CS3365 – Software Engineering **Fall 2020**

Project: HealthCare System (HCS)

Project Problem Description

You have been requested to develop a healthcare system with the following characteristics:

A hospital plans to build a healthcare system, which manages patient medical information. The hospital has a CEO, five doctors, five nurses, and two staff. A patient can make, change, or cancel an appointment by calling a staff in the hospital. When a patient makes an appointment, she/he tells a doctor name to a staff who can check if the doctor is available from 9 am to 5 pm on a half hourly basis. Each appointment is recorded in the system and updated whenever the patient change or cancel the appointment. An appointment will be cleared when a patient visits the doctor. In the case of no-show, where a patient makes an appointment but does not come to the hospital, the system clears the appointment automatically at 8 pm each day. The hospital does not accept walk-in patients.

For a new patient, a staff signs in the patient by creating a patient chart with patient information: patient id, name, address, phone number, email, social security number, and insurance name. Then the staff adds the chart to a list of patients being served by each doctor. For a returning patient, the staff signs in the patient by updating patient information on the patient chart if any changes and by adding the chart to a list of patients for a doctor. For each visit to doctor's office, a patient treatment record is added to the chart: date to visit a doctor, reason to visit, weight, height, blood pressure, treatment content, and prescription.

A nurse measures a patient's weight, height, and blood pressure every time a patient visits a doctor, updating the patient chart with the measurements. The nurse also enters the reason to visit to the patient chart. A doctor can view a patient chart before he/she meets a patient. After treating a patient, a doctor enters treatment content and prescription to the chart.

A patient pays for a doctor visit (e.g., copayment of insurance) to a staff before getting a treatment. A patient can pay it by credit/debit/cash. In the case of credit or debit card payment, the payment is validated by a card company (e.g., bank), which sends a reference number back to the hospital if approved. The reference number is stored in payment information, which includes patient id, name, date, amount, payment type (credit/debit/cash), and reference number. A patient receives a receipt for payment from a staff. If the bank does not approve a payment, the patient should pay it by another card or cash.

The system generates a daily summary report at 9 pm every business day. The report shows information about doctors' performance for a day. The report contains each doctor name, the number of patients served by a doctor, and the amount earned. The daily reports are stored in the system so that CEO can view them.

The employees can access patient medical information according to each employee role, such as CEO, doctor, nurse, and staff. A doctor has read/write permission on patient charts. A nurse has read/write permission on patient charts, but she/he cannot write down treatment and prescription. A staff can read/write only patient information on the chart and read/write patient payment information. The CEO can look at the daily reports.

You are required to develop:

Requirements Analysis:

a) Develop the static model for the healthcare system, which depicts the classes and their relationships. A class may be classified as an interface class, entity class, control class, or application logic class. Define the attributes of each entity class. (4 pts)

b) Develop the interaction model (using either communication diagram or sequence diagram) that depicts objects participating in each use case and the sequence of interactions among the objects. A use case is modeled using a communication diagram or sequence diagram. Add the classes to the static model in (a) if you identify new classes in the interaction model. (8 pts)

c) Define the operations of classes identified in (a) and (b). (2 pts)

Design:

d) Design the software architecture for the healthcare system where the architecture is defined in terms of subsystems and their interactions. Each subsystem should be represented with objects supporting the subsystem. (4 pts)

e) Design database tables to store information about appointments, patient charts, payments, and reports in the healthcare system. Define the primary key for each database table. (2 pts)

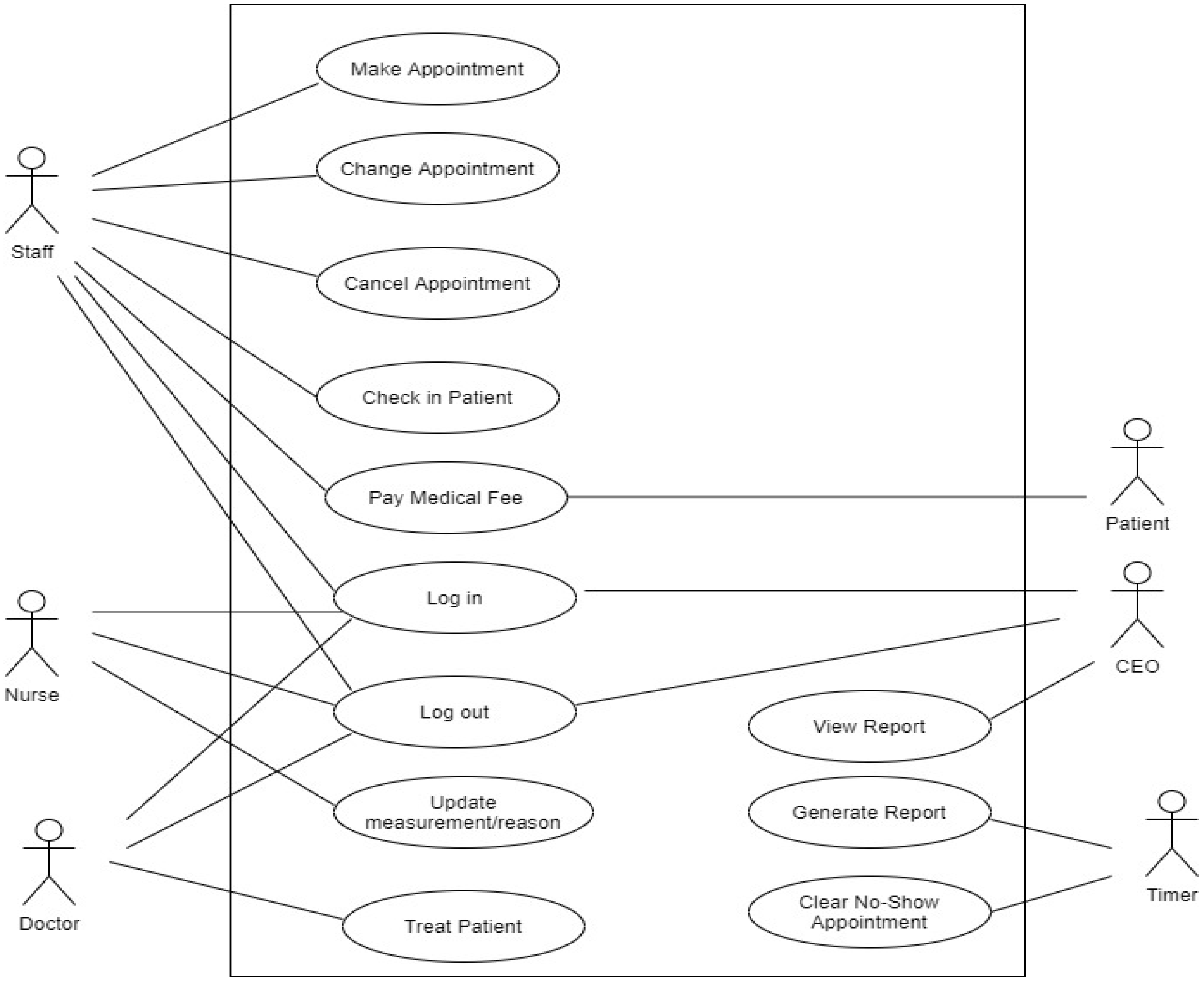
Implementation:

f) Implement the healthcare system in an object-oriented language such as Java. The data can be stored in a database, but you can store the data in files. Develop graphical user interface for your system. (10 pts)

Make your assumptions as necessary

The use case diagram for the healthcare system is given below together with use case descriptions.

Use Case Diagram for Healthcare System



Use Case Description for Healthcare System:

Use case name: Log in
Summary: Employee (staff, nurse, doctor, CEO) logs in system.
Actor: Employee (staff, nurse, doctor, CEO)
Precondition: none.
Main sequence:

1. Employee enters ID and password to the system.
2. System checks if customer ID and password are correct.
3. System displays a welcome message if ID and password are correct.

Alternative sequence:

- Step 3: If employee ID or password is not correct, system displays that ID or password is incorrect.

Postcondition: Employee has logged in system.

Use case name: Log out

Summary: Employee (staff, nurse, doctor, CEO) logs out system.

Actor: Employee (staff, nurse, doctor, CEO)

Precondition: Employee logged in system.

Main sequence:

1. Employee selects "log out".
2. System ensure that employee wants to log out.
3. Employee confirms to log out.
4. System gets employee to log out.

Alternative sequence:

- Step 2: If employee does not confirm, system does not log out and exits.

Postcondition: Employee has logged out.

Use case name: Make appointment

Summary: Staff makes an appointment for a patient.

Actor: Staff

Precondition: None.

Main sequence:

1. Staff enters a doctor name with whom a patient wants to make an appointment.
2. System displays a doctor's available date and time.
3. Staff enters patient name, date, and time if a doctor is available on a date and time that a patient wants.
4. System stores a patient appointment with patient name, doctor name, date, and time.

Alternative sequence:

- Step 3: If a doctor is unavailable on the date and time that a patient wants, staff exits.

Postcondition: Staff has made an appointment for a patient.

Use case name: Change appointment

Summary: Staff changes a patient appointment.

Actor: Staff

Precondition: A patient made an appointment.

Main sequence:

1. Staff enters patient name to the system.
2. System displays patient appointment information if patient made an appointment.
3. Staff enters a doctor name with whom a patient made an appointment.
4. System displays a doctor's available date and time.
5. Staff enters a date and time that a patient wants to change.
6. System changes appointment date and time for a patient.

Alternative sequence:

- Step 2: If patient did not make an appointment, system displays "no appointment."
- Step 5: If a doctor is unavailable on the date and time that a patient wants to change, staff does not change appointment or cancels it.

Postcondition: Staff has changed an appointment.

Use case name: Cancel appointment
Summary: Staff cancels an appointment.
Actor: Staff
Precondition: A patient made an appointment.
Main sequence:

1. Staff enters patient name to the system.
2. System displays patient appointment information if patient made an appointment.
3. Staff cancels patient appointment.
4. System deletes patient appointment.

Alternative sequence:

- Step 2: If patient did not make an appointment, system displays “no appointment.”

Postcondition: Staff has cancelled an appointment.

Use case name: Check-in patient
Summary: Staff checks in a patient to system.
Actor: Staff
Precondition: Patient had an appointment and staff had logged in.
Main sequence:

1. Staff enters patient name and date of birth to system.
2. System checks if staff (employee) has permission to view/update patient information using employee login ID.
3. System displays patient chart if patient is in the system (i.e., returning patient) and staff (employee) has permission.
4. Staff enters patient information (address, phone number, SSN, and health insurance) if any change is made in patient information.
5. System updates patient information.
6. System displays patient chart to staff.
7. Staff enters doctor name to add patient chart.
8. System attaches a treatment record to patient chart for patient visit and then adds patient chart to doctor.
9. System creates patient payment information (patient id, name, date, amount, payment type, and reference number) to charge the doctor visit.
10. System deletes the patient appointment.

Alternative sequence:

- Step 3: If staff (employ) has no permission, system displays “access denied”.
- Step 3: If patient is new, staff enters patient information and then system creates patient chart and then go to step 5.
- Step 4: If there is no change in patient information, go to step 6.

Postcondition: Staff checked in patient.

Use case name: Pay medical fee
Summary: Patient pays medical fee.
Actor: Staff, Patient, Bank
Precondition: Staff had logged in.

Main sequence:

1. Staff enters patient name for payment.
2. System checks if staff (employee) has permission to view/update patient payment information using employee login ID.
3. System displays patient payment information if staff (employee) has permission.
4. System activates card reader if patient wants to pay it by credit/debit.
5. Patient swipes credit or debit card for payment.
6. If card is debit, patient enters PIN.
7. System requests bank approving the payment. If bank approves payment, bank creates a reference number and returns it back to the system.
8. System updates patient payment information.
9. System prints a receipt for patient.

Alternative sequence:

- Step 3: If staff (employee) has no permission, system displays "access denied".
- Step 4: If patient wants to pay it by cash, staff enters payment information, and go to step 8.
- Step 7: If bank denies payment, patient should pay it by another card or cash.

Postcondition: Patient paid medical fee.

Use case name: Update Measurements/Reason
Summary: Nurse updates patient chart with physical measurements and reason to visit to doctor.
Actor: Nurse (employee)
Precondition: Nurse had logged in.
Main sequence:

1. Nurse enters patient name.
2. System checks if nurse (employee) has permission to view/update patient weight, height, blood pressure, and reason to visit using employee login ID.
3. System displays patient chart if nurse (actor) has permission.
4. Staff adds patient weight, height, blood pressure, and reason to visit.
5. System updates patient chart.

Alternative sequence:

- Step 3: If nurse (employee) has no permission, system displays "access denied".

Postcondition: Nurse updated patient chart.

Use case name: Treat patient
Summary: Doctor treats patient and updates patient chart.
Actor: Doctor (employee)
Precondition: Nurse updated patient chart and doctor had logged in.
Main sequence:

1. Doctor enters patient name.
2. System checks if doctor (employee) has permission to view/update patient chart for treatment content and prescription using employee login ID.
3. System displays patient chart if doctor (employee) has permission.
4. Doctor enters treatment content and prescription to system.
5. System updates patient chart.

Alternative sequence:

- Step 3: If doctor (employee) has no permission, system displays "access denied".

None.
Postcondition: Doctor updated patient chart.

Use case name: View Report
Summary: CEO views daily summary reports.
Actor: CEO
Precondition: Report has been generated and CEO had logged in.
Main sequence:

1. CEO requests daily reports.
2. System checks if CEO (employee) has permission to view daily reports using employee login ID.
3. System displays daily summary reports if CEO (employee) has permission.

Alternative sequence:

- Step 3: If CEO (employee) has no permission, system displays "access denied".

Postcondition: CEO had viewed daily summary reports.

Use case name: Generate report
Summary: System generates a daily summary report.
Actor: Timer
Precondition:
Main sequence:

1. Timer enters timer input to system.
2. System generates a daily summary report if it is 9 pm.
3. System stores the report.

Alternative sequence: None
Postcondition: A daily summary report has been generated.

Use case name: Clear No-Show Appointment
Summary: System clears daily no-show appointments.
Actor: Timer
Precondition:
Main sequence:

1. Timer enters timer input to system.
2. System clears no-show appointments on a date if it is 8 pm.

Alternative sequence: None
Postcondition: A daily no-show appointments has been cleared.