

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"
```

