**Software Design/Engineering Pseudocode:**

// Define roles and their associated permissions

function define\_roles():

roles = {

'Doctor': {'view\_patient', 'add\_notes', 'view\_notes', 'schedule\_appointment'},

'Nurse': {'view\_patient', 'add\_notes'},

'Admin': {'add\_user', 'modify\_user', 'remove\_user'}

}

return roles

// Assign roles to users

function assign\_role(user, role):

user.role = role

log\_action("Role " + role + " assigned to user " + user.username)

// Check if a user has a specific permission

function has\_permission(user, permission):

role = user.role

roles = define\_roles()

if permission in roles[role]:

return True

else:

return False

// Function to view patient details

function view\_patient\_details(user, patient):

if has\_permission(user, 'view\_patient'):

display\_patient\_details(patient)

else:

display\_error("Access denied: insufficient permissions")

// Function to add patient notes

function add\_patient\_notes(user, patient, notes):

if has\_permission(user, 'add\_notes'):

save\_patient\_notes(patient, notes)

log\_action("Notes added to patient " + patient.id + " by user " + user.username)

else:

display\_error("Access denied: insufficient permissions")

// User tries to view patient details

function user\_action\_view\_details(user, patient):

view\_patient\_details(user, patient)

// User tries to add notes to a patient's record

function user\_action\_add\_notes(user, patient, notes):

add\_patient\_notes(user, patient, notes)

**MongoDB Database Pseudocode:**

//Initialize Database

function initializeDatabaseConnection():

connect to MongoDB

if connection is successful:

print "Database connection successful"

else:

print "Database connection failed"

//Create Patient Record

function createPatientRecord(patientData):

try:

insert patientData into the 'patients' collection

print "Patient record created successfully"

except DatabaseError:

print "Error creating patient record"

// Read Patient Record

function readPatientRecord(patientId):

try:

patient = find document in 'patients' collection where \_id = patientId

if patient is not None:

return patient

else:

print "No record found for patient"

except DatabaseError:

print "Error reading patient record"

//Update Patient Record

function updatePatientRecord(patientId, updatedData):

try:

result = update document in 'patients' collection where \_id = patientId with updatedData

if result is successful:

print "Patient record updated successfully"

else:

print "Update failed"

except DatabaseError:

print "Error updating patient record"

//Delete Patient Record

function deletePatientRecord(patientId):

try:

result = delete document from 'patients' collection where \_id = patientId

if result is successful:

print "Patient record deleted successfully"

else:

print "No record found to delete"

except DatabaseError:

print "Error deleting patient record"