Progetto - Ingegneria del Software

IdP con protocollo OAuth2.0 e 2FA

DOCUMENTAZIONE UML – Gruppo 14

Corso Ingegneria del Software e Fondamenti Web

Docente Mongiello Maria

Mentore progetto Dell'Olio Giacomo Alberto

Componenti

Putignano Gianluca	MAT – 588237	mail – g.putignano11@studenti.poliba.it
Raimondi Federico	MAT – 587660	mail - f.raimondi@studenti.poliba.it
Salluce Luca	MAT – 587863	mail – l.salluce@studenti.poliba.it
Troilo Stefano	MAT – 588045	mail – s.troilo2@studenti.poliba.it
Volpe Antonio	MAT – 588757	mail – a.volpe18@studenti.poliba.it

DOCUMENTAZIONE UML – Gruppo 14

Scopo della documentazione

Lo scopo del nostro progetto (Identity Provider e OAuth2.0) è quello descrivere le funzionalità del sistema 2FA mostrando anche la dinamicità del design, evidenziando:

- Come gli utenti si autenticano utilizzando le credenziali e un secondo fattore di riconoscimento;
- La struttura tecnica delle componenti del sistema;
- Il flusso di comunicazione tra i moduli;

Obiettivi principali

1. **Sicurezza avanzata**
    1.1. aggiunge un secondo livello di verifica oltre alla tradizionale password;
    1.2. migliora la protezione contro attacchi comuni come phishing, brute force ecc.
2. **Identificazione robusta**
    2.1. autentica l'identità in base qualcosa che l'utente conosce o possiede;

Funzionalità principali

1. **Gestione credenziali**
    a. Registrazione e memorizzazione di username e password
    b. Hashing delle password (con algoritmi come BCRYPT)
2. **Integrazione secondo fattore**
    a. Invio OTP
    b. E-mail
    c. Validazione OTP inserito dall'utente

Benefici del sistema

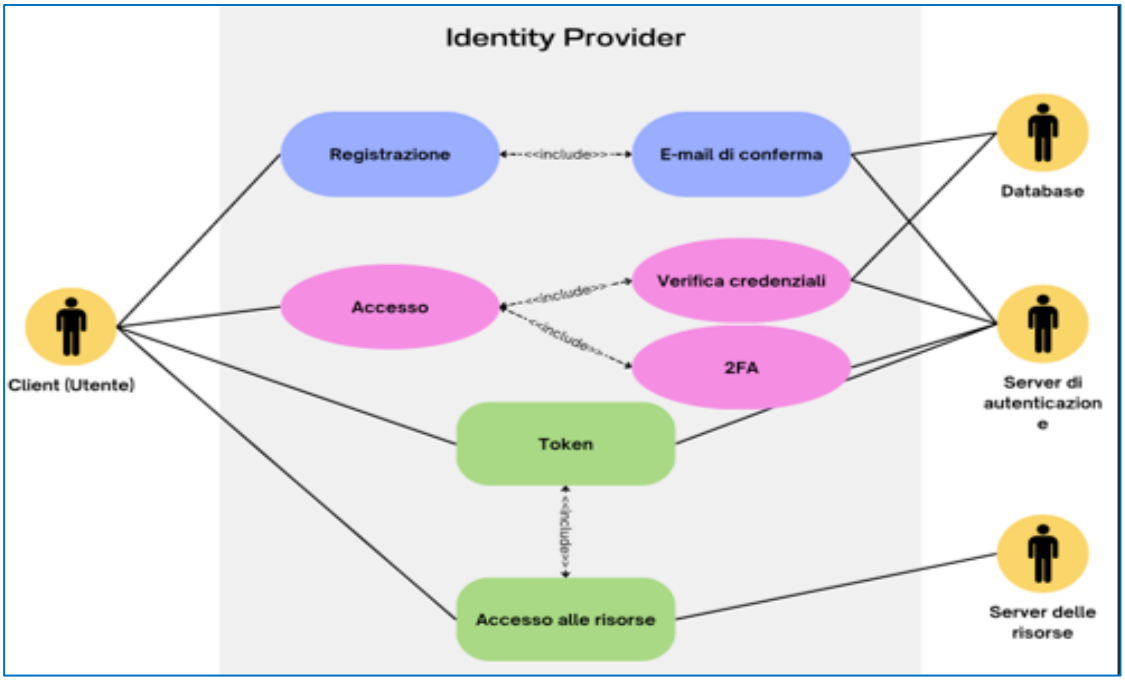
1. **Maggiore protezione degli account**
    a. Anche se un utente ha una password facilmente ricavabile, il secondo fattore riduce comunque il rischio di comprometterla
2. **Facilità d'utilizzo**
    a. L'uso del 2FA è reso accessibile anche ad utenti meno esperti
3. **Riduzione delle violazioni di sicurezza**

a. La combinazione di credenziali e un secondo fattore rende il sistema più difficile da aggirare

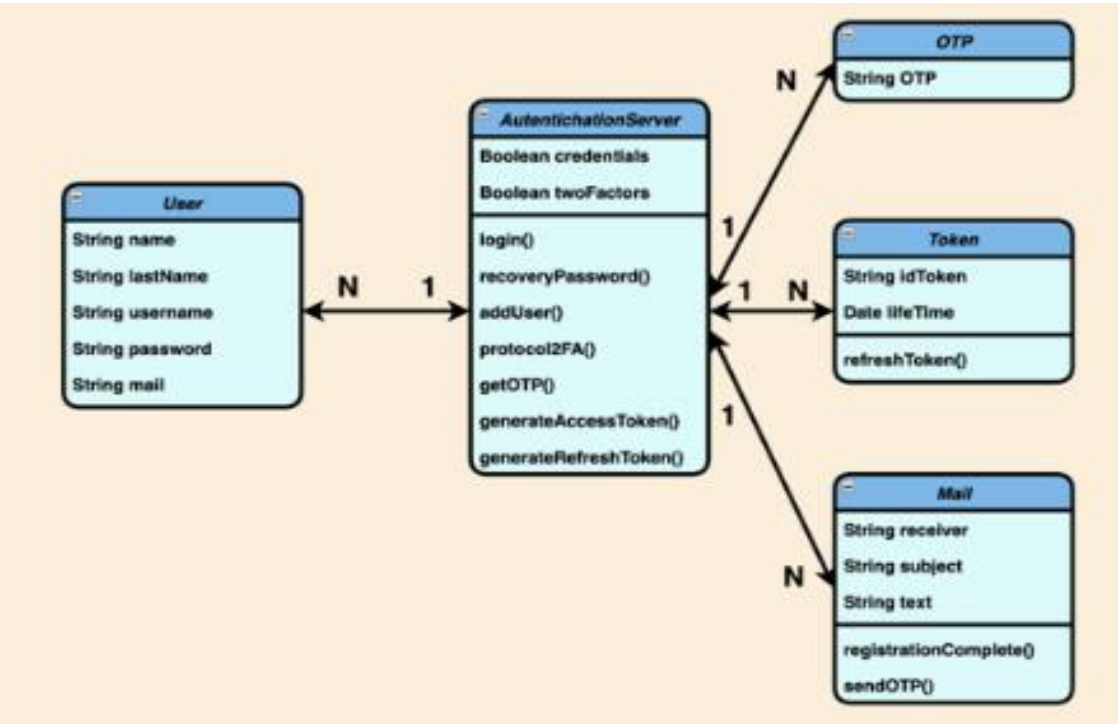
4. **Flessibilità e scalabilità**
    a. Il sistema può esser integrato in molteplici contesti come il semplice accesso ai siti web, applicazioni aziendali, piattaforme bancarie ecc.

Tipologie di diagrammi UML

1. Il **diagramma dei casi d'uso** descrive le interazioni tra gli attori e il sistema. Gli elementi principali sono:
    - Attori:
        - Client (Utente)
        - Database
        - Server di autenticazione
        - Server delle risorse
    - Casi d'uso:
        - Registrazione
        - Accesso con verifica delle credenziali tramite 2FA
        - Invio e-mail di conferma
        - Assegnazione token
        - Accesso alle risorse

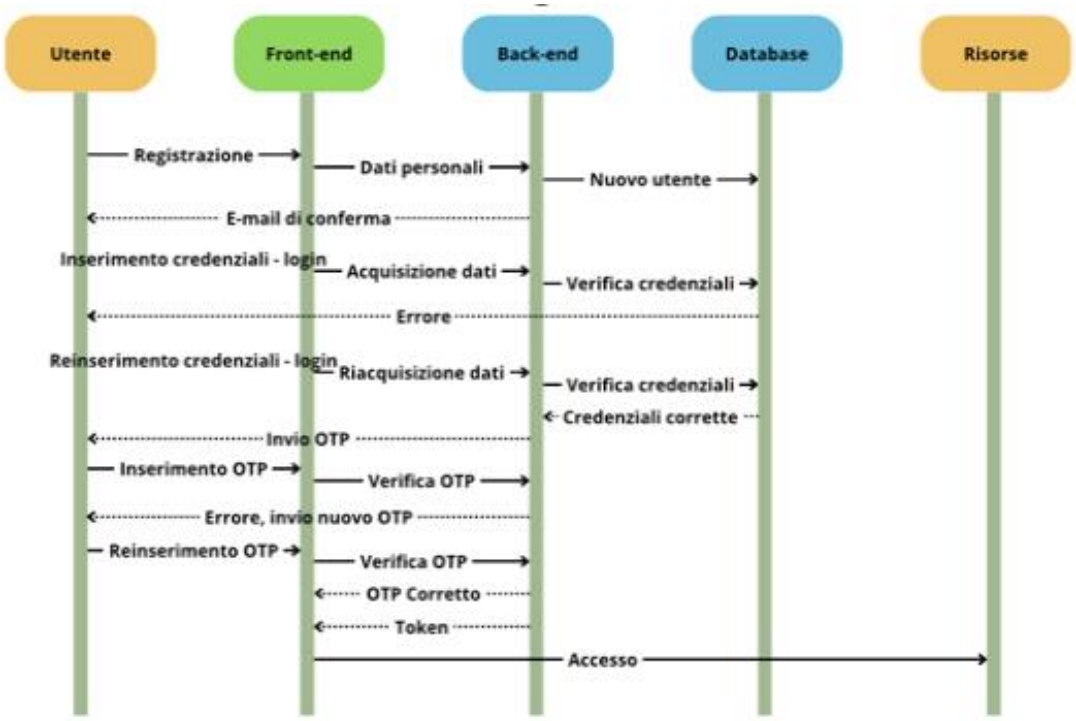


2. Il **diagramma delle classi** mostra la struttura del sistema, rappresentando le entità coinvolte e le relazioni che intercorrono tra di esse. È costituito da:
    - Classi principali:
        - Authentication Server:
            - attributi (credenziali, twoFactors)
            - metodi (login, recupero password, aggiungi utente, protocollo 2FA, ottieni OTP, genera token d'accesso, genera token di refresh)
        - User:
            - attributi (nome, cognome, username, password, mail)
        - OTP:
            - attributi (OTP)
        - Token:
            - attributi (idToken, life-time)
            - metodi (refresh token)
        - Mail:
            - attributi (destinatario, oggetto, testo)
            - metodi (registrazione completata, invio OTP)
    - Relazioni:
        - L'utente è associato alla classe server 2FA
        - Il server 2FA è associato alle classi OTP, Token e Mail



3. Il **diagramma di attività** rappresenta il flusso di processo per l'autenticazione 2FA. Le attività principali sono:
    - L'utente inserisce le credenziali
    - Le credenziali vengono verificate dal sistema
    - Il sistema invia il codice OTP
    - Viene effettuata la validazione del codice
    - Viene consentito o negato l'accesso

4. Il **diagramma delle sequenze** descrive l'interazione temporale tra le componenti del sistema durante l'autenticazione. Il diagramma delle componenti invece associa alle componenti i propri moduli

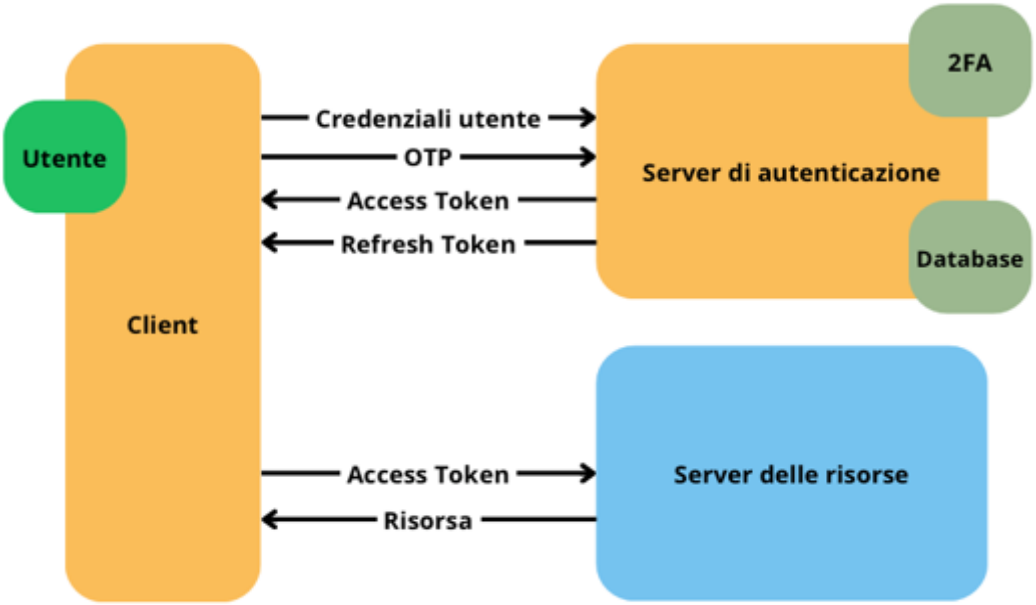


    - Utente: inserisce le informazioni richieste
    - Front-end: riguarda il modulo di login
    - Back-end: contiene la logica di autenticazione
    - Database: mantiene in memoria le credenziali e le configurazioni
    - Risorse: qualsiasi applicazione esterna

5. Il **diagramma architetturale** è una rappresentazione grafica che descrive la struttura, le componenti principali e le relazioni interne del sistema. Serve ad

osservare come le varie parti del sistema comunicano tra loro e come sono organizzate. Il diagramma prevede:

- **Chiarezza**: aiuta a spiegare il funzionamento del sistema a diversi stakeholder (sviluppatori o progettisti) in modo semplice e intuitivo.
- **Pianificazione**: aiuta ad individuare i problemi e i conflitti prima della costruzione
- **Documentazione**: serve come forma di documento per manutenzione future o passaggi di consegne.
- **Analisi e Miglioramento**: permette di identificare le aree di miglioramento e può essere utilizzato per analizzare la scalabilità, la sicurezza e le prestazioni.
- **Complessità**: divide il sistema in parti più piccole, quindi più semplici, per evitare errori.



Appendici

- Glossario:
  - OTP: One Time Password
  - 2FA: Autenticazione a 2 fattori
- Risorse esterne:
  - Modulo Python: mail Service (SMTP)
  - MySQL connection
  - FLASK-Cors
  - Modulo BCRYPT
  - Modulo SMTPLIB