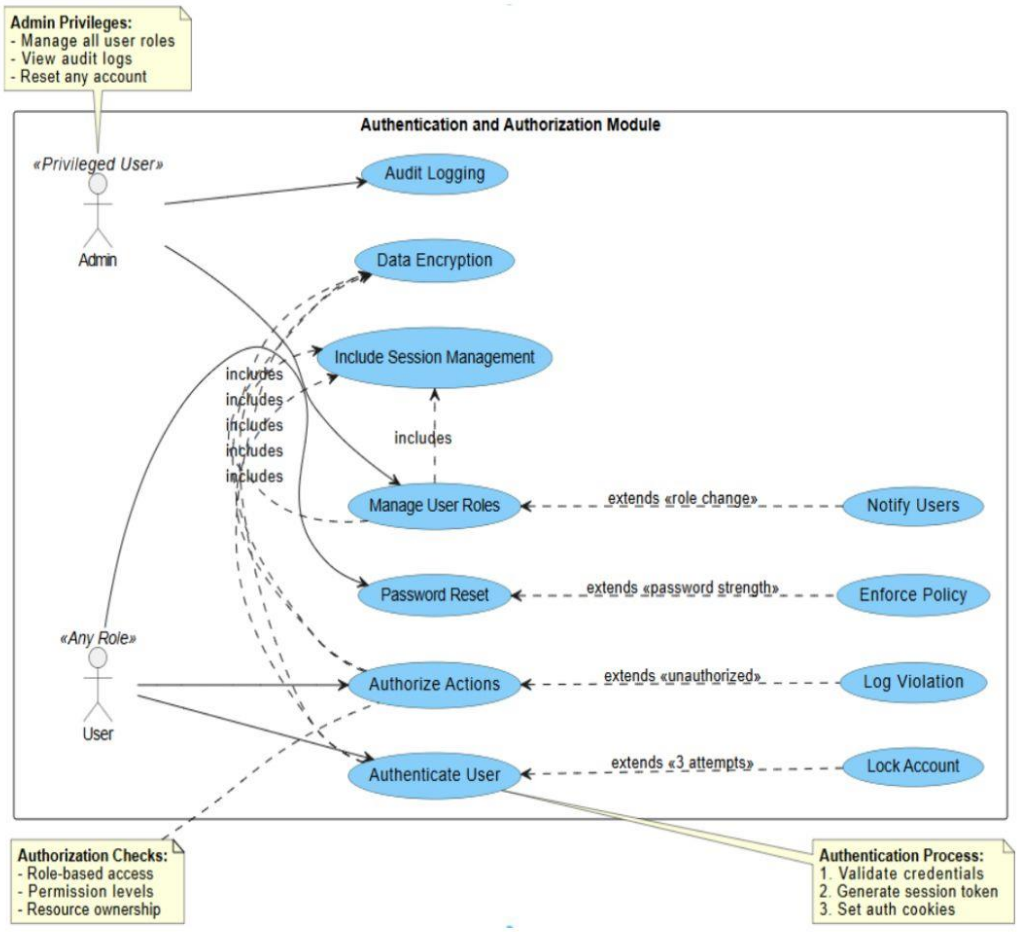
NAME:WIHAJ TAHIR **REG NO:FA22-BSE-158**



Use Case: Manage Admin Privileges

Actor: System Admin

Preconditions:

- Admin is logged in
- Admin has necessary privileges

Basic Flow:

1. Admin selects action (manage roles/view logs/reset account)
2. System verifies permissions

3. Admin performs action
4. System:
    - Executes request
    - Logs activity
    - Notifies affected users (if needed)
5. Confirms completion

Alternate Flows:

- *Unauthorized action* → Log violation, notify security
- *Invalid request* → Show error, cancel action

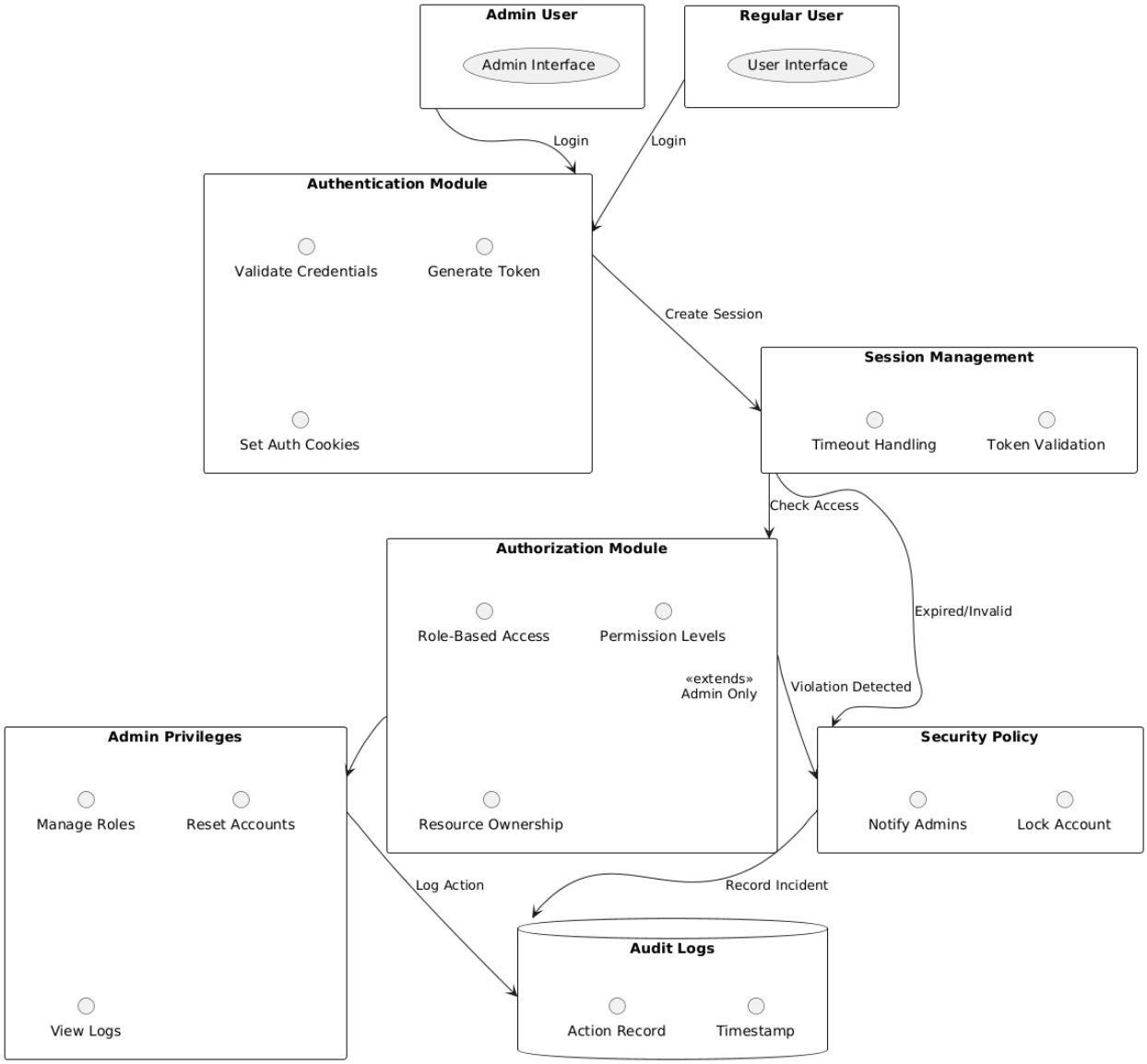
Postconditions:

- Action completed
- Audit log updated

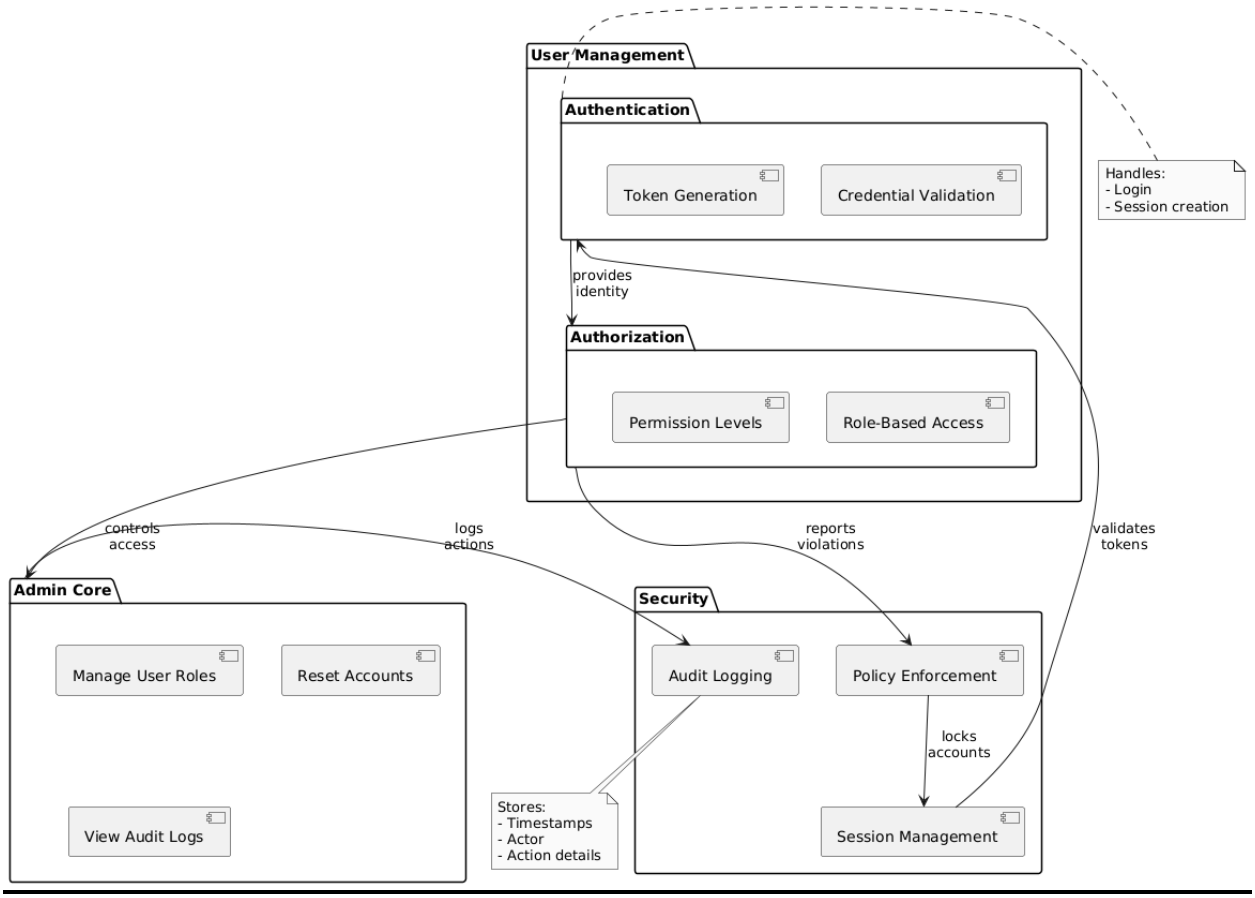
Special Notes:

- All actions are logged
- Temporary passwords expire after 24hrs

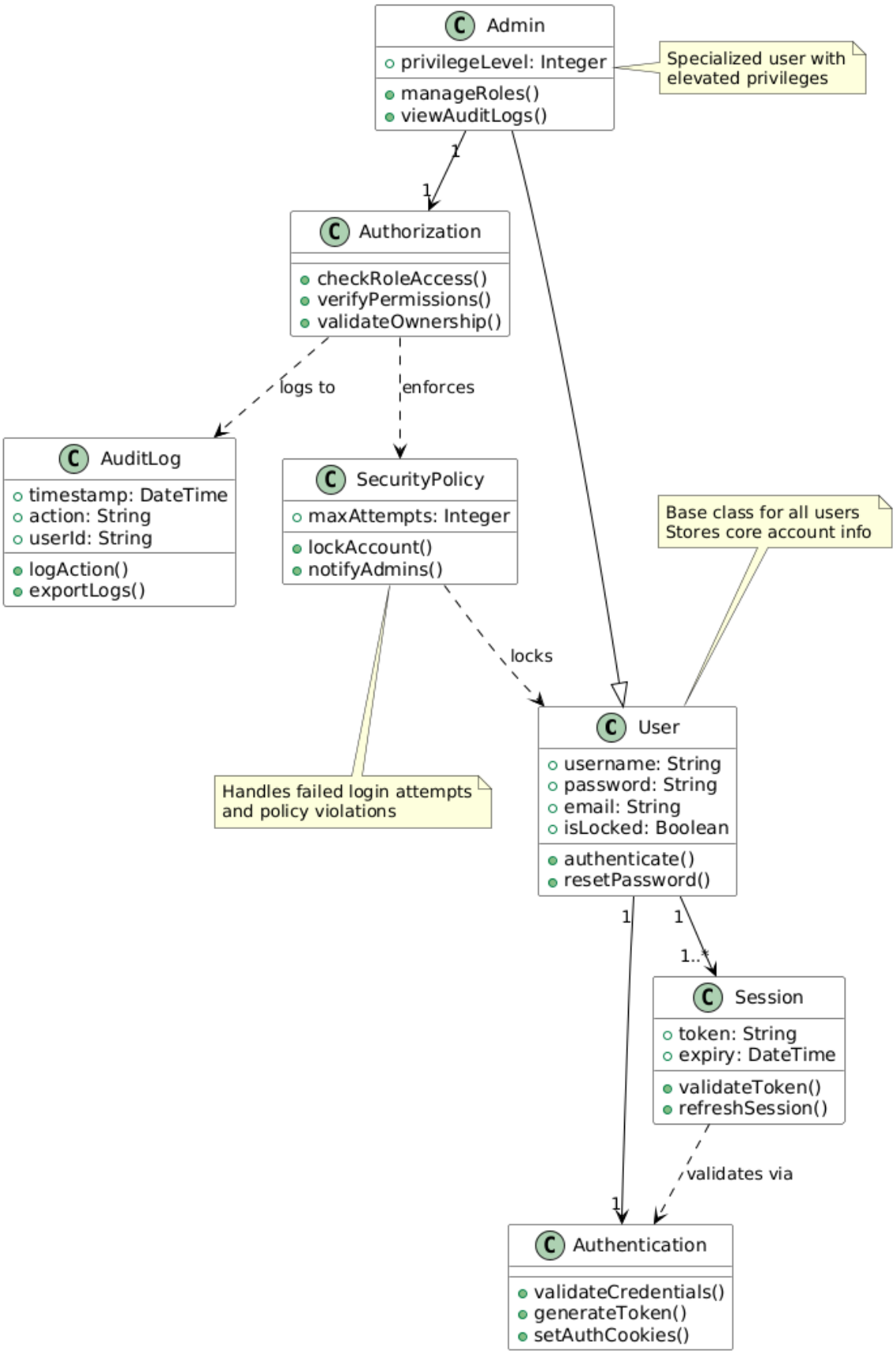
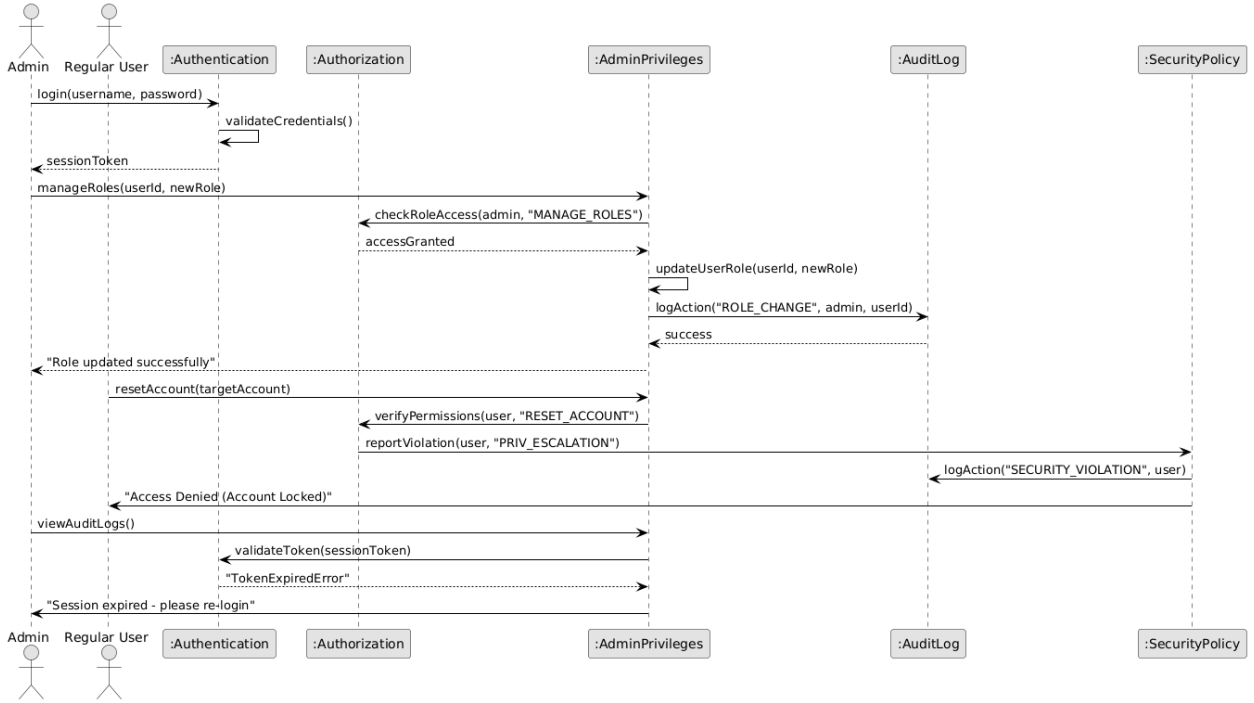
System use case:



PACKAGE DIAGRAM:



Class diagram:


Admin
privilegeLevel: Integer
manageRoles()
viewAuditLogs()
Specialized user with elevated privileges
1
1
Authorization
checkRoleAccess()
verifyPermissions()
validateOwnership()
logs to
enforces
AuditLog
timestamp: DateTime
action: String
userId: String
logAction()
exportLogs()
SecurityPolicy
maxAttempts: Integer
lockAccount()
notifyAdmins()
Base class for all users
Stores core account info
locks
Handles failed login attempts
and policy violations
User
username: String
password: String
email: String
isLocked: Boolean
authenticate()
resetPassword()
1
1
1..*
Session
token: String
expiry: DateTime
validateToken()
refreshSession()
validates via
1
Authentication
validateCredentials()
generateToken()
setAuthCookies()

SYSTEM SEQUENCE DIAGRAM: