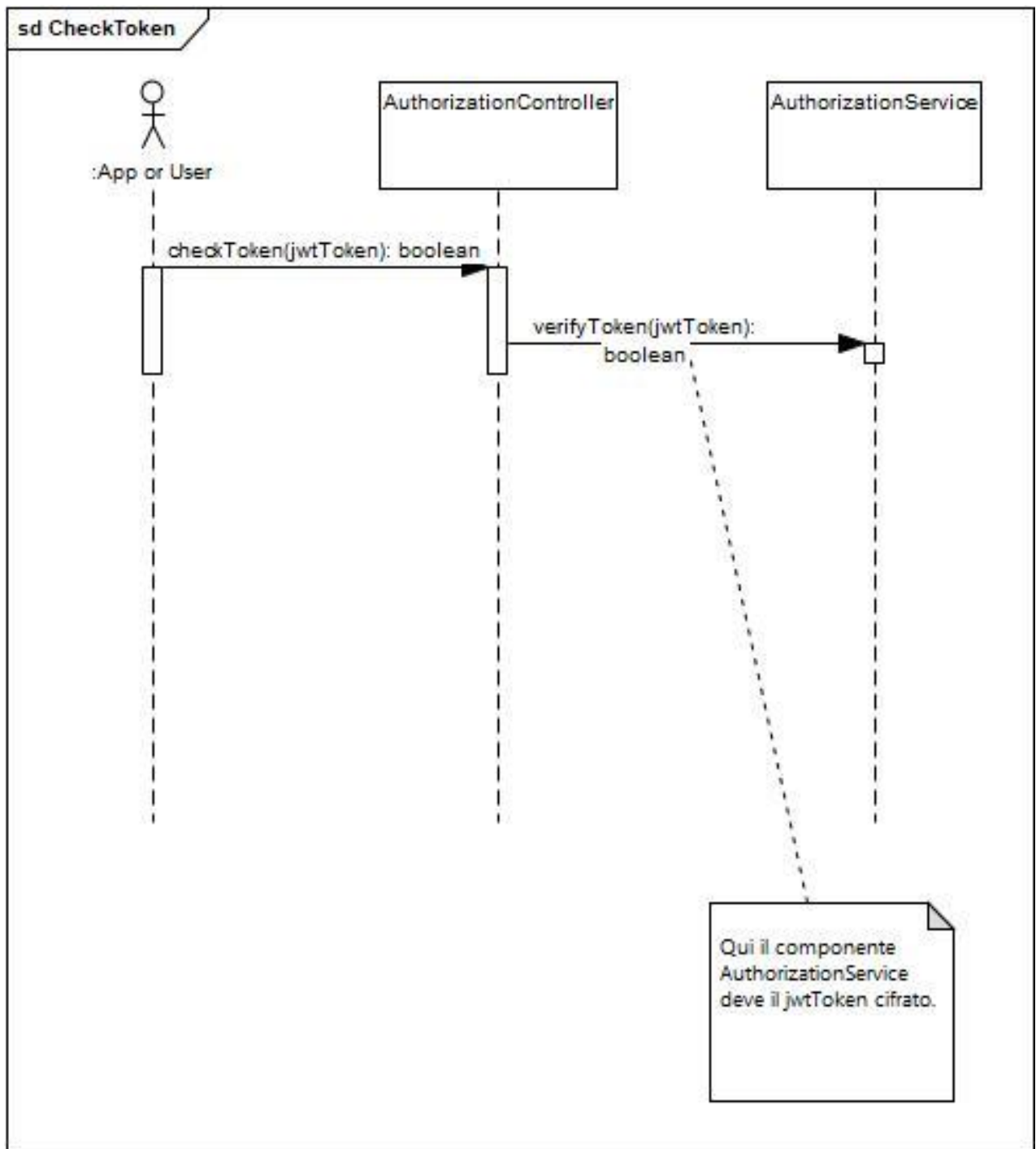


figura 6

7.4. Class Diagram

Nella figura 8 viene riportato la struttura del progetto prv-app-login. NB le firme e il numero dei attributti devono essere riportate nel codice sorgente.

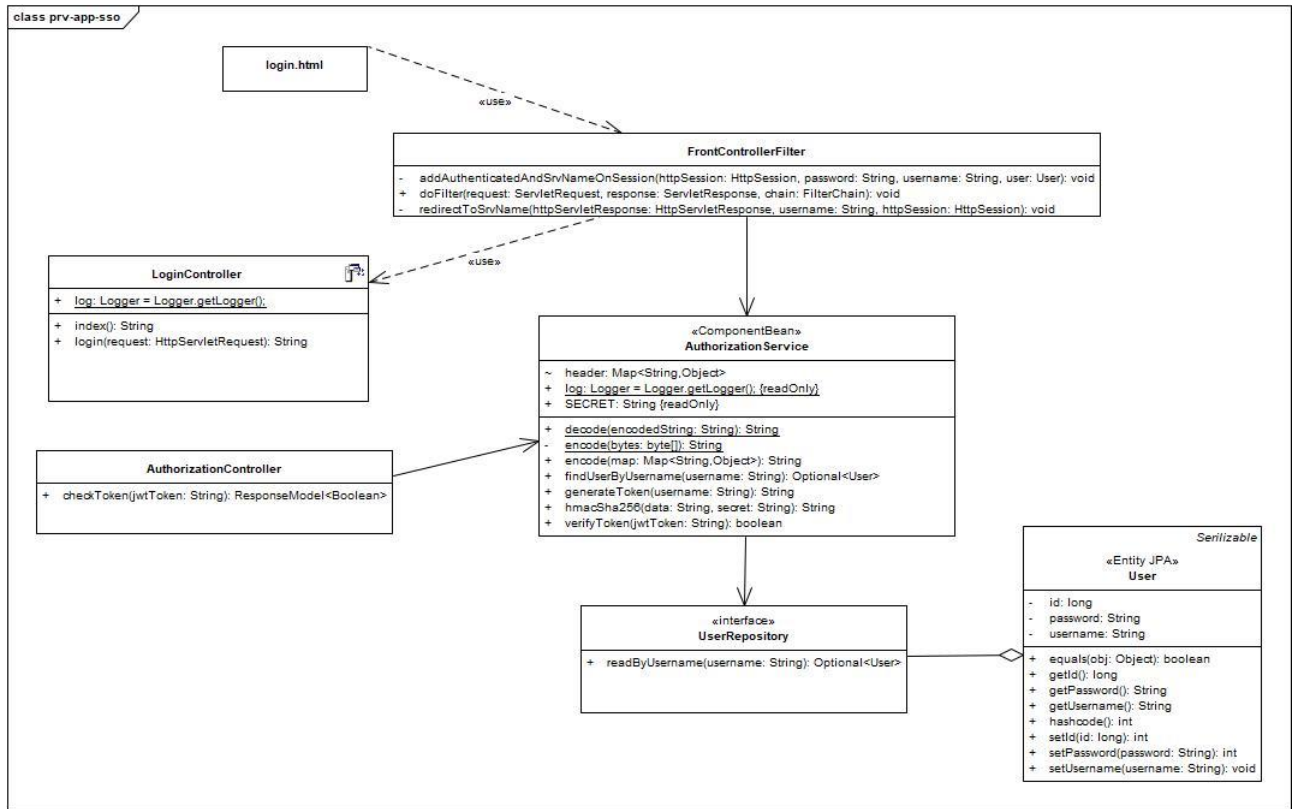


figura 8

7.5. Component Diagram

Nella Figura 9 vediamo il component diagram per il prv-app-login.

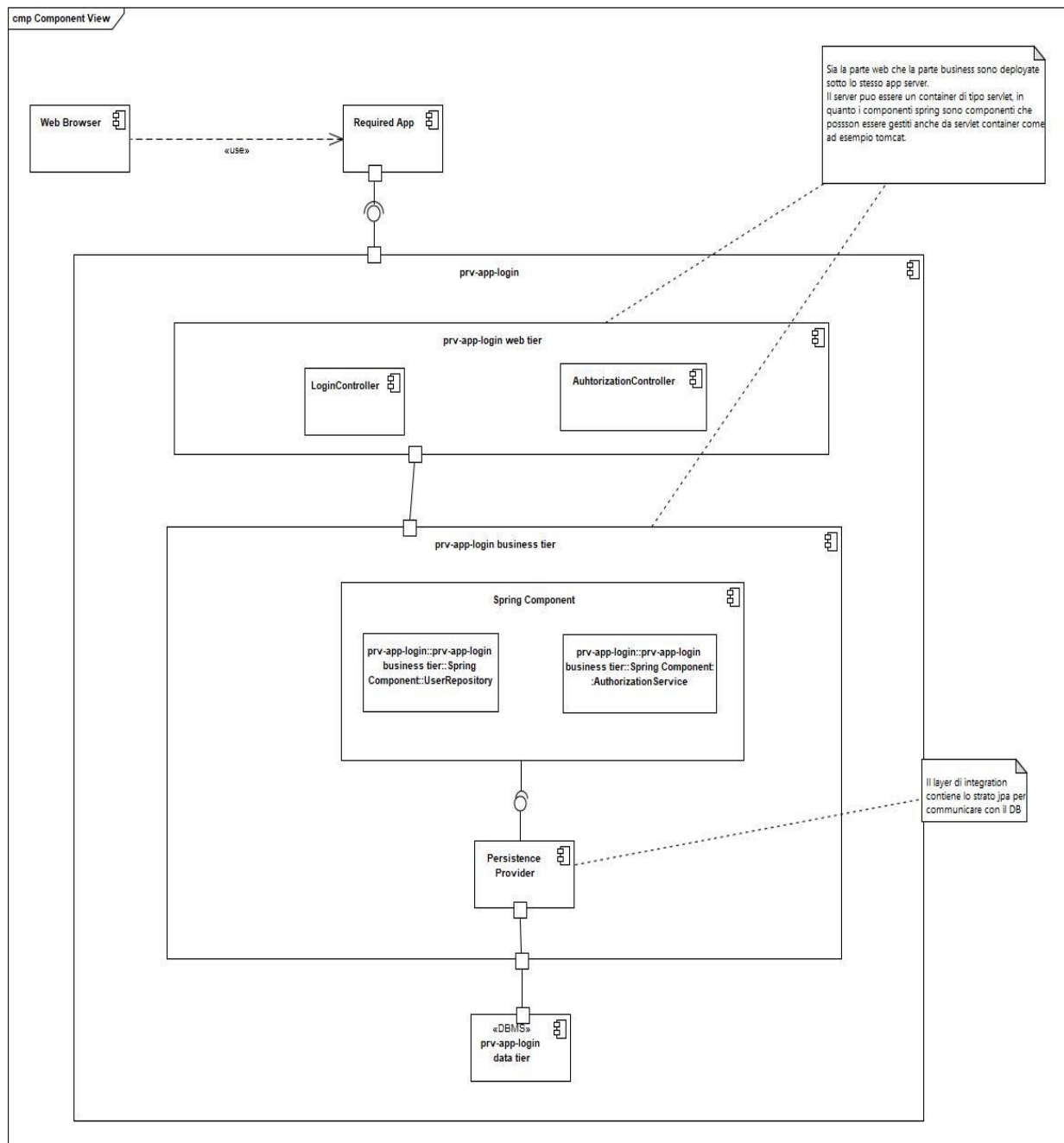


figura 9

7.6. Deployment Diagram

Nella Figura 10 vediamo gli hardware coinvolti per il deploy del prv-app-login.

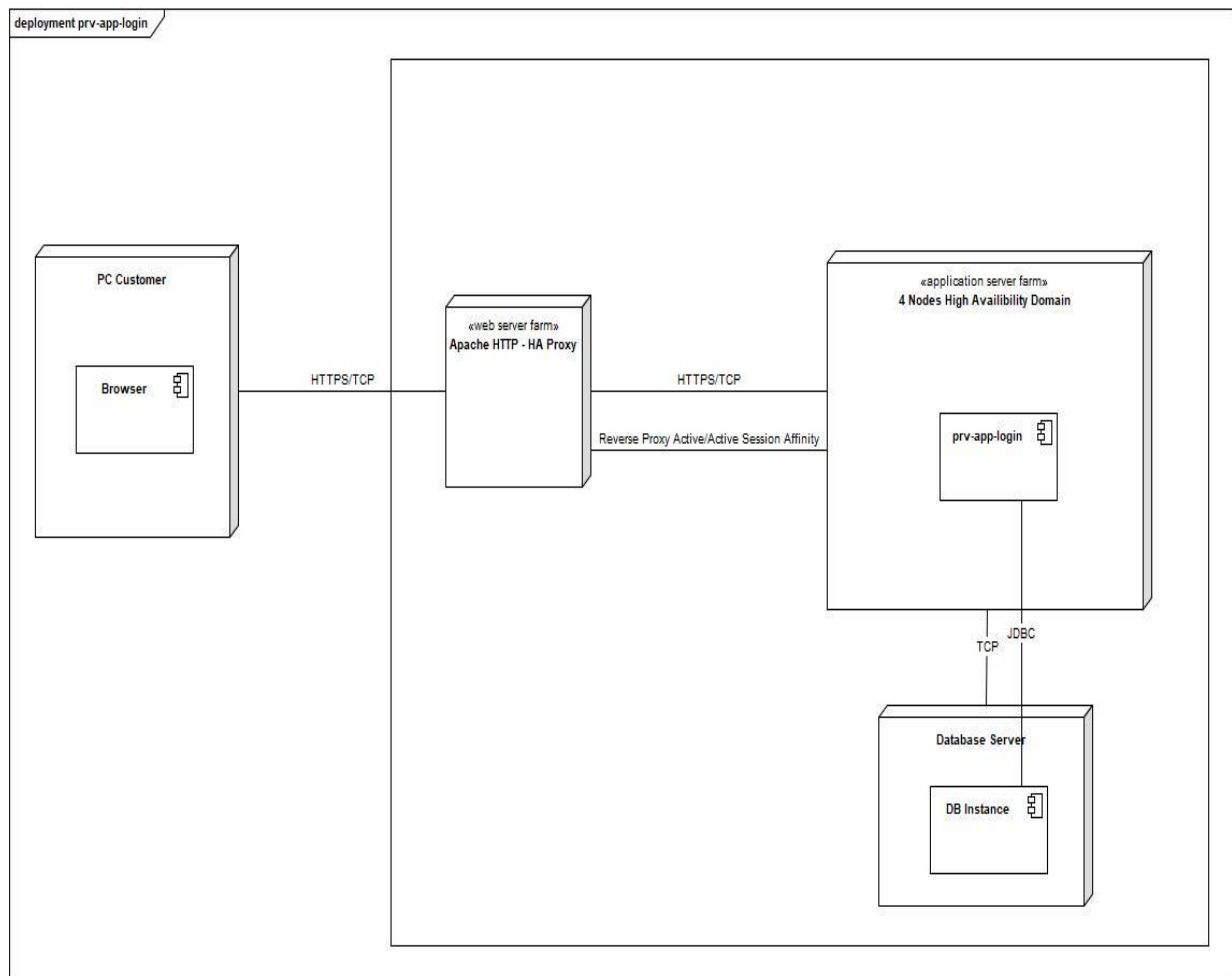


figura 10