Hospital Management System

Scenario

Hospital management system helps in registering information about patients and handles patient's query. A unique ID is generated for each patient after registration. This helps in implementing customer relationship management and also maintains medical history of patient. This system also monitors the doctor appointments, when the ID is generated the patient receives the appointment time and number from the receptionist and accordingly visit the doctor. This system also deals with testing appointments as and when ID is generated the patient receives the appointment time and number and accordingly undergoes the test.

It also deals with bed allotments to various patients by checking their ID. It also undergoes various operations by diagnosing the patients. The system identifies whether the person is a doctor or staff and handles various activities such as draw salary and give salary, also it adds doctor/staff information into database. This system is responsible for handling various other activities like deleting, editing doctor/staff information into the database. As per doctor diagnoses the patient, gives treatment and gives suggestions to patients and prescribe laboratory tests and medicines. This system also takes care of medical equipment, doctor visit, vitals recording, patient case sheet, diet ordering, blood requisition, transfer information and discharge information, maintenance of wards, inter and intrawards transfers also it generates patient's discharge summary which includes patients health at the time of discharge, medical history, various diagnosis and drug prescriptions, history of patients illness and course in hospital. Patient can pay bill through credit card, cash or cheque whose information is maintained by this system.

UML Diagrams with plantUML codes

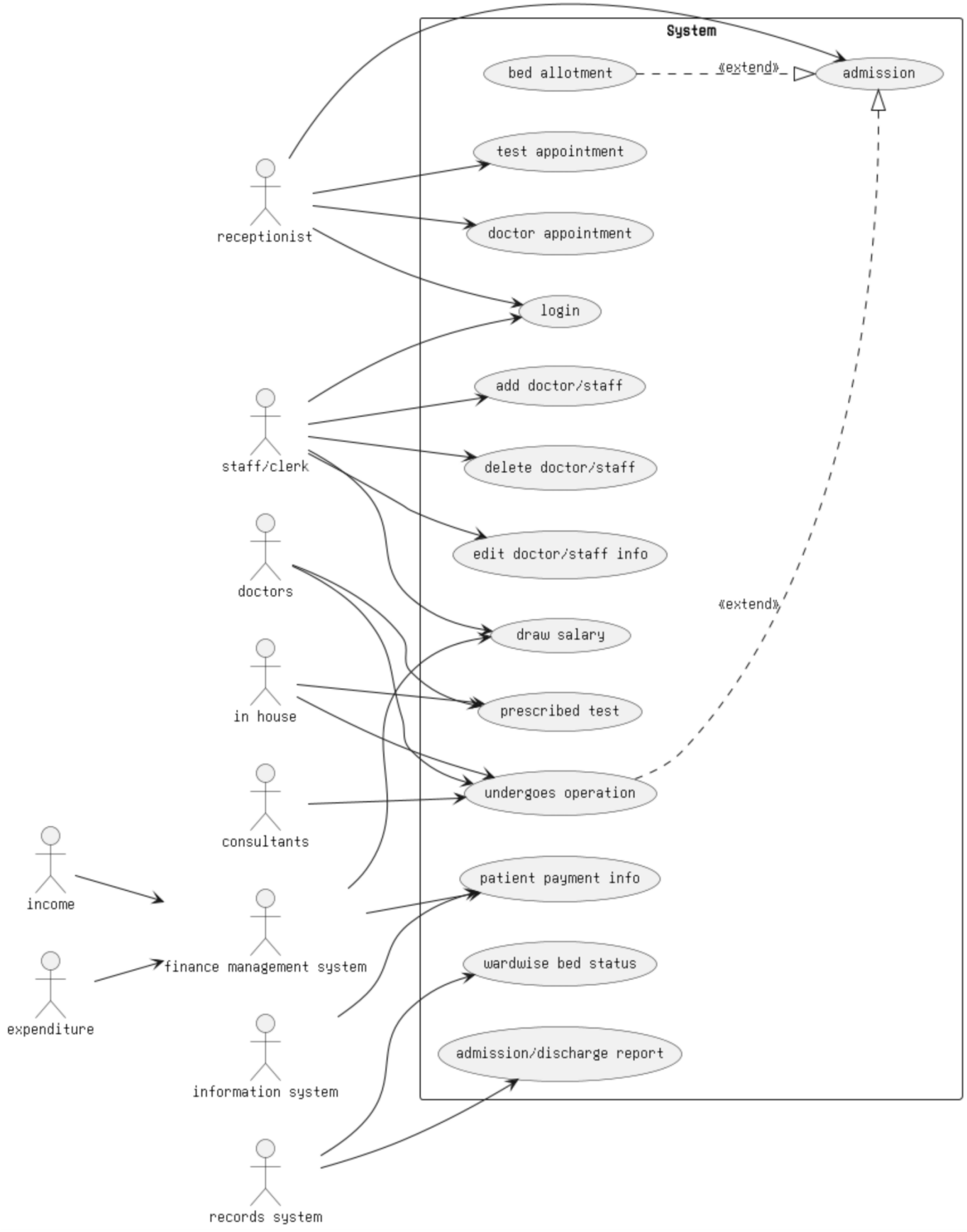
1) Use Case Diagram

PlantUML code:

```
@startuml
left to right direction

actor receptionist
actor "staff/clerk" as staff
actor "in house" as inhouse
actor doctors
actor consultants
actor "finance management system" as finance
actor income
actor expenditure
actor "records system" as records
actor "information system" as info

rectangle System {
   usecase "admission" as UC1
```

```
    usecase "bed allotment" as UC2
    usecase "doctor appointment" as UC3
    usecase "test appointment" as UC4
    usecase "undergoes operation" as UC5
    usecase "login" as UC6
    usecase "draw salary" as UC7
    usecase "add doctor/staff" as UC8
    usecase "delete doctor/staff" as UC9
    usecase "edit doctor/staff info" as UC10
    usecase "prescribed test" as UC11
    usecase "wardwise bed status" as UC12
    usecase "admission/discharge report" as UC13
    usecase "patient payment info" as UC14
}

UC2 ..|> UC1 : <<extend>>
UC5 ..|> UC1 : <<extend>>

receptionist --> UC1
receptionist --> UC3
receptionist --> UC4
receptionist --> UC6

staff --> UC7
staff --> UC6

inhouse --> UC5
inhouse --> UC11

doctors --> UC5
doctors --> UC11

consultants --> UC5

staff --> UC8
staff --> UC9
staff --> UC10

finance --> UC7
finance --> UC14
income --> finance
expenditure --> finance

records --> UC12
records --> UC13

info --> UC14

@enduml
```

Diagram:

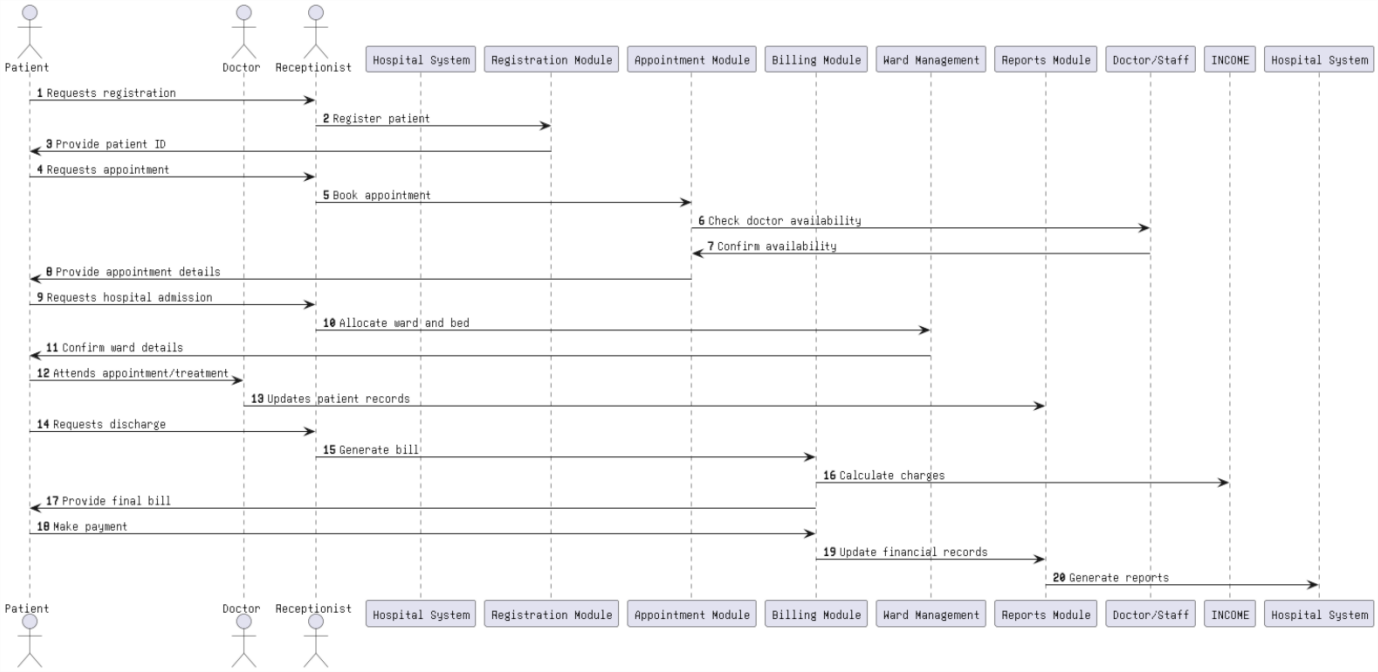

System
bed allotment
«extend»
admission
test appointment
doctor appointment
login
add doctor/staff
delete doctor/staff
edit doctor/staff info
draw salary
prescribed test
undergoes operation
patient payment info
wardwise bed status
admission/discharge report
«extend»
receptionist
staff/clerk
doctors
in house
consultants
income
finance management system
expenditure
information system
records system

2) Sequence Diagram

PlantUML code:

```
@startuml

autonumber

actor Patient
actor Doctor
actor Receptionist

participant "Hospital System" as HS
participant "Registration Module" as RM
participant "Appointment Module" as AM
participant "Billing Module" as BM
participant "Ward Management" as WM
participant "Reports Module" as RPM

Patient -> Receptionist: Requests registration
Receptionist -> RM: Register patient
RM -> Patient: Provide patient ID

Patient -> Receptionist: Requests appointment
Receptionist -> AM: Book appointment
AM -> "Doctor/Staff": Check doctor availability
"Doctor/Staff" -> AM: Confirm availability
AM -> Patient: Provide appointment details

Patient -> Receptionist: Requests hospital admission
Receptionist -> WM: Allocate ward and bed
WM -> Patient: Confirm ward details

Patient -> Doctor: Attends appointment/treatment
Doctor -> RPM: Updates patient records

Patient -> Receptionist: Requests discharge
Receptionist -> BM: Generate bill
BM -> INCOME: Calculate charges
BM -> Patient: Provide final bill
Patient -> BM: Make payment

BM -> RPM: Update financial records
RPM -> "Hospital System": Generate reports

@enduml
```

Diagram:



3) Class Diagram

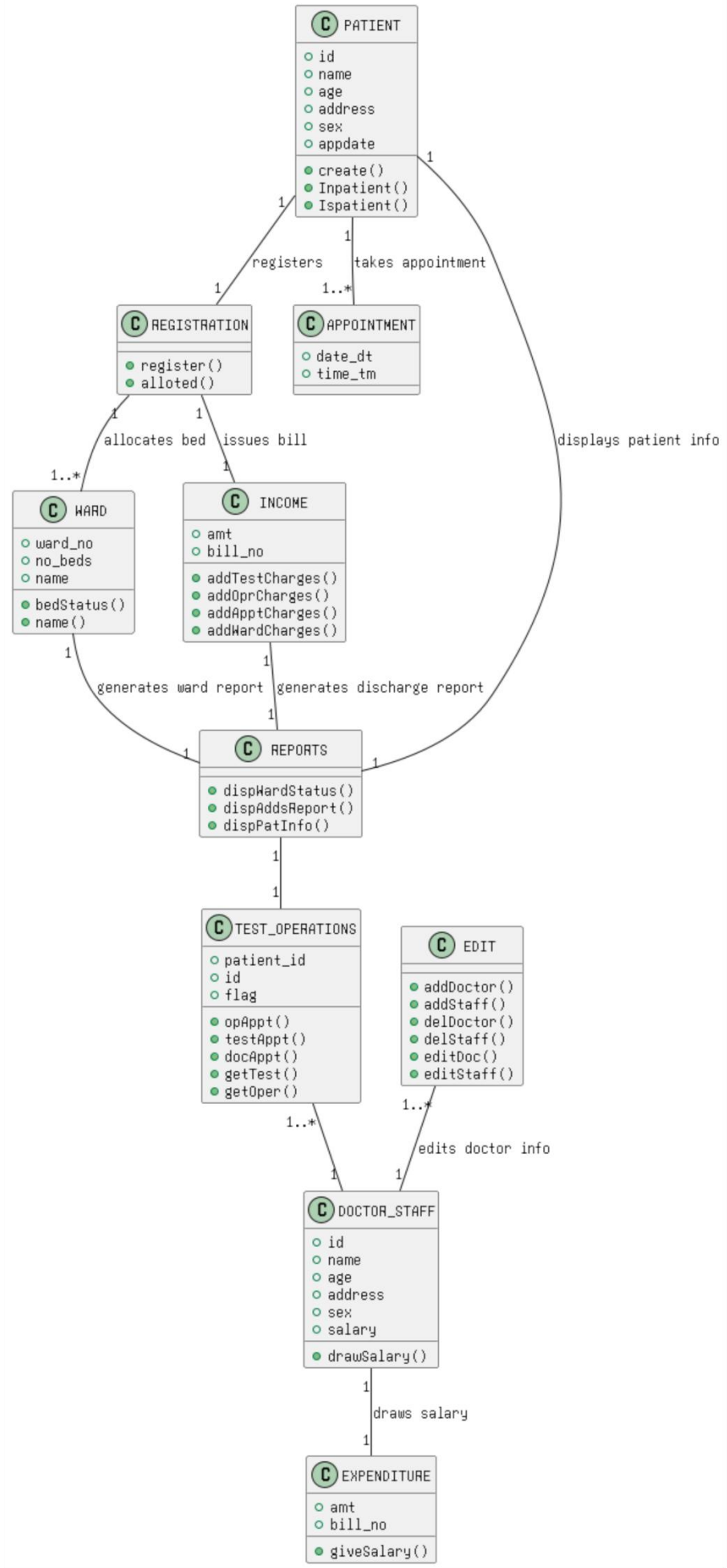
PlantUML code:

```
@startuml

class PATIENT {
  +id
  +name
  +age
  +address
  +sex
  +appdate
  +create()
  +Inpatient()
  +Ispatient()
}

class REGISTRATION {
  +register()
  +alloted()
}

class WARD {
  +ward_no
  +no_beds
  +name
```

```
  +bedStatus()
  +name()
}

class INCOME {
  +amt
  +bill_no
  +addTestCharges()
  +addOprCharges()
  +addApptCharges()
  +addWardCharges()
}

class APPOINTMENT {
  +date_dt
  +time_tm
}

class EDIT {
  +addDoctor()
  +addStaff()
  +delDoctor()
  +delStaff()
  +editDoc()
  +editStaff()
}

class TEST_OPERATIONS {
  +patient_id
  +id
  +flag
  +opAppt()
  +testAppt()
  +docAppt()
  +getTest()
  +getOper()
}

class REPORTS {
  +dispWardStatus()
  +dispAddsReport()
  +dispPatInfo()
}

class DOCTOR_STAFF {
  +id
  +name
  +age
  +address
```

```
  +sex
  +salary
  +drawSalary()
}

class EXPENDITURE {
  +amt
  +bill_no
  +giveSalary()
}

PATIENT "1" -- "1" REGISTRATION : registers
REGISTRATION "1" -- "1" INCOME : issues bill
REGISTRATION "1" -- "1..*" WARD : allocates bed
PATIENT "1" -- "1..*" APPOINTMENT : takes appointment
WARD "1" -- "1" REPORTS : generates ward report
INCOME "1" -- "1" REPORTS : generates discharge report
REPORTS "1" -- "1" TEST_OPERATIONS
TEST_OPERATIONS "1..*" -- "1" DOCTOR_STAFF
EDIT "1..*" -- "1" DOCTOR_STAFF : edits doctor info
DOCTOR_STAFF "1" -- "1" EXPENDITURE : draws salary
REPORTS "1" -- "1" PATIENT : displays patient info

@enduml
```

Diagram:


PATIENT
id
name
age
address
sex
appdate
create()
Inpatient()
Ispatient()
registers
takes appointment
displays patient info
REGISTRATION
register()
alloted()
APPOINTMENT
date_dt
time_tm
allocates bed
issues bill
WARD
ward_no
no_beds
name
bedStatus()
name()
INCOME
amt
bill_no
addTestCharges()
addOprCharges()
addApptCharges()
addWardCharges()
generates ward report
generates discharge report
REPORTS
dispWardStatus()
dispAddsReport()
dispPatInfo()
TEST_OPERATIONS
patient_id
id
flag
opAppt()
testAppt()
docAppt()
getTest()
getOper()
EDIT
addDoctor()
addStaff()
delDoctor()
delStaff()
editDoc()
editStaff()
edits doctor info
DOCTOR_STAFF
id
name
age
address
sex
salary
drawSalary()
draws salary
EXPENDITURE
amt
bill_no
giveSalary()

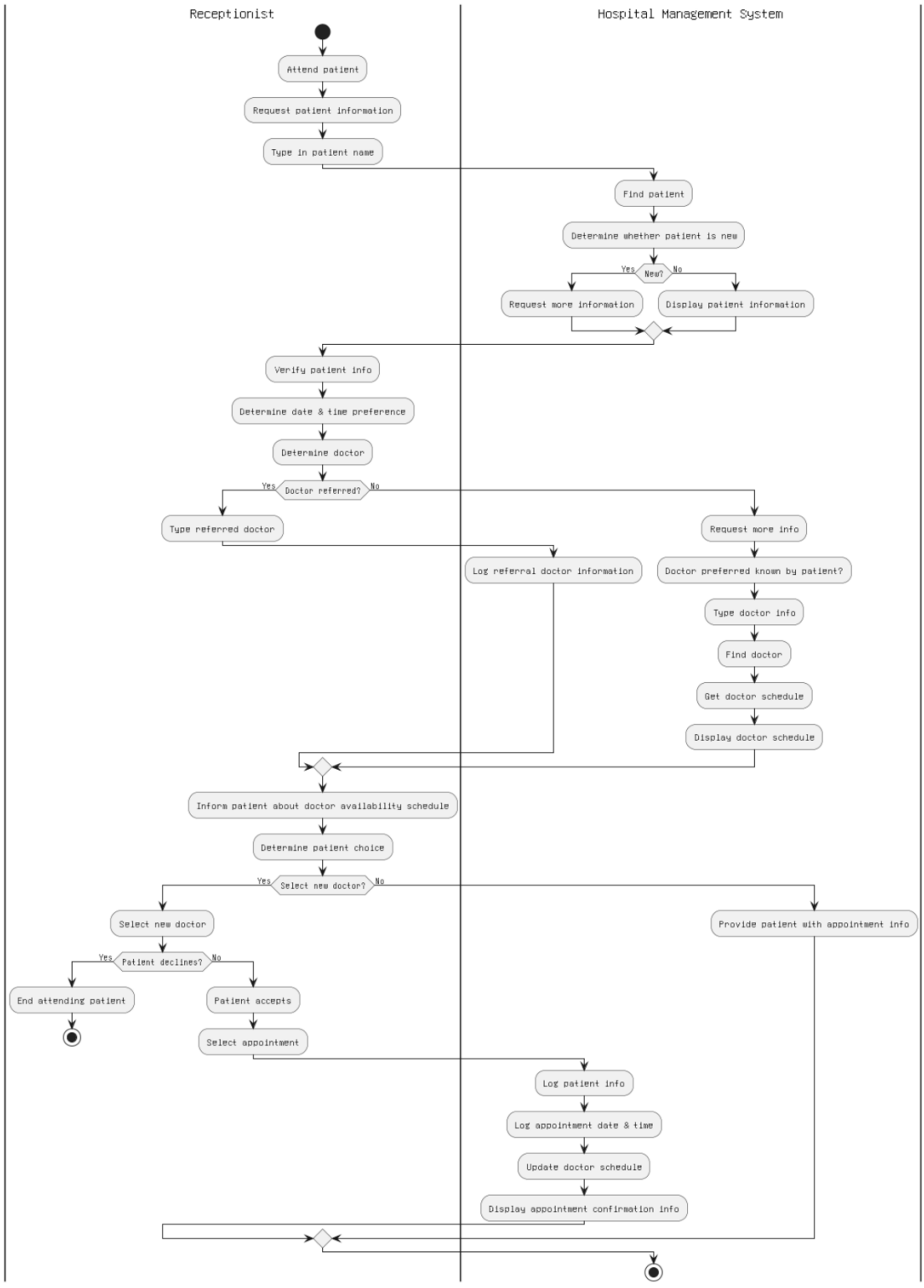
4) Activity Diagram

PlantUML code:

```
@startuml
|Receptionist|
start
:Attend patient;
:Request patient information;
:Type in patient name;
|Hospital Management System|
:Find patient;
:Determine whether patient is new;
if (New?) then (Yes)
  :Request more information;
else (No)
  :Display patient information;
endif
|Receptionist|
:Verify patient info;
:Determine date & time preference;
:Determine doctor;
if (Doctor referred?) then (Yes)
  :Type referred doctor;
  |Hospital Management System|
  :Log referral doctor information;
else (No)
  :Request more info;
  :Doctor preferred known by patient?;
  :Type doctor info;
  |Hospital Management System|
  :Find doctor;
  :Get doctor schedule;
  :Display doctor schedule;
endif
|Receptionist|
:Inform patient about doctor availability schedule;
:Determine patient choice;
if (Select new doctor?) then (Yes)
  :Select new doctor;
  if (Patient declines?) then (Yes)
    :End attending patient;
    stop
  else (No)
    :Patient accepts;
    :Select appointment;
    |Hospital Management System|
    :Log patient info;
    :Log appointment date & time;
```

```
   :Update doctor schedule;
   :Display appointment confirmation info;
  endif
else (No)
  :Provide patient with appointment info;
endif
stop
@enduml
```

Diagram:

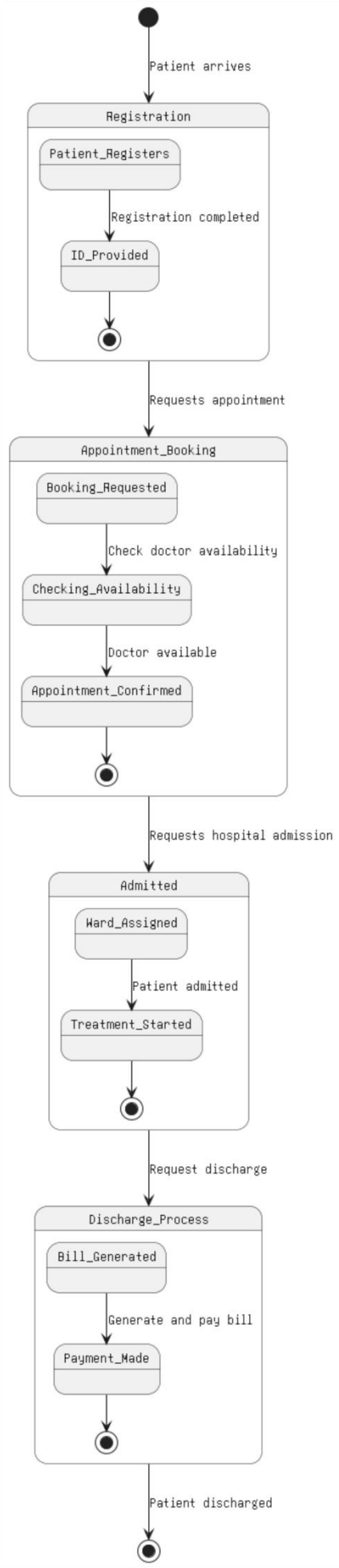


5) State Diagram

PlantUML code:

```
@startuml

[*] --> Registration : Patient arrives

state Registration {
  Patient_Registers --> ID_Provided : Registration completed
  ID_Provided --> [*]
}

Registration --> Appointment_Booking : Requests appointment
state Appointment_Booking {
  Booking_Requested --> Checking_Availability : Check doctor availability
  Checking_Availability --> Appointment_Confirmed : Doctor available
  Appointment_Confirmed --> [*]
}

Appointment_Booking --> Admitted : Requests hospital admission
state Admitted {
  Ward_Assigned --> Treatment_Started : Patient admitted
  Treatment_Started --> [*]
}

Admitted --> Discharge_Process : Request discharge
state Discharge_Process {
  Bill_Generated --> Payment_Made : Generate and pay bill
  Payment_Made --> [*]
}

Discharge_Process --> [*] : Patient discharged

@enduml
```

Diagram:


Patient arrives
Registration
Patient_Registers
Registration completed
ID_Provided
Requests appointment
Appointment_Booking
Booking_Requested
Check doctor availability
Checking_Availability
Doctor available
Appointment_Confirmed
Requests hospital admission
Admitted
Ward_Assigned
Patient admitted
Treatment_Started
Request discharge
Discharge_Process
Bill_Generated
Generate and pay bill
Payment_Made
Patient discharged

6) Collaboration Diagram

PlantUML code:

```
@startuml

object Patient
object Receptionist
object Doctor
object Registration
object Appointment
object Billing
object Ward
object Reports

Patient -> Receptionist : 1. Register request
Receptionist -> Registration : 2. Register Patient
Registration -> Patient : 3. Provide ID

Patient -> Receptionist : 4. Book appointment
Receptionist -> Appointment : 5. Check doctor availability
Appointment -> Doctor : 6. Confirm availability
Doctor -> Appointment : 7. Approve appointment
Appointment -> Patient : 8. Appointment confirmed

Patient -> Receptionist : 9. Request admission
Receptionist -> Ward : 10. Assign ward/bed
Ward -> Patient : 11. Admission confirmed

Doctor -> Reports : 12. Update patient records

Patient -> Receptionist : 13. Request discharge
Receptionist -> Billing : 14. Generate bill
Billing -> Patient : 15. Provide bill
Patient -> Billing : 16. Make payment

Billing -> Reports : 17. Update financial records
Reports -> Receptionist : 18. Generate reports

@enduml
```