



# **Assignment 02**

## **Project UML Sec Diagrams**

### **Secure Software Design**

#### **Team Members:**

- Uzair Zahid (22i-1619)
- Haziq Hussain (22i-1685)
- Hussain Abdullah (22i-1639)
- Hassaan Siddiqui (22i-1642)



# **Password-less Authentication System using One-Time Passwords (OTPs)**

## **0. Requirements Definition with Applied Framework**

### **0.1 Functional Requirements**

- **FR1:** User Registration - System allows users to register with email/username credentials
- **FR2:** OTP Request Generation - Users can request one-time password for authentication
- **FR3:** OTP Validation - System validates submitted OTP within specified time limit
- **FR4:** Secure Session Management - Create and maintain secure user sessions post-authentication
- **FR5:** Protected Resource Access - Authenticated users can access authorized system resources
- **FR6:** Secure Logout - Users can safely terminate active sessions

### **0.2 Non-Functional Requirements**

- **NFR1:** Performance - OTP generation and validation completed within 2 seconds
- **NFR2:** Scalability - System supports minimum 1000 concurrent user sessions
- **NFR3:** Availability - System maintains 99.9% uptime availability
- **NFR4:** Usability - Authentication process completed in maximum 3 user interactions
- **NFR5:** Compatibility - Cross-platform browser support (Chrome, Firefox, Safari, Edge)
- **NFR6:** Reliability - Zero data loss during authentication processes



## National University of Computer and Emerging Sciences Islamabad Campus

---

### 0.3 Domain Requirements

- **DR1:** Email Integration - SMTP server integration for OTP delivery mechanism
- **DR2:** Database Management - Secure relational database for user data and OTP storage
- **DR3:** Web Application Interface - Responsive web-based user interface
- **DR4:** API Architecture - RESTful API endpoints for client-server communication
- **DR5:** Cryptographic Operations - Secure random number generation for OTP creation

### 0.4 Security Requirements

- **SR1:** Authentication Security
  - OTP validity period: Maximum 3 minutes lifespan
  - Cryptographically secure OTP generation using approved algorithms
  - Single-use OTP policy enforcement
- **SR2:** Access Control
  - Session-based authorization mechanisms
  - Role-based access control implementation
- **SR3:** Data Protection
  - TLS 1.3 encryption for all client-server communications
  - Encryption at rest for sensitive data storage
  - Secure session token generation and management
- **SR4:** Input Validation
  - Email address format validation
  - OTP format and length validation
  - SQL injection prevention through parameterized queries
- **SR5:** Rate Limiting Controls
  - Maximum 5 OTP requests per user per 15-minute window



## National University of Computer and Emerging Sciences Islamabad Campus

---

- Maximum 3 failed OTP validation attempts before account lockout
- Temporary account lockout duration: 15 minutes
- **SR6: Security Monitoring**
  - Comprehensive audit logging for authentication events
  - Failed authentication attempt tracking
  - Suspicious activity detection and alerting

### 1. Misuse Case and Abuse Case Analysis

#### 1.1 Identified Misuse Cases

##### MU1: OTP Brute Force Attack

- **Description:** Malicious actor attempts systematic guessing of OTP codes
- **Impact:** Unauthorized system access, account compromise
- **Likelihood:** Medium (rate limiting makes this challenging but possible)

##### MU2: OTP Interception Attack

- **Description:** Attacker intercepts OTP during email transmission
- **Impact:** Account takeover, unauthorized access to protected resources
- **Likelihood:** Low (requires email account compromise or network interception)

##### MU3: Session Hijacking

- **Description:** Attacker captures and reuses valid session tokens
- **Impact:** Impersonation of legitimate users, unauthorized data access
- **Likelihood:** Low (requires network access or XSS vulnerability)

##### MU4: OTP Replay Attack

- **Description:** Attacker reuses previously captured valid OTP codes
- **Impact:** Unauthorized authentication, system access violation
- **Likelihood:** Very Low (OTPs are single-use and time-limited)



## National University of Computer and Emerging Sciences Islamabad Campus

---

### 1.2 Identified Abuse Cases

#### AB1: Account Enumeration

- **Description:** Attacker discovers valid user accounts through system responses
- **Impact:** Information disclosure, targeted attack preparation
- **Likelihood:** Medium (common reconnaissance technique)

#### AB2: Denial of Service via OTP Flooding

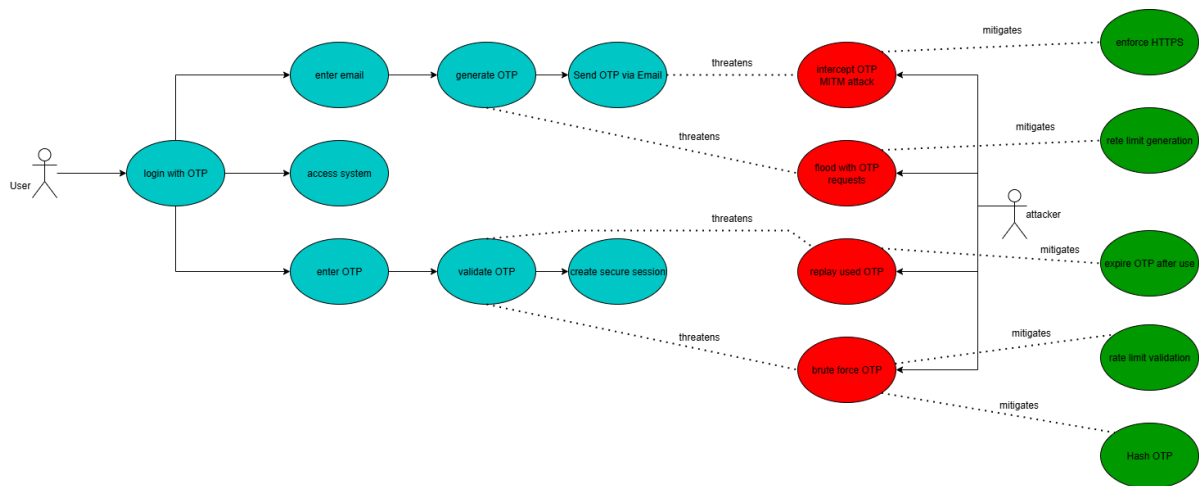
- **Description:** Attacker overwhelms email service with excessive OTP requests
- **Impact:** System unavailability, legitimate user access prevention
- **Likelihood:** Medium (rate limiting provides partial protection)

#### AB3: Social Engineering OTP Extraction

- **Description:** Attacker manipulates users to disclose received OTP codes
- **Impact:** Account compromise, unauthorized system access
- **Likelihood:** High (human factor vulnerability)

### 1.3 Security Countermeasures

- **Rate Limiting:** Prevents brute force and flooding attacks
- **TLS Encryption:** Protects OTP transmission from interception
- **Single-Use Policy:** Eliminates replay attack possibilities
- **Session Security:** Secure cookies and token management prevent hijacking
- **Input Validation:** Prevents injection and malformed request attacks
- **Account Lockout:** Temporary protection against persistent attacks



## 2. Secure Sequence Diagram

### 2.1 OTP Authentication Flow

The secure sequence diagram illustrates the complete authentication process with integrated security controls:

#### Phase 1: OTP Request

1. User submits email/username to Frontend
2. Frontend sends POST request to Backend (/request-otp)
3. Backend validates email format and checks rate limiting
4. Backend generates cryptographically secure OTP
5. Backend stores hashed OTP in database with timestamp
6. Backend sends OTP via encrypted email (SMTP/TLS)
7. Backend returns success response (without revealing OTP)
8. Frontend displays confirmation message to user

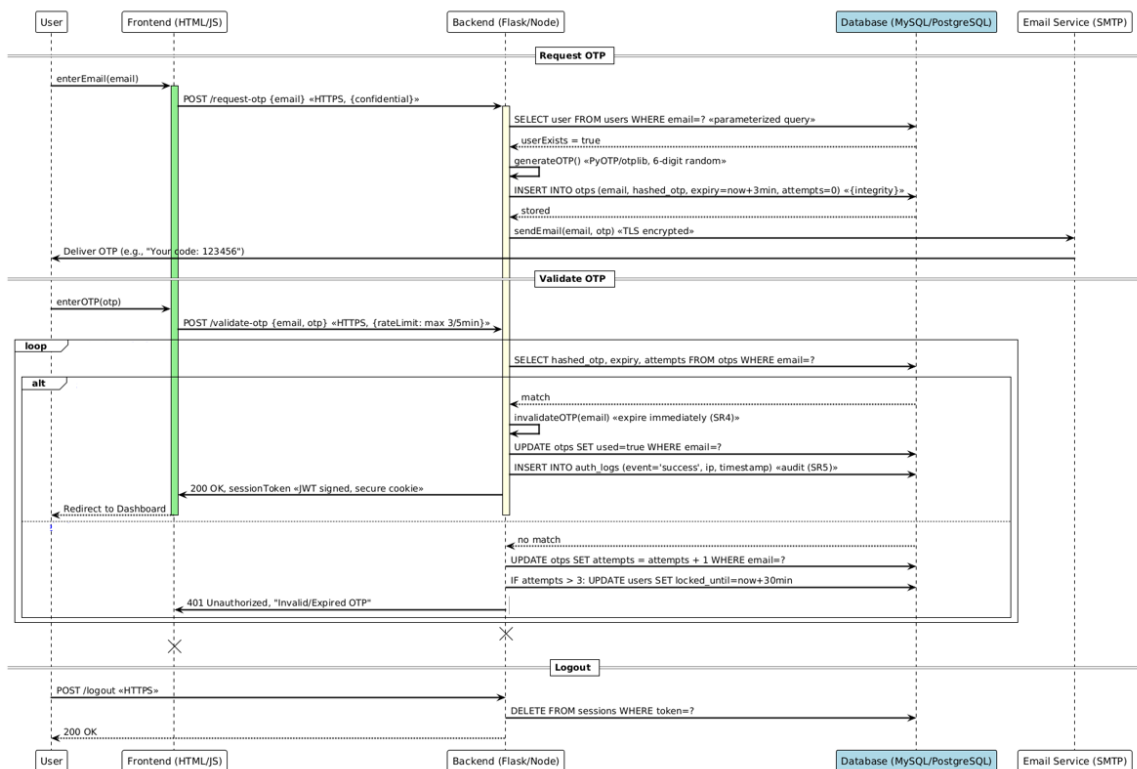
**Phase 2: OTP Validation**

9. User enters received OTP in Frontend
10. Frontend submits OTP to Backend (/verify-otp)
11. Backend retrieves stored OTP hash from database
12. Backend validates OTP code and expiration timestamp
13. Backend marks OTP as used (single-use enforcement)
14. Backend generates secure session token
15. Backend returns authentication success with session token
16. Frontend stores session token in secure cookie
17. Frontend redirects user to protected dashboard



## 2.2 Security Annotations

- <<TLS>>: All communications encrypted with Transport Layer Security
- <<Validate>>: Input validation applied at each interaction point
- <<RateLimit>>: Rate limiting enforced for OTP requests and validations
- <<SecureRandom>>: Cryptographically secure random number generation
- <<SingleUse>>: OTP invalidation after successful authentication
- <<SessionSecure>>: Secure session token generation and storage





### **3. Secure Class Diagram**

#### **3.3 Core Security Classes**

##### **User Entity Class**

- Attributes: id (UUID), email (encrypted), username, created\_at, is\_locked, failed\_attempts
- Security Methods: isLocked(), incrementFailedAttempts(), resetFailedAttempts()
- Security Features: Account lockout management, failed attempt tracking

##### **OTPToken Entity Class**

- Attributes: id (UUID), user\_id, code (hashed), created\_at, expires\_at, is\_used, attempt\_count
- Security Methods: generate(), validate(), isExpired(), markAsUsed()
- Security Features: Cryptographic hashing, expiration control, single-use enforcement

##### **AuthenticationController Class**

- Dependencies: OTPService, EmailService, RateLimiter
- Security Methods: requestOTP(), verifyOTP(), logout()
- Security Features: Rate limit integration, input validation, secure response handling

##### **OTPService Class**

- Dependencies: CryptoProvider, OTPRepository
- Security Methods: generateOTP(), validateOTP(), cleanExpiredOTPs()
- Security Features: Cryptographic operations, secure storage management

##### **SessionService Class**

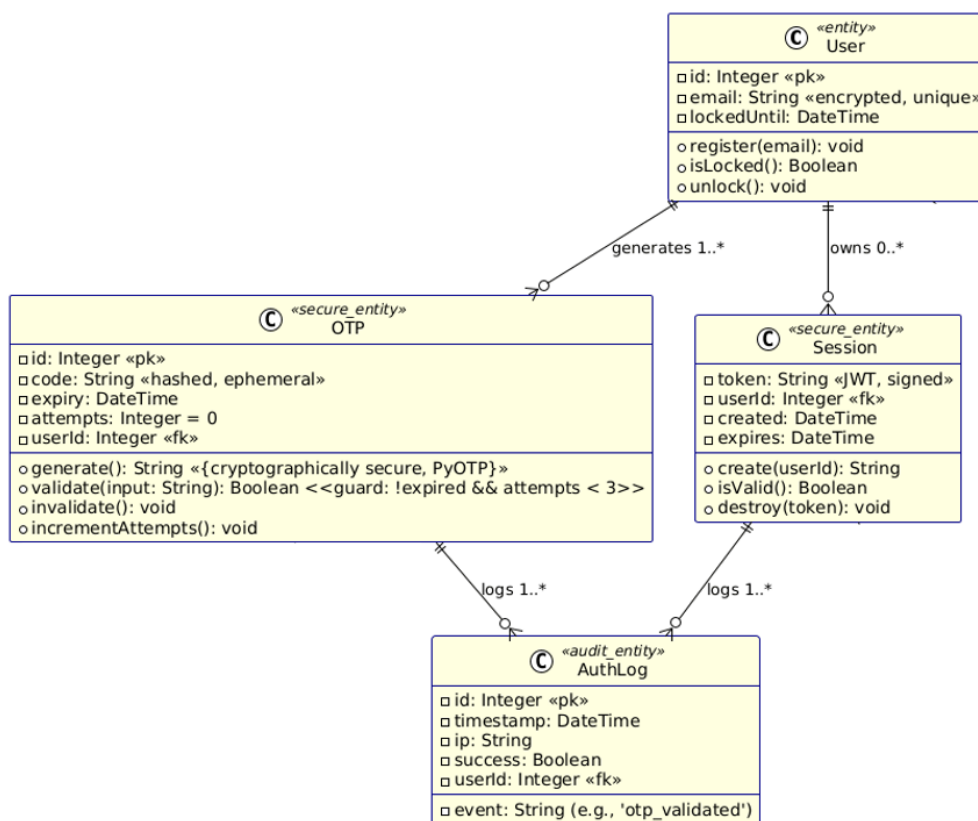
- Dependencies: TokenGenerator, SessionStore
- Security Methods: createSession(), validateSession(), revokeSession()
- Security Features: Secure token generation, session lifecycle management

### RateLimiter Security Class

- Attributes: limits (Map), storage (RateLimitStore)
- Security Methods: checkLimit(), recordAttempt(), resetLimits()
- Security Features: Attack prevention, resource protection

### 3.4 Security Stereotypes

- <<encrypted>>: Data encryption at rest
- <<hashed>>: Cryptographic hash storage
- <<validated>>: Input validation enforcement
- <<ratelimited>>: Rate limiting application
- <<secure>>: Cryptographically secure implementation





## **4. Secure State Chart Diagram**

### **4.1 Authentication State Machine**

#### **Initial State: [Unauthenticated]**

- User has no active session or authentication
- All protected resources are inaccessible
- Transition: request\_otp / validate\_email, check\_rate\_limit

#### **State: [OTP Requested] <<security: rate\_limited>>**

- Email validation completed successfully
- Rate limiting checks passed
- Transition: otp\_generated / send\_otp, start\_timer

#### **State: [OTP Sent] <<security: time\_bounded>>**

- OTP delivered via encrypted email channel
- 3-minute countdown timer initiated
- Multiple possible transitions based on user actions

#### **State: [OTP Failed] <<security: attempt\_limited>>**

- Invalid OTP submission recorded
- Failed attempt counter incremented
- Conditional transitions based on attempt count

#### **State: [Account Locked] <<security: locked>>**

- Maximum failed attempts exceeded (3 attempts)
- Account temporarily inaccessible
- Automatic unlock after 15-minute timeout



## National University of Computer and Emerging Sciences Islamabad Campus

---

**State: [Authenticated]** <<security: session\_protected>>

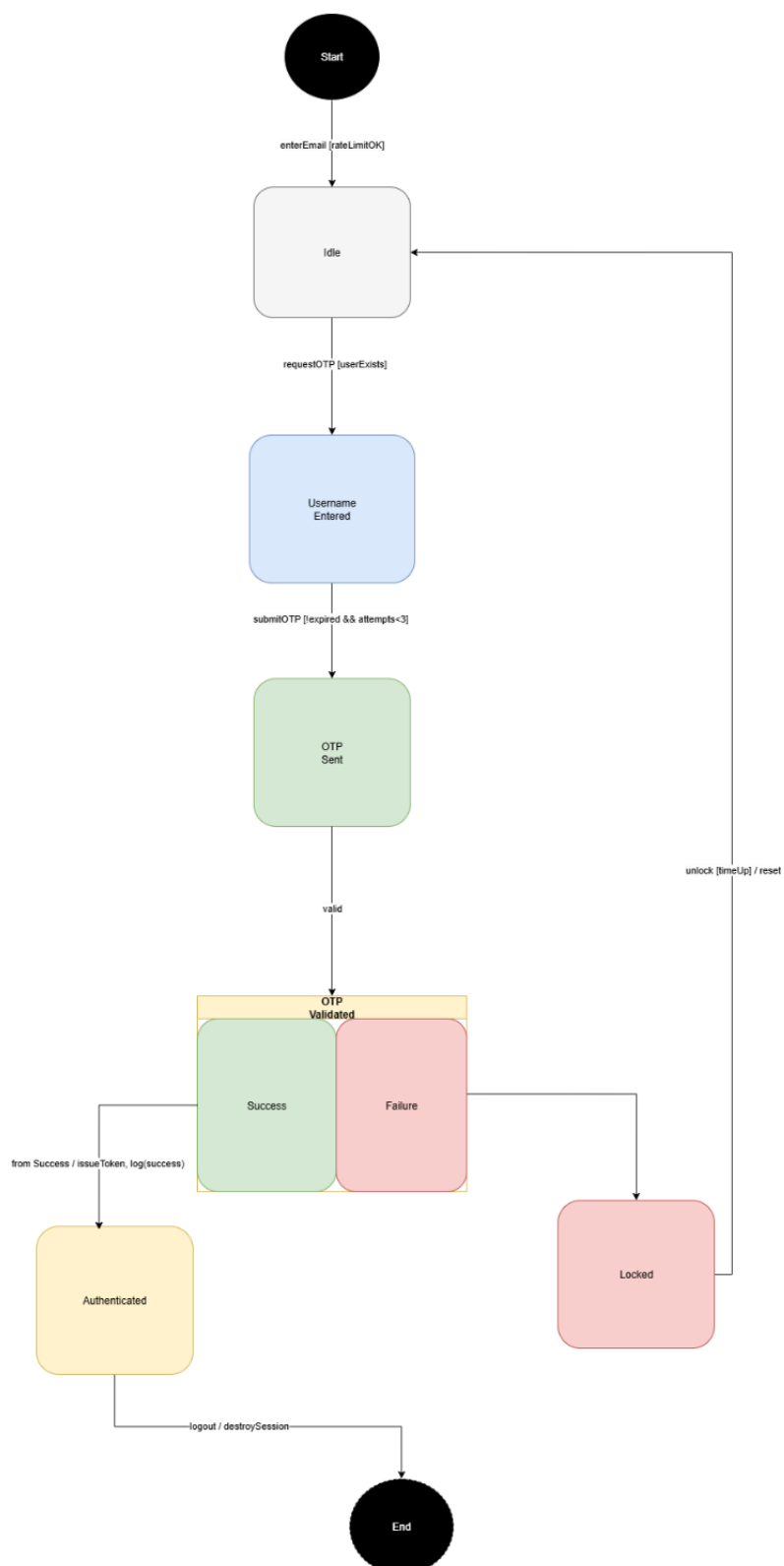
- Valid session established
- Protected resources accessible
- Session timeout and logout transitions available

**State: [OTP Expired]**

- 3-minute time limit exceeded
- OTP invalidated and removed from storage
- Return to unauthenticated state required

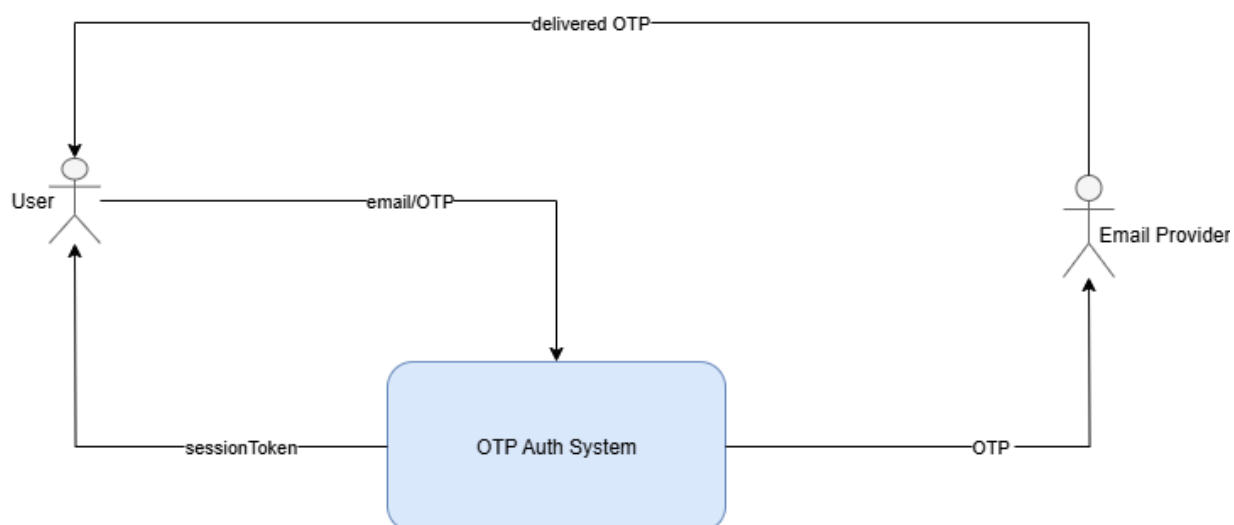
### 4.2 Security State Transitions

- **Timeout Protection:** Automatic OTP expiration after 3 minutes
- **Attempt Limiting:** Account lockout after 3 failed validation attempts
- **Session Management:** Secure session creation and termination
- **Rate Limiting:** Request throttling at OTP generation stage

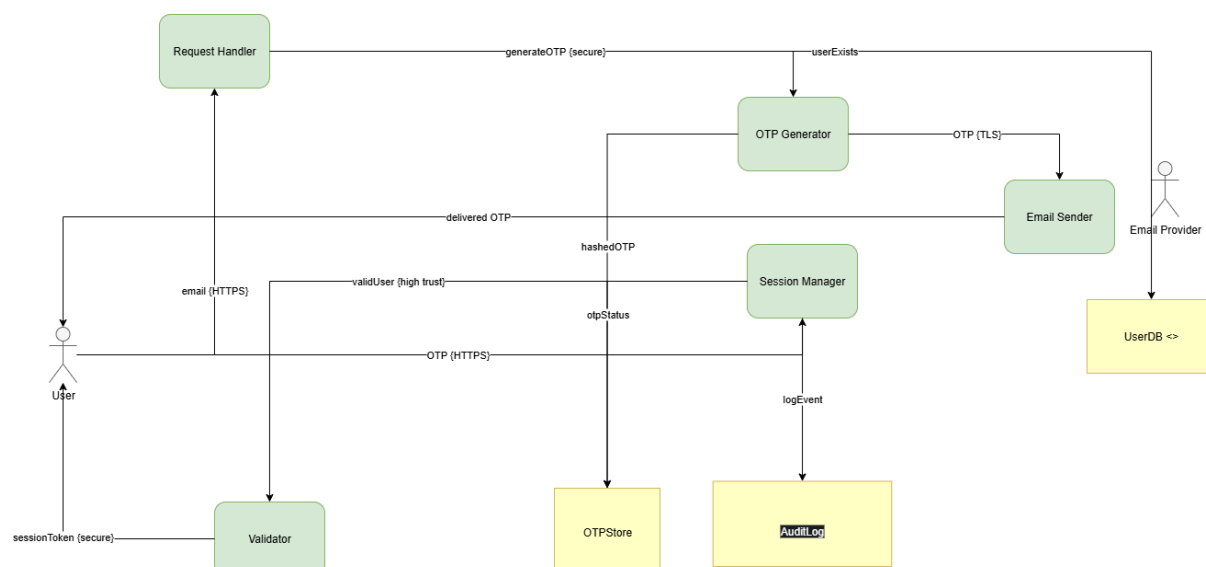


## 5. Secure Data Flow Diagram

### 5.1 System Context (Level 0)

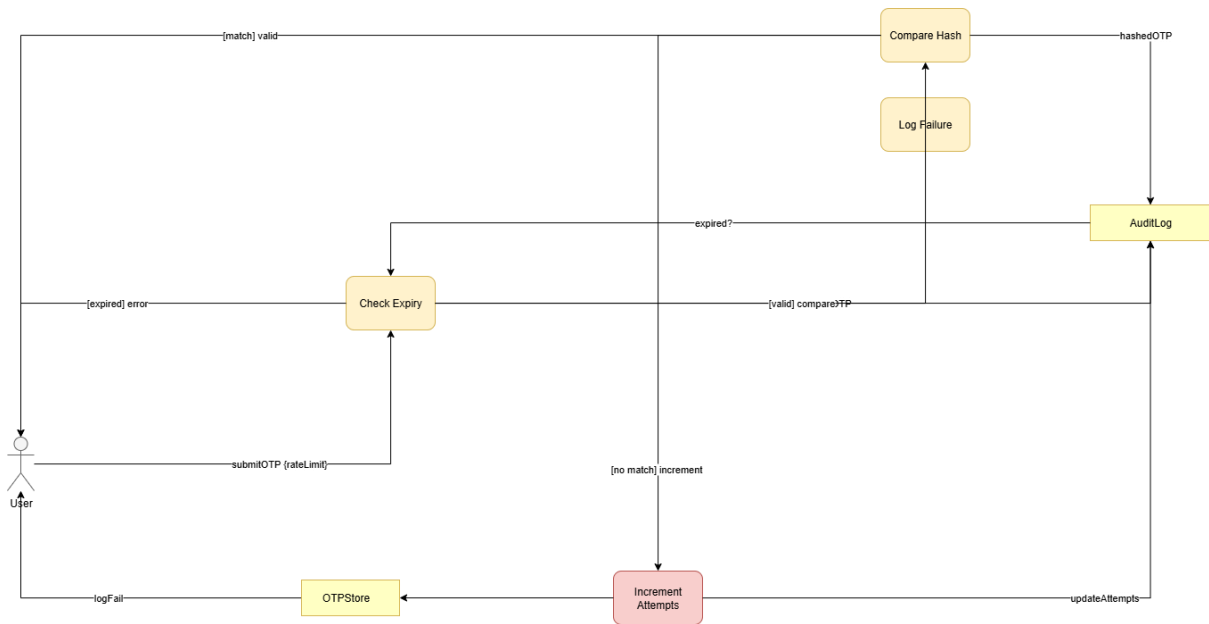


### 5.2 System Decomposition (Level 1)





### 5.3 Trust Boundaries and Security Controls (Level 2)



## Conclusion

This UMLSec design analysis provides comprehensive security modeling for the password-less authentication system using OTPs. Each diagram addresses specific security requirements while maintaining alignment with OWASP security principles and the project's stated objectives. The integrated approach ensures security controls are embedded throughout the system architecture, from user interaction through data storage and external service integration.

The design emphasizes defense-in-depth through multiple security layers including encryption, authentication, authorization, input validation, rate limiting, and comprehensive monitoring. This foundation supports secure implementation and provides clear security requirements for the development and testing phases.