ROMÂNIA
MINISTERUL APĂRĂRII NAŢIONALE
ACADEMIA TEHNICĂ MILITARĂ „FERDINAND I”
FACULTATEA DE SISTEME INFORMATICE ŞI SECURITATE CIBERNETICĂ
Specializarea: Calculatoare și sisteme informatice pentru apărare și securitate națională

Sistem de autentificare multifactor utilizând dispozitive mobile

CONDUCĂTOR ŞTIINŢIFIC:

Cpt. Lect. Dr. Ing. **Iulian ACIOBĂNIȚEI**

STUDENT:

Std. Sg. Maj. **Bianca IONAȘCU**

BUCUREŞTI 2023

Cuprins

1. Introducere

1.1. Scurtă introducere

Acest proiect își propune să dezvolte și să implementeze un sistem robust de autentificare multi-factor (MFA) utilizând Keycloak ca server de autentificare centralizat. Prin integrarea cu Active Directory și prin adăugarea de module personalizate pentru TOTP, Push Notification și Token Hardware, sistemul va oferi o soluție de securitate avansată pentru gestionarea accesului utilizatorilor la diverse resurse și aplicații.

1.2. Scopul

Scopul acestui proiect este de a dezvolta și implementa un sistem de autentificare multi-factor personalizat și open-source, utilizând Keycloak, răspunzând astfel la tendințele actuale și la creșterea adopției Keycloak la nivel mondial. Proiectul își propune să extindă funcționalitatea Keycloak prin integrarea TOTP (printr-o implementare proprie diferită de cea existența), a notificărilor push și a tokenurilor hardware, oferind o soluție MFA completă și adaptabilă, care poate fi utilizată într-o varietate de contexte organizaționale și industriale.

Proiectul își propune să ofere o soluție echivalentă cu cele existente pe piață, dar cu avantajul flexibilității și adaptabilității specifice unui produs open-source.

1.3. Motivație

Motivația acestui proiect este stimulată de popularitatea în creștere a Keycloak ca soluție de Identity and Access Management (IAM), conform unor statistici din 2023 care afirmă că Keycloak este una dintre cele mai bune, cele mai testate și cunoscute opțiuni de pe piață [1]. Cu o bază largă de utilizatori și o comunitate activă, Keycloak reprezintă o platformă ideală pentru inovații în securitatea autentificării. Implementarea acestor module MFA nu numai că vor îmbunătăți securitatea utilizatorilor finali, dar vor aduce, de asemenea, o valoare adăugată considerabilă întregii comunități Keycloak, oferind opțiuni suplimentare de securitate care pot fi adaptate la nevoile specifice ale diferitelor organizații.

1.4. Obiective

În cadrul acestui proiect, îmi propun un sistem avansat de autentificare multi-factor (MFA) integrat cu Keycloak, care să răspundă eficient la provocările actuale de securitate cibernetică. Scopul principal este de a automatiza procesul de autentificare și de a oferi mai multe straturi de securitate, fără a compromite experiența utilizatorului. Câteva dintre obiectivele principale ale proiectului sunt:

- Configurarea Keycloak pentru a utiliza Active Directory ca sursă de autentificare, sincronizând utilizatorii și grupurile. Multe dintre companiile de astăzi dețin un Active Directory, care are o evidență clară a utilizatorilor și Keycloak oferă suport pentru integrarea cu LDAP/Kerberos (User Federation).
- Crearea unui modul Keycloak care generează și validează coduri TOTP, bazat pe standardele RFC (6238 si 4226 (HOTP)).
- Dezvoltarea unui modul Keycloak care integrează FCM pentru a trimite și gestiona notificări push către dispozitivele mobile ale utilizatorilor.
- Adăugarea suportului pentru tokenuri hardware (de ex. cum oferă Yubico => YubiKey) ca metodă de autentificare. Acest lucru îl voi face prin integrarea unui servicu, precum Nexus, în Keycloak pentru a facilita comunicarea cu tokenurile hardware și validarea autentificarilor.
- Utilizarea API-urilor Keycloak pentru a automatiza procesul de autentificare și pentru a îmbunătăți experiența utilizatorului.
- Realizarea de teste ample pentru a verifica securitatea, performanța și conformitatea sistemului cu standardele de securitate.
- Crearea unei aplicații mobile care nu doar afișează codurile OTP, dar și oferă un management eficient al autentificarii multi-factor, inspirat din schema Aegis [2].
- Dezvoltarea unei aplicații web simpliste care să servească drept platformă de testare și demonstrație a eficacității sistemului de autentificare multi-factor dezvoltat. Construirea unui site web simplu, dar funcțional, care utilizează Keycloak pentru autentificare. Acest site va include funcționalități de bază pentru a permite utilizatorilor să se autentifice folosind metodele MFA implementate: TOTP, notificări push și tokenuri hardware. Scopul este de a demonstra cum sistemul integrat lucrează într-un mediu real, simulând fluxul de autentificare de la început până la sfârșit.

Scopul acestui proiect este să dezvolt module care nu doar întăresc securitatea datelor, dar și îmbunătățesc substanțial experiența utilizatorilor. As dori sa concep un sistem de autentificare care să fie atât de intuitiv și de accesibil, încât să motiveze un număr cât mai mare de persoane să adopte practici eficiente de securitate a datelor. Prin focalizarea pe ușurința în utilizare, doresc sa fac ca utilizarea metodelor avansate de protecție a datelor sa nu fie doar o necesitate, ci și o preferință a utilizatorilor.

2. Cerințe

2.1. Cerințe funcționale

- Sistemul trebuie să suporte autentificarea multi-factor, inclusiv TOTP, push notification și tokenuri hardware.
- AD (extern) trebuie să fie configurat ca sursă primară de gestionare a utilizatorilor și a grupurilor, oferind servicii de director pentru autentificarea inițială a utilizatorilor.
- Trebuie dezvoltată o aplicație Android (care să se integreze cu Keycloak) care să suporte generarea de coduri TOTP și recepționarea notificărilor push pentru autentificarea în doi factori.

    - ***Cerințe Keycloak:***

        - Keycloak trebuie să se poată integra cu Active Directory pentru autentificarea primară a utilizatorilor. [3]
        - Primul Factor: Utilizatorul când va accesa aplicația web de test în cazul în care nu are o sesiune activă, va fi redirecționat către pagina de Log-in a Keycloak-ului unde va trebui să introducă username-ul/email-ul și parola asociate contului.
        - Dacă credențialele de conectare nu sunt corecte va trebui să reîncerce logarea în cont, în schimb dacă au fost corecte acesta va trece la cel de-al doilea factor de autentificare. Cel de-al doilea factor de autentificare poate să fie unul din cele 3 module puse la dispoziție: TOTP, Push Notification sau Token Hardware.
        - Al doilea factor: Utilizatorul în funcție de ce metodă de 2FA are configurată va avea de realizat anumite challenge-uri:
            - ***TOTP***: user-ul va fi redirecționat către o pagină web unde i se va solicita codul OTP corespunzător aplicației de pe aplicația mobilă.
            - ***Push Notification***: user-ul o să fie redirectionat către o pagină web unde i se va afișa un cod format din 2 cifre pe care îl va solicita prin push notification și telefonul. În interfața mobile o să aibe un spațiu unde ar trebui să completeze cu codul din interfața web și două butoane: Decline și Accept. Butonul Decline va refuza conectarea pe aplicația asociată, butonul de Accept va cere un cod valid în câmpul dat => Sistemul trebuie să poată trimite notificări push către dispozitivele mobile ale utilizatorilor pentru autentificarea 2FA.
            - ***Token Hardware***: utilizatorul va introduce tokenul în device.
        - Utilizatorul își poate schimba modulul de 2FA utilizat în funcție de preferințe.

- Sistemul Keycloak trebuie să valideze autenticitatea codurilor TOTP, notificărilor push și a autentificării prin tokenul hardware înainte de a permite accesul.
- Functionalitate de resetare a parolei.

- ***Cerințe aplicație web de test:***

  - Existența unei aplicații web de test: o aplicație de vot care va avea 3 roluri: user (votant), administrator, candidat.
  - Site-ul web va fi înregistrat în keycloak, care se va ocupa de partea de autentificare a utilizatorilor.

    - *Cerințe pentru user (votant):*

      - Acestui utilizator i se va deschide o pagină în care se vor afla toți candidații cu o descriere sumară a lor, cu câte o poză corespunzătoare fiecăruia și cu câte un buton de Vote.
      - Va dispune de un navbar în care va exista un buton de View Profile, care ar trebui să deschidă o nouă pagină care să aibe detaliile contului utilizatorului conectat alături de butoane de modify asupra anumitor caracteristici ale contului (de ex: change password, change profile photo).
      - După ce votează acesta nu mai are dreptul de vot asupra altui candidat, îi va apărea de fiecare dată când reintră pe cont o pagină cu mesajul „Votul tău a fost înregistrat! Vă mulțumim!”.

    - *Cerințe pentru administrator:*

      - Acestui utilizator i se va deschide o pagină de Overview atunci când intră în cont. Această va conține un sidebar cu mai multe opțiuni de pagini ce întrunesc anumite informații pentru administrator: Overview, Candidates, Results, Voters, Position, Election Title, View Profile.
      - Pagină Overview: un grafic cu statistica numărului de voturi/ora și alte informații despre participarea la vot, numărul de candidați etc.
      - Pagină Candidates: afișarea tuturor candidaților cu prezentarea lor sumară împreună cu imaginea corespunzătoare a fiecăruia și alte detalii despre ei.
      - Pagină Results: reprezentarea într-un grafic a rezultatelor candidaților.

- Pagină Voters: informații despre procentajul de participare la vot împreună cu un tabel care prezintă votanții care au votat deja.
- Pagină Position: o ierarhie a pozițiilor ocupate de candidați în funcție de numărul de voturi primit.
- În pagina Election Title se vor afișa informații despre alegerea electorală în desfășurare, iar în pagina View Profile utilizatorul își poate administra informațiile despre propriul cont (va fi exact ca la user).

- <u>*Cerințe pentru candidat:*</u>

  - Acestui utilizator i se va deschide o pagină care va conține numărul de voturi acumulat și un grafic care prezintă ierarhia locurilor ocupate de fiecare candidat în funcție de numărul de voturi primit.
  - O să aibe și un navbar asemănător cu cel de la user, doar că acesta își poate modifica și sloganul pentru prezenatrea electorală.

- ***Cerințe pentru aplicația android:***

  - Aplicația mobilă trebuie să poată genera coduri TOTP conform standardului RFC 6238.
  - Aplicația mobilă trebuie să poată primi și afișa notificări push.
  - Interfața aplicației va fi formată din: o pagină cu toate aplicațiile la care este înrolat user-ul împreună cu codul OTP asociat, un timer pentru fiecare cod în parte și niște setări configurabile privind interfața.

2.2. Cerințe nefuncționale

- Sistemul Keycloak trebuie să fie configurat pentru a opera în spatele unui reverse proxy.
- Adăugarea întregii aplicații în Kubernetes/Docker.

- ***Cerințe Keycloak:***

  - Adăugarea unor SPI-uri (Service Provider Interface) personalizate pentru cele 3 module de 2FA: TOTP, Push Notification și Token Hardware.
  - Administratorul sistemului Keycloak configurează diferitele metode disponibile de autentificare în doi factori (2FA) pentru întregul sistem.

Aceste metode includ TOTP, notificări push, tokenuri hardware. Această configurare este globală, adică stabilește ce opțiuni de 2FA sunt disponibile pentru toți utilizatorii din sistem.

- Dacă utilizatorul dorește ulterior să schimbe metoda de 2FA (de exemplu, să treacă de la TOTP la notificări push sau la un token hardware), acesta poate face modificarea din setările contului său individual în Keycloak. Important de subliniat aici este ca aceasta schimbare este specifica doar contului utilizatorului respectiv si nu afecteaza alti utilizatori ai sistemului. Fiecare utilizator are controlul asupra metodei sale de 2FA în cadrul propriului cont. Dacă utilizatorul schimbă metoda de 2FA în Keycloak, schimbarea va fi aplicată pentru toate autentificările ulterioare în site-urile sau aplicațiile care folosesc Keycloak.
- Înregistrarea diverselor metode de 2FA:
  - ***TOTP***: Utilizatorul se autentifică în Keycloak și navighează către setările de securitate ale contului său. Utilizatorul selectează opțiunea de a configura TOTP. Keycloak generează un cod QR. Utilizatorul utilizează aplicatia android pentru a scana codul QR. Aceasta va lega contul utilizatorului de aplicația mobilă și va începe să genereze coduri TOTP. Utilizatorul introduce un cod TOTP generat de aplicația mobilă în Keycloak pentru a finaliza procesul de configurare.
  - ***Push Notification***: Utilizatorul descarcă aplicația mobilă care suportă notificări push pe dispozitivul mobil pentru sistemul Keycloak. Utilizatorul se autentifică în aplicația mobilă și își asociază contul de Keycloak, ceea ce poate implica scanarea unui cod QR. Odată configurată aplicația, utilizatorul va primi notificări push pentru autentificare.

FCM – Firebase Cloud Messaging

Pentru modulul de Push Notification o să fie nevoie și de un proiect Firebase pentru a ne folosi de serviciul FCM. Firebase Cloud Messaging (FCM) este un serviciu gratuit oferit de Google care permite dezvoltatorilor să trimită notificări push și mesaje de date către dispozitivele mobile și aplicațiile web, facilitând comunicarea în timp real între server și aplicații. În keycloak va trebui dezvoltat un SPI(Service Provider Interface) care să gestioneze autentificarea prin notificări push. Acest SPI va comunica cu FCM pentru a trimite notificări push către dispozitivele utilizatorilor. Prin intermediul SDK-ului Firebase se va genera la lansarea aplicației mobile un token FCM al dispozitivului care va trebui trimis la keycloak.Acest token va fi folosit de Keycloak împreună cu cheia de server FCM (obținută la configurarea SPI-ului) pentru a trimite request-uri către backend-ul FCM care mai departe va trimite notificări către device. [4] [5]

- ***Token Hardware***: Utilizatorul conectează tokenul hardware la dispozitiv. În setările contului Keycloak, utilizatorul selectează

opțiunea de a adăuga un token hardware ca metodă de 2FA. Procesul de înregistrare poate implica apăsarea unui buton pe token sau introducerea unui cod PIN, în funcție de tipul de token și de configurarea sistemului. Keycloak validează tokenul și îl asociază cu contul utilizatorului.

- Sistemul trebuie să permită autentificarea utilizând tokenuri hardware, prin intermediul unui serviciu compatibil, precum Nexus.

NEXUS

Nexus este o soluție care facilitează utilizarea tokenurilor hardware, cum ar fi smart carduri, pentru autentificarea la diverse servicii fără a necesita plugin-uri de browser sau alte extensii. Acesta acționează ca un middleware între tokenul hardware și aplicațiile care solicită autentificarea. [6]

Modul de funcționare:

- Utilizatorul instalează aplicația Nexus pe sistemul său. Aceasta este o aplicație desktop care rulează în background pe dispozitivul utilizatorului.
- Utilizatorul înregistrează tokenul hardware cu Nexus. Acest lucru se face o singură dată, iar Nexus stochează detaliile necesare pentru a comunica cu tokenul.
- Utilizatorul încearcă să acceseze un serviciu web protejat care necesită autentificare.
- Serviciul web redirecționează utilizatorul către Keycloak pentru autentificare.
- Utilizatorul introduce credentialele în interfața Keycloak. Aceste credențiale sunt validate de AD.
- După ce primul factor a fost efectuat cu succes, Keycloak inițiază cel de-al doilea factor (tokenul hardware).
- Keycloak comunică cu Nexus, printr-un API local sau un serviciu web, pentru a iniția procesul de autentificare cu tokenul hardware.
- Utilizatorul introduce tokenul hardware în portul USB și, dacă este necesar, introduce un PIN sau atinge senzorul de amprentă pentru activare.
- Pentru a finaliza autentificarea, tokenul este solicitat să semneze digital o anumită informație sau un „challenge” trimis de serverul de autentificare, keycloak (poate fi un mesaj generat de server sau o dată unică).
- Tokenul hardware, prin intermediul Nexus, generează un răspuns de autentificare. Acesta poate fi o semnătură digitală bazata pe o cheie privată stocată în siguranță pe token. Tokenul hardware conține de obicei o cheie privată care este stocată în mod securizat și nu poate fi extrasă din dispozitiv. Cheia privată este folosită pentru a genera semnături. De asemenea, acesta poate stoca un certificat digital asociat cu cheia privată. Certificatul include cheia publică corespunzătoare și este emis de o autoritate de certificare de încredere.
- Nexus trimite răspunsul de autentificare(semnatura challenge-ului) înapoi la Keycloak.
- Keycloak validează răspunsul primit. Semnătura digitală este verificată folosind cheia publică din certificatul asociat cheii private. Dacă semnătura este validă, autentificarea este considerată un succes.

- Utilizatorul este acum autentificat complet și Keycloak îi permite accesul la serviciul web protejat.

    - ***Cerințe de securitate:***

        - Sistemul trebuie să mențină un nivel înalt de securitate în timpul integrării și comunicației cu Nexus pentru gestionarea tokenurilor hardware, asigurând criptarea completă a datelor și a cererilor de autentificare între Nexus și serverul Keycloak, precum și între Nexus și tokenurile hardware ale utilizatorilor.
        - Cheia privată de pe tokenul hardware nu este niciodată expusă sau transmisă în afara dispozitivului, ceea ce asigură un nivel înalt de securitate.
        - Folosirea unui reverse proxy in fata serverului Keycloak va oferi un strat suplimentar de securitate, protejând Keycloak de accesul direct neautorizat și posibile atacuri. În plus, utilizarea unui reverse proxy va permite o gestionare mai eficientă a traficului, echilibrarea încărcării și capacitatea de a aplica reguli de filtrare a traficului și de protecție împotriva atacurilor de tip DDoS.
        - Pentru procesul de autentificare și autorizare, sistemul va implementa standardul OAuth 2.0, asigurând un mecanism securizat și eficient pentru gestionarea accesului la resursele API. Pentru autentificarea și identificarea utilizatorilor, sistemul va adopta protocolul OpenID Connect, care extinde OAuth 2.0, oferind o soluție standardizată pentru autentificarea end-to-end a utilizatorilor în diverse aplicații și servicii. Această abordare va asigura conformitatea cu cele mai bune practici în domeniul securității și va facilita integrarea sigură și eficientă cu diverse servicii externe.

    - ***Cerințe privind tehnologiile folosite:***

        - ***<u>Compatibilitate și Interoperabilitate:</u>*** Nexus asigură compatibilitatea și interoperabilitatea extinsă între diverse tipuri de tokenuri hardware și sistemul Keycloak, facilitând o integrare fără probleme și eficientă, ceea ce permite utilizatorilor să utilizeze o gamă largă de tokenuri hardware pentru autentificarea multi-factor.
        - ***<u>Eficiența și Fiabilitatea Notificărilor Push:</u>*** Folosirea FCM în proiect asigură o metodă eficientă și fiabilă pentru trimiterea notificărilor push către dispozitivele mobile ale utilizatorilor. Datorită infrastructurii robuste și scalabile oferite de Google, FCM permite o distribuție rapidă și sigură a mesajelor, esențială pentru promptitudinea și fiabilitatea procesului de [4] autentificare multi-factor. Implementarea de la zero a

unui astfel de serviciu nu ar dispune de caracteristicile avansate a serviciilor deja existente. FCM gestionează aspecte complexe ale livrării notificărilor, cum ar fi gestionarea dispozitivelor offline și reîncercările de livrare, lucru care ar fi complicat și consumator de timp dacă ar trebui implementat manual pentru fiecare sistem.

- ***Centralizarea și Simplificarea Gestionării Autentificării***: Keycloak oferă o soluție centralizată și eficientă pentru gestionarea autentificărilor și identităților utilizatorilor, facilitând administrarea accesului și a politicilor de securitate într-o manieră uniformă și coerentă. Aceasta permite o integrare simplă cu diverse sisteme și aplicații, îmbunătățind securitatea și reducând complexitatea administrativă asociată cu gestionarea mai multor sisteme de autentificare separate.
- ***Uniformizarea și Securitatea Gestionării Credențialelor:*** Folosirea Active Directory de la Microsoft asigură o gestionare centralizată și securizată a credențialelor și identităților utilizatorilor în cadrul organizației. Aceasta facilitează autentificarea unificată și gestionarea accesului, permițând integrarea eficientă cu diverse servicii și aplicații, inclusiv cu Keycloak, și asigurând conformitatea cu standardele de securitate corporative și industriale.
- Utilizarea PostgreSQL ca bază de date pentru Keycloak din mai multe considerente (recomandata si in documentatia oficiala Keycloak [3]):
    - oferă caracteristici avansate de securitate, cum ar fi criptarea la nivel de coloană și suportul pentru SSL, esențiale pentru protejarea datelor sensibile gestionate de Keycloak.
    - având o capacitate excelentă de scalabilitate, PostgreSQL poate susține cu ușurință creșterea volumului de date și a numărului de tranzacții, adaptându-se nevoilor în evoluție ale sistemului Keycloak.
    - fiind una dintre bazele de date recomandate de documentația Keycloak, PostgreSQL este testat și optimizat pentru a lucra eficient cu Keycloak, asigurând astfel o integrare fără probleme și minimizând riscul de incompatibilități.
    - postgreSQL beneficiază de o comunitate extinsă și de un suport larg, oferind acces la resurse ample și asistență în cazul oricăror provocări tehnice.

- ***Cerințe frontend:***

Pentru crearea interfeței site-ului web de test se va folosi React împreună cu Bootstrap:

- Combinarea React pentru partea de cod și Bootstrap pentru design ne ajută să construim rapid pagini web moderne și practice.
- Bootstrap vine cu componente deja implementate: butoane, formulare, card-uri etc. și asigura o scalabilitate ferestrelor prin sistemul de grid flexibil.

- ***Cerințe backend:***

Utilizarea Spring Boot împreună cu Java 17 si Maven pentru gestionarea proiectului, constituie o abordare solidă și eficientă pentru dezvoltarea backendului și a modulelor pentru Keycloak:

- Folosind Java 17, versiunea recentă și stabilă a limbajului Java, beneficiem de îmbunătățiri ale performanței, securității și suportul pentru caracteristici moderne ale limbajului.
- Spring Boot oferă o platformă robustă și flexibilă pentru dezvoltarea rapidă a aplicațiilor. Cu configurare automată și management simplificat al dependențelor, accelerăm dezvoltarea și reducem timpul necesar pentru punerea în funcțiune a aplicației.
- Dat fiind că Keycloak este scris în Java, utilizarea Java 17 pentru dezvoltarea modulelor asigură o integrare fără probleme și o compatibilitate optimă.
- Având o comunitate extinsă și o mulțime de resurse disponibile, Spring Boot asigură acces la o varietate mare de ghiduri și cele mai bune practici, facilitând rezolvarea problemelor și accelerarea dezvoltării.
- Spring Boot, împreună cu Spring Security, oferă un cadru solid pentru implementarea securității, inclusiv pentru protecția endpoint-urilor și gestionarea autentificării și autorizării.

- ***Cerințe aplicație android:***

Dezvoltarea aplicației android în Kotlin prin Android Studio:

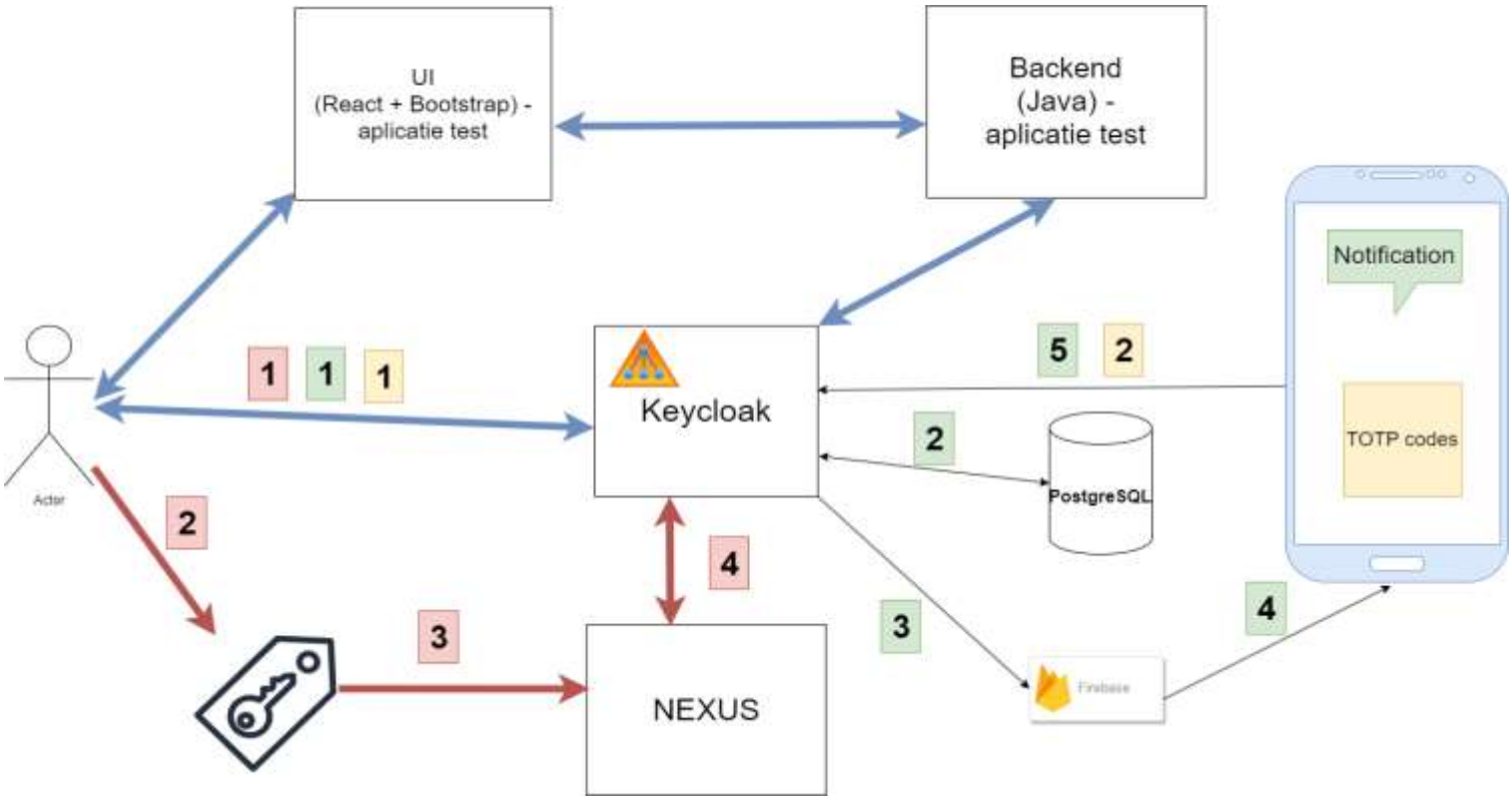
- Kotlin este susținut complet de Android Studio, oferind integrare nativă și acces la toate funcționalitățile Android SDK.
- Kotlin este complet interoperabil cu Java, permițând utilizarea bibliotecilor Java existente și integrarea ușoară cu codul Java existent.
- Fiind un limbaj modern, Kotlin continuă să fie actualizat și îmbunătățit, oferind cele mai recente caracteristici și optimizări pentru dezvoltarea aplicațiilor mobile.

3. Arhitectura sistemului

Sistemul este construit în jurul serverului Keycloak, care servește ca punct central de autentificare și autorizare, folosind Active Directory (AD) pentru gestionarea credențialelor utilizatorilor. Procesul de autentificare este întărit cu un strat de securitate suplimentar prin utilizarea autentificării cu doi factori (2FA), care include opțiuni precum TOTP, notificări push și tokenuri hardware.

Aplicația Android, dezvoltată în Kotlin și integrată cu Keycloak, generează TOTP și gestionează notificările push. Backend-ul, implementat folosind Spring Boot și Java 17, comunică cu Keycloak pentru a valida tokenurile și a accesa resursele protejate.

Această arhitectură sprijină o interfață de utilizator reactivă și o experiență fluidă pentru utilizatori, în timp ce menține securitatea datelor și integritatea sistemului. Scalabilitatea este asigurată prin tehnologii cloud-ready, iar gestionarea erorilor este încorporată pentru a asigura notificări adecvate utilizatorilor și administratorilor.



Fluxul rosu: reprezintă fluxul de autentificare prin token hardware. Se face mai întâi autentificarea prin keycloak (primul pas – autentificarea primară - 1), după utilizatorul introduce

tokenul hardware în device (2). La pasul (3) nexus preia informațiile necesare de pe token (cheia privată + certificat digital) pentru a face semnarea unui challenge ce va fi trimis de keycloak la pasul 4. Semnătură challenge-ului se va trimite la keycloak împreună cu certificatul digital tot la pasul (4). Keycloak va face verificarea semnăturii și în cazul în care este valida va returna user-ului tokenii necesari accesării resurselor protejate.

Fluxul verde: reprezintă fluxul de autentificare prin push notification. Se face la fel, mai întâi autentificarea prin keycloak (1), după keycloak va prelua din baza de date postgreSQL tokenul FCM asociat utilizatorului. Va trimite acest token FCM către serviciul FCM împreună cu mesajul notificării. Serviciul de FCM va trimite notificarea către device-ul mobil. Utilizatorul va introduce codul afișat din interfața web grafică în notificare și va da Approve. Răspunsul se va duce la keycloak care va verificq codul și dacă acesta este valid va returna tokenii.

Fluxul galben: se va efectua primul pas de autentificare (1) și după utilizatorul va prelua din aplicația mobilă codul TOTP corespunzător site-ului dorit și îl va introduce în interfața web. Keycloak va verifica codul TOTP (va genera și el acest cod TOTP și îl va verifica cu cel primit de la user => acest lucru este posibil deoarece aplicația mobilă și serverul sunt conectați la un server ntp și sunt sincronizați). Dacă codul este valid, keycloak va returna tokenii.

Pentru a facilita pornirea și gestionarea componentelor sistemului meu, voi folosi Docker. Voi crea fișiere Dockerfile pentru fiecare componentă, cum ar fi Keycloak, backend-ul Spring Boot și orice alte servicii necesare. Apoi, voi defini un fișier *docker-compose.yml* care să orchestreze toate serviciile, setându-le rețeaua, volumul și dependențele necesare. Acest lucru va permite pornirea întregului stack de aplicații printr-o singură comandă, ‚docker-compose up', economisind timp și asigurând o uniformitate a mediului de rulare indiferent de platforma pe care o folosesc, fie că este dezvoltare, testare sau producție.

4. Diagrame UML

4.1. Diagrama pentru cazurile de utilizare

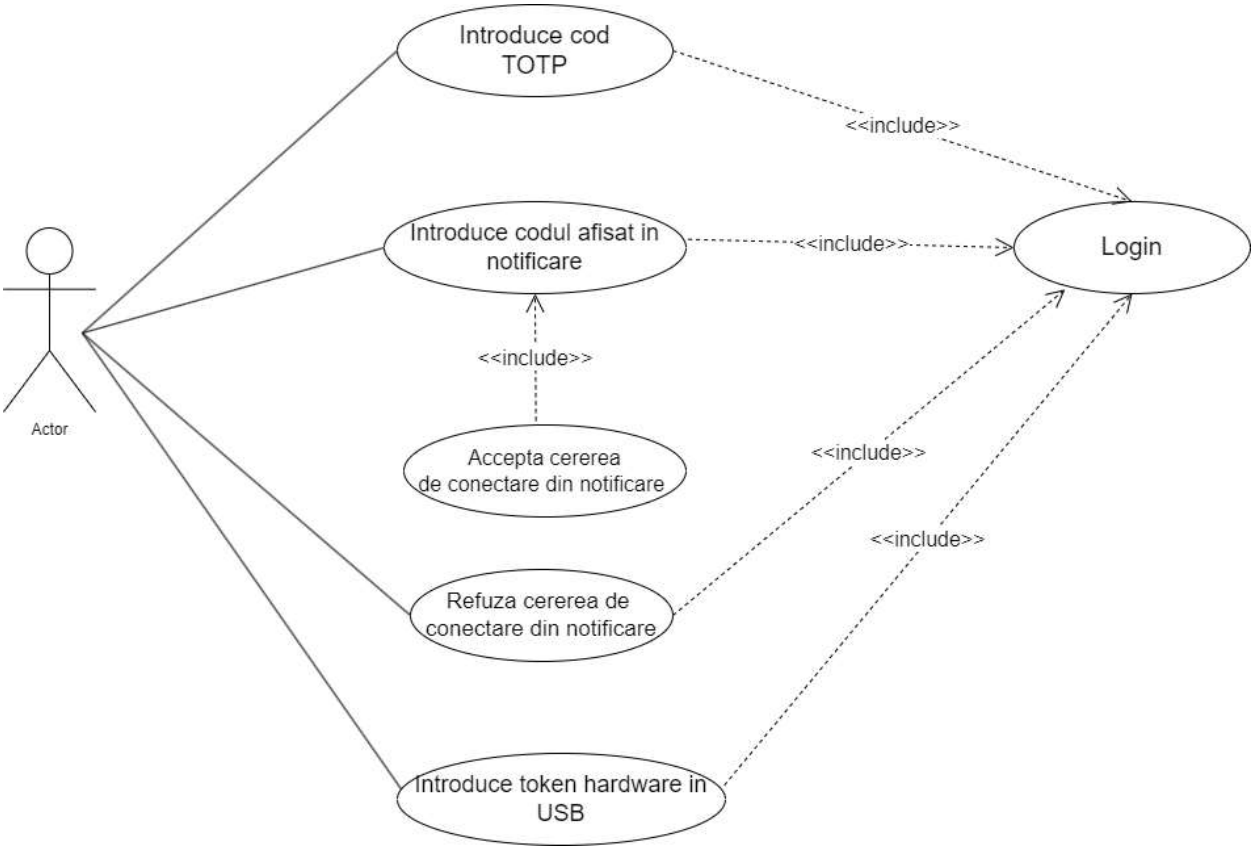


Fig.2. Diagrama cazurilor de utilizare pentru un utilizator obișnuit din AD

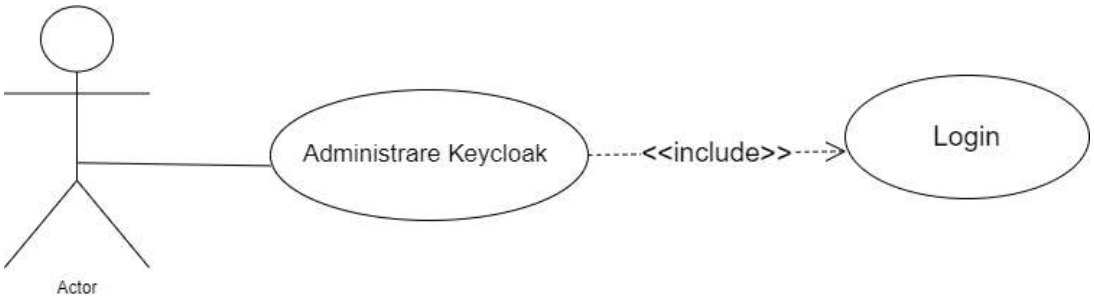


Fig.3. Diagrama cazurilor de utilizare pentru administrator

4.2. Diagrama de activitate

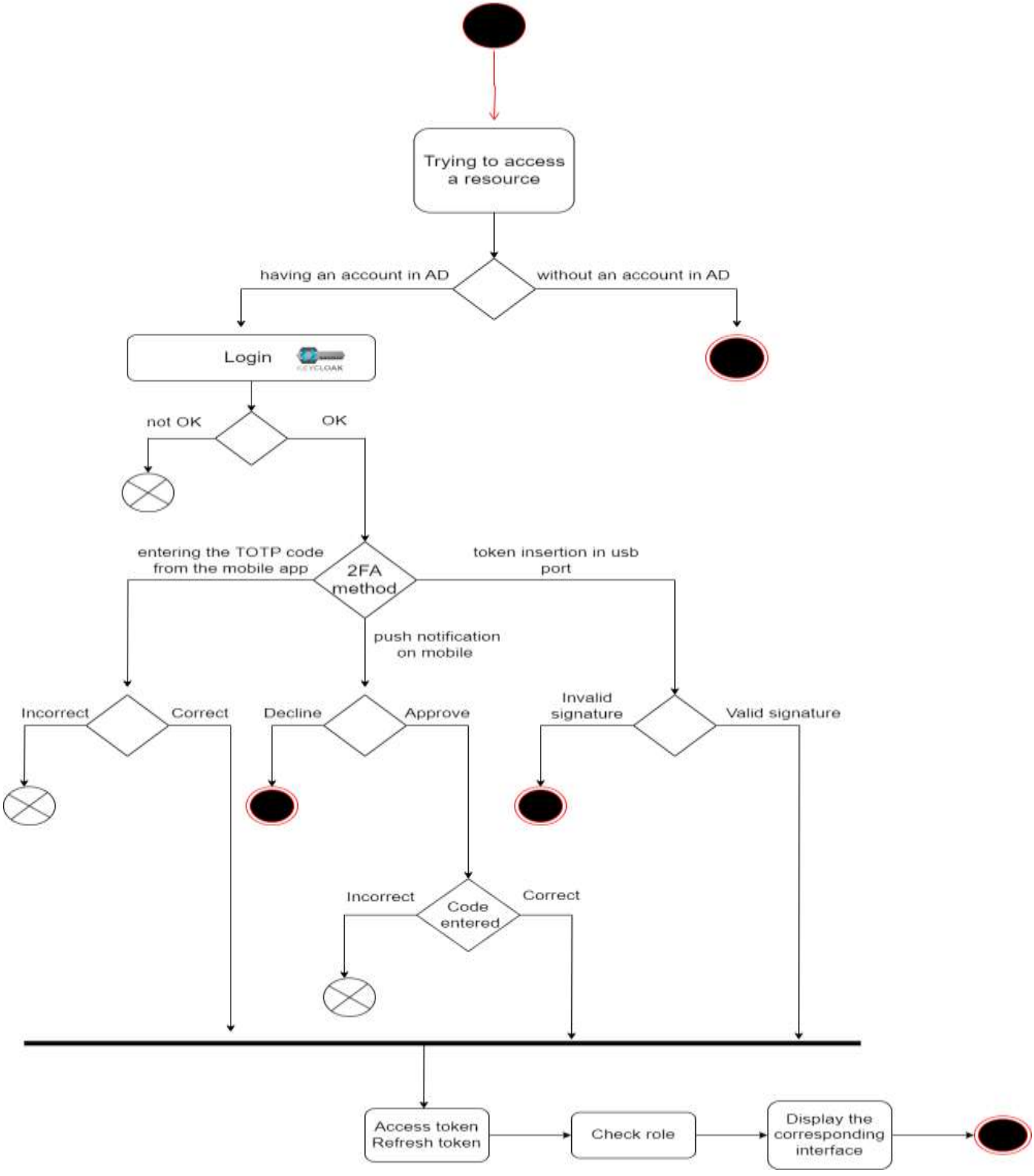


Fig.4. Diagrama de activitate

4.3. Diagrame de secvență

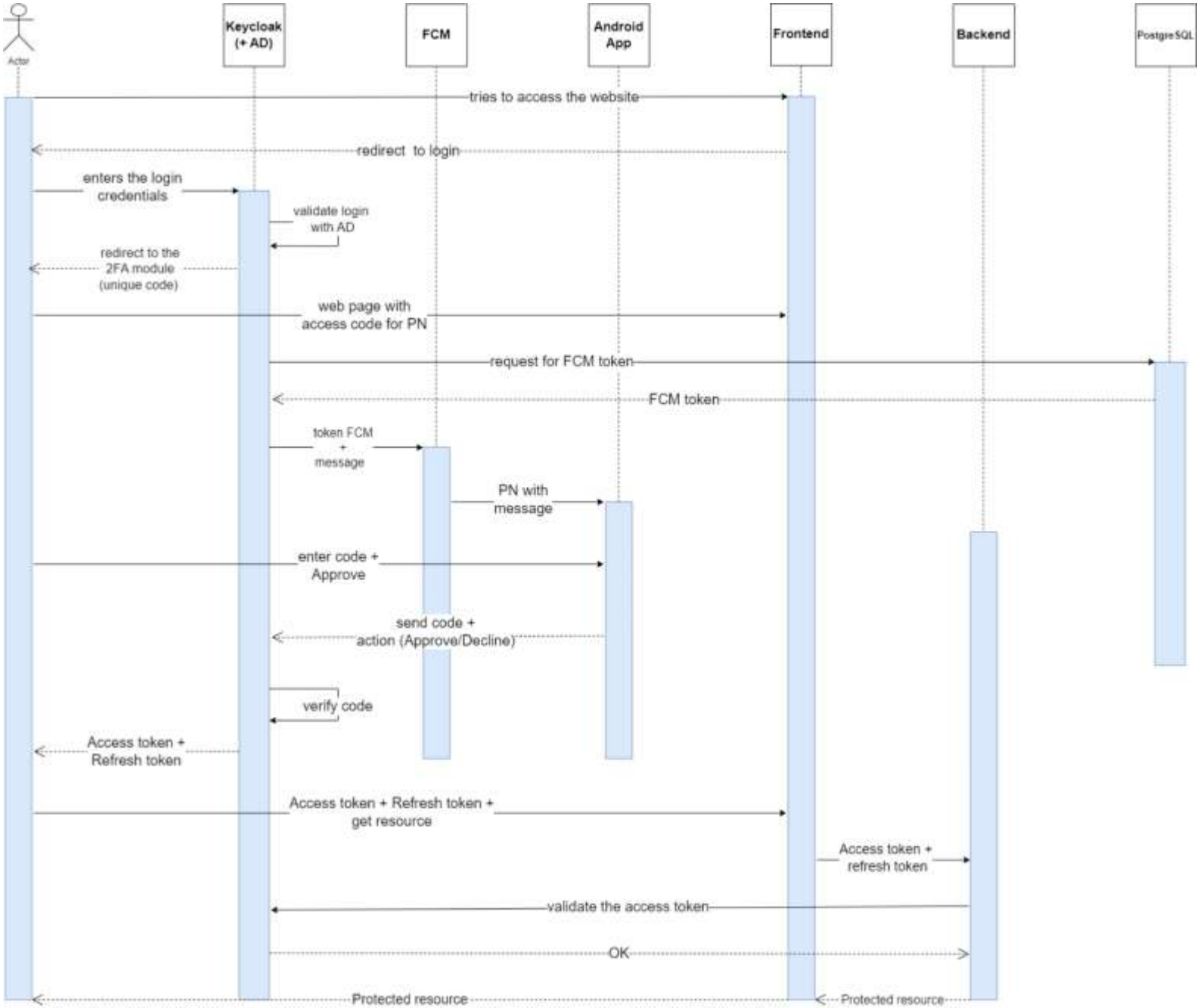


Fig.5. Diagrama de secvență pentru PN

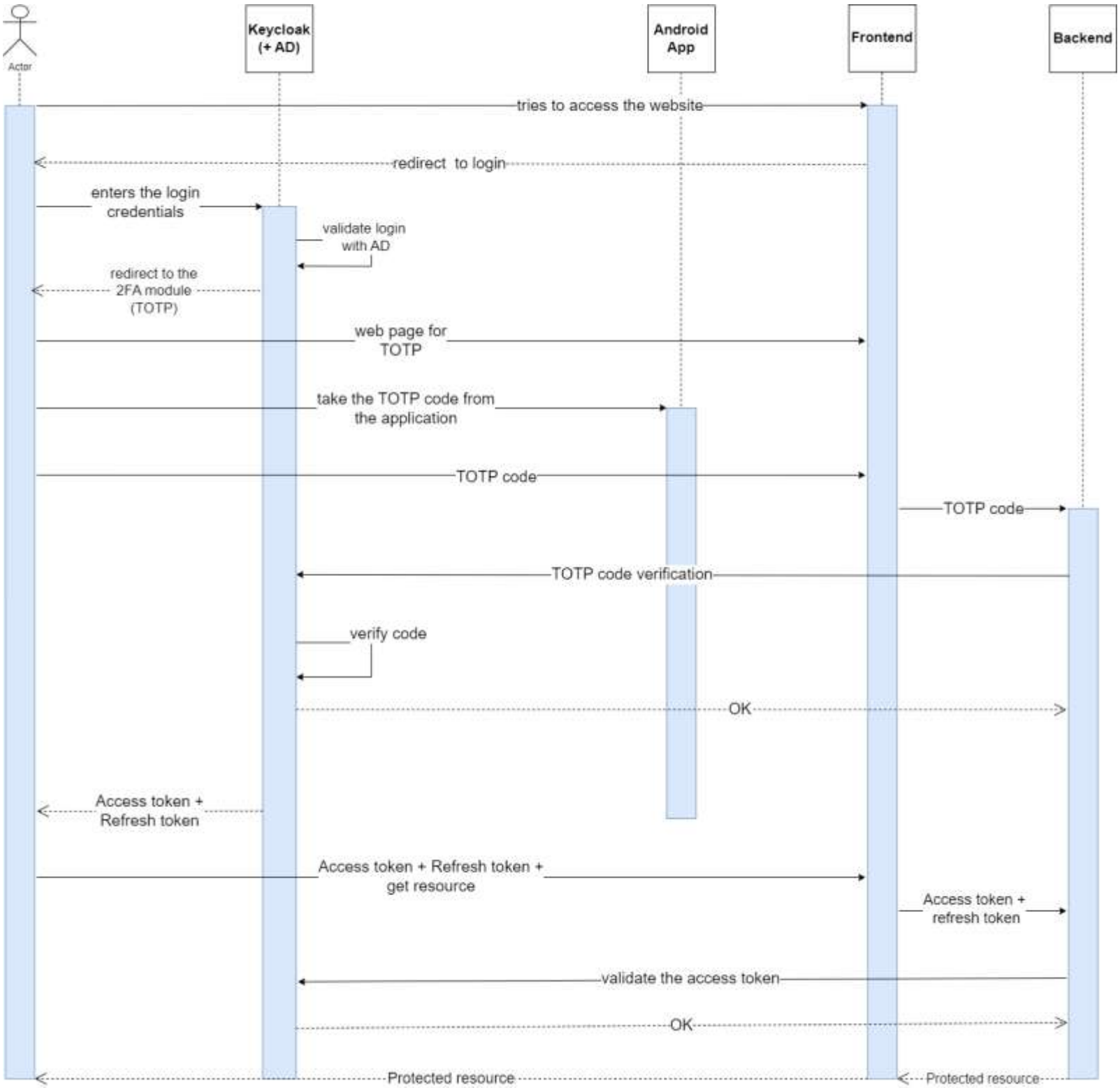


Fig.6. Diagrama de secvență pentru TOTP

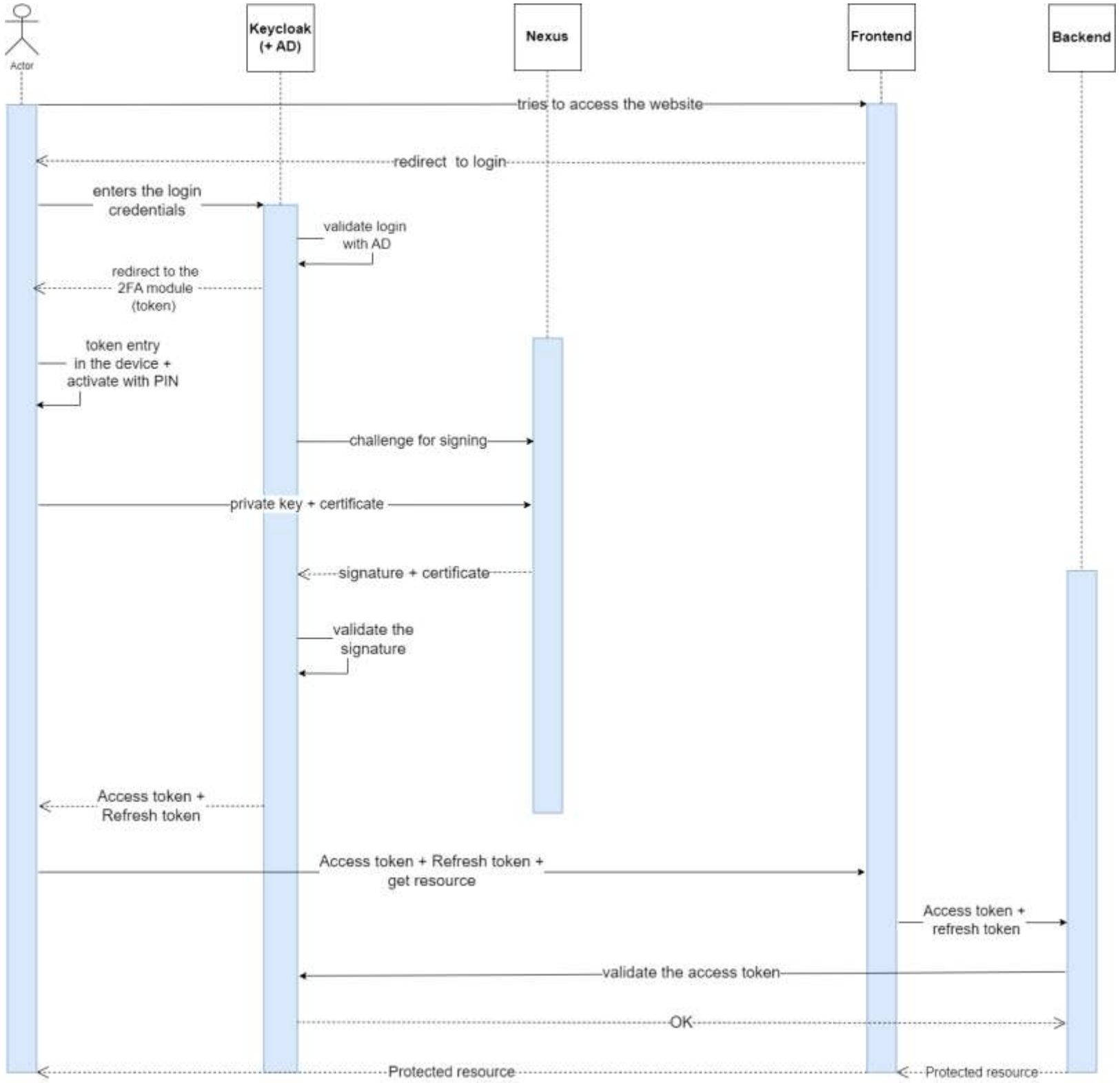


Fig.7. Diagrama de secvență pentru Token Hardware

5. Plan de testare

Teste funcționale la nivelul sistemului

Test	Scop	Pași	Rezultate Așteptate
1.Autentificare prin Keycloak	Redirectionare la pagina de log-in Keycloak. Verifică credentialele prin AD.	1.Introducere credențiale. 2.Trece la cel de-al doilea factor.	Acces permis după autentificare reușită
2.Configurare metodă de 2FA folosita	Alegerea metodei de 2FA de către utilizator.	1.Userul se conectează la contul Keycloak. 2. Merge în setările contului și alege o metodă de 2FA.	Opțiunea 2FA este setată pentru cont => următorul pas după autentificarea primară să fie metoda 2FA setată.
3.Configurare TOTP	Configurarea TOTP în Keycloak.	1.Scanare QR code. 2.Validare TOTP.	TOTP este configurat și funcțional.
4.Generare cod TOTP	Generarea unui cod TOTP in aplicația mobilă.	1.Deschidere aplicație. 2.Generare cod.	Cod TOTP generat în aplicație.
5.Verificare cod TOTP	Verificarea codului TOTP introdus.	1.Introducere cod în interfața web. 2. Confirmare cod.	Keycloak verifică codul introdus și acceptă codul dacă acesta este valid.
6.Primire notificari push	Primirea notificărilor push pe dispozitivul mobil.	1.Așteptare notificare. 2.Verificare recepție.	Afisarea notificarii push pe dispozitivul mobil ca o notificare pop-up.
7. Autentificare Push Notification	Autentificarea prin introducerea unui cod din interfață web în notificare și aprobarea acesteia.	1.Introducere cod. 2.Aprobare notificare. 3.Confirmare.	Utilizatorul este autentificat prin push.

8.Decline la Push Notification	Verificarea butonului de Decline din notificare.	1.Userul primește notificarea push. 2.Apasă butonul Decline.	Utilizatorul a refuzat autentificarea prin push.
9.Conectare Token Hardware	Conectarea tokenului hardware la sistem.	1.Inserare token. 2.Așteptare recunoaștere.	Tokenul hardware este recunoscut de sistem.
10. Autentificare Token Hardware	Autentificarea folosind un token hardware.	1.Generare răspuns nexus. 2.Confirmare semnatura de catre keycloak.	Sistemul validează tokenul.
11. Emitere token-uri	Emiterea unui access token și a unui refresh token.	1. Utilizatorul se autentifica în doi pași cu succes. 2.Keycloak emite token-uri.	Token-uri emise și valide.
12. Accesare resurse protejate	Accesarea resurselor protejate cu access token valid.	1.Introducere token in header-ul de autorizare. 2.Cerere resursă.	Resursa protejată este accesată.
13. Dezactivare 2FA	Dezactivarea metodei de 2FA.	1.Alegere dezactivare. 2.Confirmare dezactivare.	2FA este dezactivat pentru utilizator.
14. Reautentificare	Reautentificarea după expirare sesiune.	1.Reintroducere credențiale. 2.Modul de 2FA. 3. Confirmare autentificare.	Utilizatorul este reautentificat cu succes.
15. Schimbare metodă 2FA	Schimbarea metodei de 2FA.	1.Alegere nouă metodă de 2FA din setarile contului Keycloak.	Metoda 2FA este schimbată cu succes.

		2.Confirmare schimbare.	
16. Validare Refresh Token	Utilizarea refresh token-ului pentru a obține un nou access token.	1.Introducere refresh token. 2.Cerere token nou.	Un nou access token este obținut.
17. Logout	Deconectarea utilizatorului din sistem.	1.Apăsare buton logout. 2.Confirmare logout.	Utilizatorul este deconectat (nu mai are un access token/refresh token valid).
18. Verificare rolului utilizatorului pentru autorizare	Verificarea rolului utilizatorului.	1.Cerere resursă protejată. 2.Verificare rol pentru accesul la resursă.	Returnare resursă daca rolul este corespunzător.
19. Audit Securitate	Verificarea logurilor de securitate pentru activitatea recentă.	1.Accesare loguri. 2.Analiză activitate.	Activitatea este înregistrată și verificată.

Teste axate pe gestionarea erorilor

Test	**Scop**	**Pași**	**Rezultate Așteptate**
1. Eroare autentificare AD	Verificarea răspunsului sistemului la credențiale incorecte.	1.Introducere credențiale greșite. 2.Tentativă de login.	Introducere credențiale greșite. Tentativă de login.
2. Cod TOTP incorect/expirat	Testează validarea codurilor TOTP greșite.	1.Introducere cod TOTP expirat/invalid.	Mesaj de eroare pentru cod TOTP incorect. Blocare acces.
3. Respingere Notificare Push	Verificarea comportamentului sistemului la refuzarea notificării push.	1. Apasă butonul Decline în notificarea push.	Confirmare respingere. Cerere nouă de autentificare.
4. Conexiune eșuată Token Hardware	Testează sistemul la deconectarea tokenului hardware.	1. Deconectare token în timpul autentificării.	Mesaj de eroare. Solicitare reconectare token.
5. Expirare Refresh Token	Verificarea comportamentului sistemului când un refresh token expiră.	1. Utilizare refresh token expirat pentru acces.	Mesaj de eroare. Solicitare reautentificare. Redirectare pe pagina de log-in.
6. Indisponibilitate Keycloak	Simulare cădere a serverului Keycloak.	1. Încercare de acces în timpul indisponibilității.	Mesaj de eroare server. Instrucțiuni de urmat.

6. Bibliografie

[1] „Pretius-Keycloak sso," [Interactiv]. Available: https://pretius.com/blog/keycloak-sso/.

[2] „GitHub-Aegis," [Interactiv]. Available: https://github.com/beemdevelopment/Aegis/blob/master/docs/vault.md.

[3] „PostgreSQL," [Interactiv]. Available: https://www.keycloak.org/2022/02/dbs.

[4] „FCM," [Interactiv]. Available: https://firebase.google.com/docs/cloud-messaging.

[5] „Google-cloud," [Interactiv]. Available: https://www.googlecloudcommunity.com/gc/Cloud-Forums/ct-p/cloud-forums.

[6] „Nexus-documentation," [Interactiv]. Available: https://doc.nexusgroup.com/display/PUB/Set+up+hardware+token.

[7] „Nexus," [Interactiv]. Available: https://help.sonatype.com/repomanager3/integrations.

[8] „Keycloak-documentation," [Interactiv]. Available: https://www.keycloak.org/documentation.html.