

## HOSPITAL MANAGEMENT SYSTEM (HMS)

- Application would be used by:
  - Physician(s)
  - Nurse(s)
  - Patient(s)
- What each of them can do?
  - Patient
    - 1. Can check his/her details & download the same in JSON/CSV format.
    - 2. Can check and/or download his/her visit details.
  - Nurse
    - 1. Can schedule an appointment for the patient with the physician.
  - Physician
    - 1. Can check his/her upcoming patient visit(s).
    - 2. Can view his/her patient's visit history.
    - 3. Can capture patient health details during a patient visit.
- Modules (*DLLs in case of a Console Application*)
  - InboxModule (*InBoxModule.dll*)
    - 1. All tasks for a Physician or Nurse.
    - 2. Will be for a Nurse & Physician.
    - 3. For a Nurse:
      - 1. List of appointments to be scheduled with Physician(s).
    - 4. For a Physician:
      - 1. List of appointments scheduled with Patient(s).
  - PatientVisitModule (*PatientVisitModule.dll*)
    - 1. Will be for a Physician.
      - 1. All patient vitals like height, weight, blood pressure, last visit date, medications prescribed, diagnosis and additional notes.
  - AppointmentModule (*AppointmentModule.dll*)
    - 1. Allow a patient to request for an appointment.
    - 2. Allow Nurse(s) to schedule an appointment with Physician(s).
  - PatientModule (*PatientModule.dll*)
    - 1. Patient can check & download his/her details in JSON format.
    - 2. New Patient registration.
  - AdminModule (*AdminModule.dll*)
    - 1. New Nurse & Physician registration in the hospital.
    - 2. View, Edit, Remove Nurse and/or Physician details.
  - LoginModule (*LoginModule.dll*)
    - 1. Patient, Physician, Nurse or Admin login.
    - 2. Forgot password & change password.
    - 3. Register new Patient, Physician or Nurse.

- LoggerModule (*LoggerModule.dll*)
  1. Contains **ILogger** interface with one method:  
**void Log(string message);**
  2. There could be different types of loggers, each which would write the log to a different destination:
    - FileLogger** → writes logs to a text file
    - DBLogger** → writes logs to a SQL Server database
    - EmailLogger** → writes logs via an email
  3. A static method named **WriteLog(ILogger logger)** calls the **Log()** method of the appropriate logger to write logs.
  4. FOR THIS PROJECT, USE **FileLogger** which writes logs to a text file.

- ValidationModule (*ValidationModule.dll*)
  - 1. Validate password
    - 1. Ensures that password:
      - Must contain at least 5 chars and not more than 15.
      - Must contain at least 1 uppercase char & at least one numeric char
      - Must not contain any special chars (*including spaces*)
  - 2. Validate email
    - 1. The email must look like an email.
- EmailModule (*EmailModule.dll*)
  - 1. SendEmail
    - 1. The email must be sent to the patient's email address.
    - 2. If the email is for acknowledging appointment schedule, the email body must contain the appointment details.
    - 3. If the email is for patient visit, the email body must contain the e-prescription containing the visit details.
- Enumerations
  - HMSRole
    - 1. Physician
    - 2. Nurse
    - 3. Patient
    - 4. Admin
  - HMSBloodPressureType
    - 1. Systolic
    - 2. Diastolic
  - DiseaseCategory
    - 1. Cardio
    - 2. Diabetes
    - 3. Neuro
    - 4. Orthopedic
- Entities
  - Physician
    - 1. Id
    - 2. Name
    - 3. Password
    - 4. DOB
    - 5. Email
    - 6. RoleId
    - 7. Role (*must be of type HMSRole*)
    - 8. DiseaseCategoryId
    - 9. DiseaseCategory (*of type HMSDiseaseCategory*)

- Nurse
  1. Id
  2. Name
  3. Password
  4. DOB
  5. Email
  6. RoleId
  7. Role (*of type HMSRole*)
  8. DiseaseCategoryId
  9. DiseaseCategory (*of type HMSDiseaseCategory*)
  
- Patient
  1. Id
  2. Name
  3. Password
  4. Email
  5. DOB
  6. RoleId
  7. Role (*of type HMSRole*)
  8. DiseaseCategoryId
  9. DiseaseCategory (*of type PhysicianSpeciality*)
  
- PatientVisit
  1. VisitId
  2. AppointmentId
  3. PhysicianId (*this has to be the Id of the Physician*)
  4. PatientId (*this has to be the Id of the Patient*)
  5. BookedBy (*this has to be the Id of the Nurse*)
  6. VisitDate
  7. Height
  8. Weight
  9. BloodPressureId
  10. BloodPressureType (*of type HMSBloodPressureType*)
  11. DiagnosisId
  12. MedicationId
  13. AllergyId
  14. NextVisitDate

- Appointment
  1. AppointmentId
  2. NurseId (*this has to be the Id of the Nurse*)
  3. PatientId (*this has to be the Id of the Patient*)
  4. PhysicianId (*this has to be the Id of the Physician*)
  5. AppointmentDate
  6. IsScheduled
  7. IsVisitCompleted
  
- **Application workflow outline**
  - Patient raises an appointment request to a nurse.
  - The nurse schedules the appointment for the patient with the physician & marks the appointment as **scheduled**.
  - An email is sent to the patient. Email contains the appointment details.
  - Patient visits the physician.
  - During the visit, the physician captures patient's vital details, prescribes medication & marks the visit as **completed**.
  - An email is sent to the patient with an **E-Prescription** containing the visit details.
  
- **Application detailed workflow (for a console app)**
  - The app starts with the following menu and inputs the choice:
    1. Login
    2. Forgot Password
    3. Change Password
    4. Quit

*Enter choice (1-4): \_*
  
- What happens when each option is selected?
  - Option 1 → Login
    1. Prompt for id & password.
    2. If invalid, throw an exception **InvalidLoginException** & stay in the menu.
    3. If invalid login for the same id is attempted 3 times consecutively, display a message **Please try after 1 minute again**. Within a minute if the same id tries to login again, the same error must be displayed. Login for that id must be allowed only after 1 minute.
    4. Log invalid login details:
      1. Who tried to login?
      2. What was the password?
      3. Login date and time
    5. If login is valid, display a message **Welcome <<login name>>**.

6. Log valid login details:
  1. Who tried to login?
  2. What was the password?
  3. Login date and time
7. Display the following menu if the logged in user is a **Patient**:
  1. View my personal details.
    - Display the details of the logged in patient.
  2. Download my personal details.
    - Save the logged in patient's details in a .csv file with the patient's name.
  3. View my visit history.
    - Display the logged in patient's visit history.
    - In case there is no visit history, display a message **NO VISIT HISTORY FOUND**.
  4. Raise an appointment request.
    - Display all nurse ids and names whose **DiseaseCategoryId** matches the **DiseaseCategoryId** of the nurses in the HSM.
    - Prompt the patient to enter a nurse id to whom the appointment request must be sent.
    - Prompt the patient to input an appointment date.
    - Display a message **APPOINTMENT REQUEST RAISED. YOUR APPOINTMENT ID IS <<display appointment id for the request>>**.
  5. Logout.
    - Return to the previous main menu.
8. Display the following menu if the logged in user is a **Nurse**:
  1. View my Inbox.
    - Display the appointment request list for the logged in nurse.
  2. Process appointment request(s) and schedule the appointment.
    - Prompt the nurse to enter an appointment id.
    - The list of physicians with whom appointment can be scheduled is displayed.
    - Prompt to enter the physician id with whom appointment is to be scheduled.
    - Nurse schedules the patient's appointment with a physician and marks the appointment as **scheduled**.
    - Send an appointment scheduled confirmation email to the patient mentioning the appointment details.
    - Also write the appointment confirmation to the log.
  3. Logout.
    - Return to the previous main menu.
9. Display the following menu if the logged in user is a **Physician**:

1. View my inbox.
    - Display all appointments which have been marked as **scheduled** with the logged in physician.
  2. Process patient appointment.
    - Prompt physician to enter appointment id which he wants to process.
    - Create a patient visit entry, capture that patient's vital information, prescribe medication, update allergy info and mark the appointment as **completed**.
    - Physician updates the next visit date accordingly.
    - Send an email to the patient mentioning the e-prescription created by the physician during the visit.
    - Also write the visit completed details to the log.
  3. Logout
    - Return to the previous main menu.
10. Display the following menu if the logged in user is an **Admin**:
1. New Nurse registration.
  2. New Physician registration.
  3. Remove Nurse/Physician
    - Prompt for id.
    - Ask confirmation whether to remove or not.
    - Remove the details from backend.
  4. Logout
    - Return to the previous main menu.
- Option 2 → Forgot Password
    1. Prompt user to enter id.
    2. Validate id.
    3. Retrieve the password from the backend and email it to the user.
  - Option 3 → Change Password
    1. Prompt user to enter id.
    2. Validate id.
    3. Prompt user to enter old password.
    4. Validate old password.
    5. Prompt user to enter new password.
    6. Prompt user to reenter new password.
    7. Validate new password.
    8. Update the user's password with new password in the backend.

**Appointment Scheduled Confirmation Content in the email body:**

Subject: HMS - Appointment Scheduled

Body

Hello <<patientname>>

Your appointment has ben scheduled on <appointment date> with Dr. <<physician name>>.

Your appointment id is <<appointment id>>.

Thanks & Regards,

<<nurse name>>

**E-Prescription Content in the email body:**

Subject: Visit Details - EPrescription

Patient Id: <<patient id>>

Full Name: <<patient name>>

Date of Birth: <<patient's DOB>>

Visit Date: <<visit date>>

Height: <<height of the patient>>

Weight during visit: <<weight of the patient>>

Diagnosis: <<put the description of the diagnosis>>

Medication suggested: <<put the description of medication>>

Allergies: <<put details of allergies of the patient>>

Next Visit Suggested on: <<put the next visit date as suggested by the physician>>



## Database Tables with dummy data

**Table 1: HMSRoles**

RoleId	Role
HMS-ROLE1	Admin
HMS-ROLE2	Physician
HMS-ROLE3	Nurse
HMS-ROLE4	Patient

**Table 2: HMSDiseaseCategories**

DiseaseCategoryId	Disease
HMS-DISCAT1	Cardio
HMS-DISCAT2	Diabeties
HMS-DISCAT3	Neuro
HMS-DISCAT4	Orthopedic

**Table 3: HMSBloodPressureTypes**

BloodPressureId	Type
HMS-BP1	Systolic
HMS-BP2	Diastolic

**Table 4: HMSDiagnosis**

DiagnosisId	DiseaseCategoryId	Description
DIAG-1	HMS-DISCAT1	Cardio Diagnosis 1
DIAG-2	HMS-DISCAT1	Cardio Diagnosis 2
DIAG-3	HMS-DISCAT2	Diabeties Diagnosis 1
DIAG-4	HMS-DISCAT2	Diabeties Diagnosis 2
DIAG-5	HMS-DISCAT3	Neuro Diagnosis 1
DIAG-6	HMS-DISCAT3	Neuro Diagnosis 2
DIAG-7	HMS-DISCAT4	Ortho Diagnosis 1
DIAG-8	HMS-DISCAT4	Ortho Diagnosis 2

**Table 5: HMSMedicationsHM**

MedicationId	DiagnosisId	Medicine	Description
HMS-MED1	DIAG-1	M1	
HMS-MED2	DIAG-1	M2	
HMS-MED3	DIAG-2	M3	
HMS-MED4	DIAG-2	M4	
HMS-MED5	DIAG-3	M5	
HMS-MED6	DIAG-3	M6	
HMS-MED7	DIAG-4	M7	
HMS-MED8	DIAG-4	M8	
HMS-MED9	DIAG-5	M9	
HMS-MED10	DIAG-5	M10	
HMS-MED11	DIAG-6	M11	
HMS-MED12	DIAG-6	M12	
HMS-MED13	DIAG-7	M13	
HMS-MED14	DIAG-7	M14	
HMS-MED15	DIAG-8	M15	
HMS-MED16	DIAG-8	M16	

**Table 6: HMSAllergies**

AllergyId	Allergies
HMS-ALG1	A1, A2, A3
HMS-ALG2	A4, A5, A6
HMS-ALG3	A7, A8, A9
HMS-ALG4	A10, A11, A12

**Table 7: HMSMaster**

Id	Name	Password	DOB	Email	RoleId	DiseaseCategoryId	
HMS-PH1	Physician1				2	1	PHYSICIANS
HMS-PH2	Physician2				2	1	
HMS-PH3	Physician3				2	2	
HMS-PH4	Physician4				2	2	
HMS-PH5	Physician5				2	3	
HMS-PH6	Physician6				2	3	
HMS-PH7	Physician7				2	4	
HMS-PH8	Physician8				2	4	
HMS-NR1	Nurse1				3	1	NURSES
HMS-NR2	Nurse2				3	1	
HMS-NR3	Nurse3				3	2	
HMS-NR4	Nurse4				3	2	
HMS-NR5	Nurse5				3	3	
HMS-NR6	Nurse6				3	3	
HMS-NR7	Nurse7				3	4	
HMS-NR8	Nurse8				3	4	
HMS-ADM1	Admin1				1	NULL	ADMINS
HMS-ADM2	Admin2				1	NULL	
HMS-PAT1	Patient1				4	1	PATIENTS
HMS-PAT2	Patient2				4	2	
HMS-PAT3	Patient3				4	3	
HMS-PAT4	Patient4				4	4	

**Table 8: HMSPatientVisits**

VisitId	AppointmentId	PhysicianId	PatientId	BookedBy	VisitDate	Height	Weight	BloodPressureId	DiagnosisId	MedicationId	AllergyId
---------	---------------	-------------	-----------	----------	-----------	--------	--------	-----------------	-------------	--------------	-----------

**Table 9: HMSAppointments**

AppointmentId	NurseId	PatientId	PhysicianId	AppointmentDate	IsScheduled	IsVisitCompleted
---------------	---------	-----------	-------------	-----------------	-------------	------------------