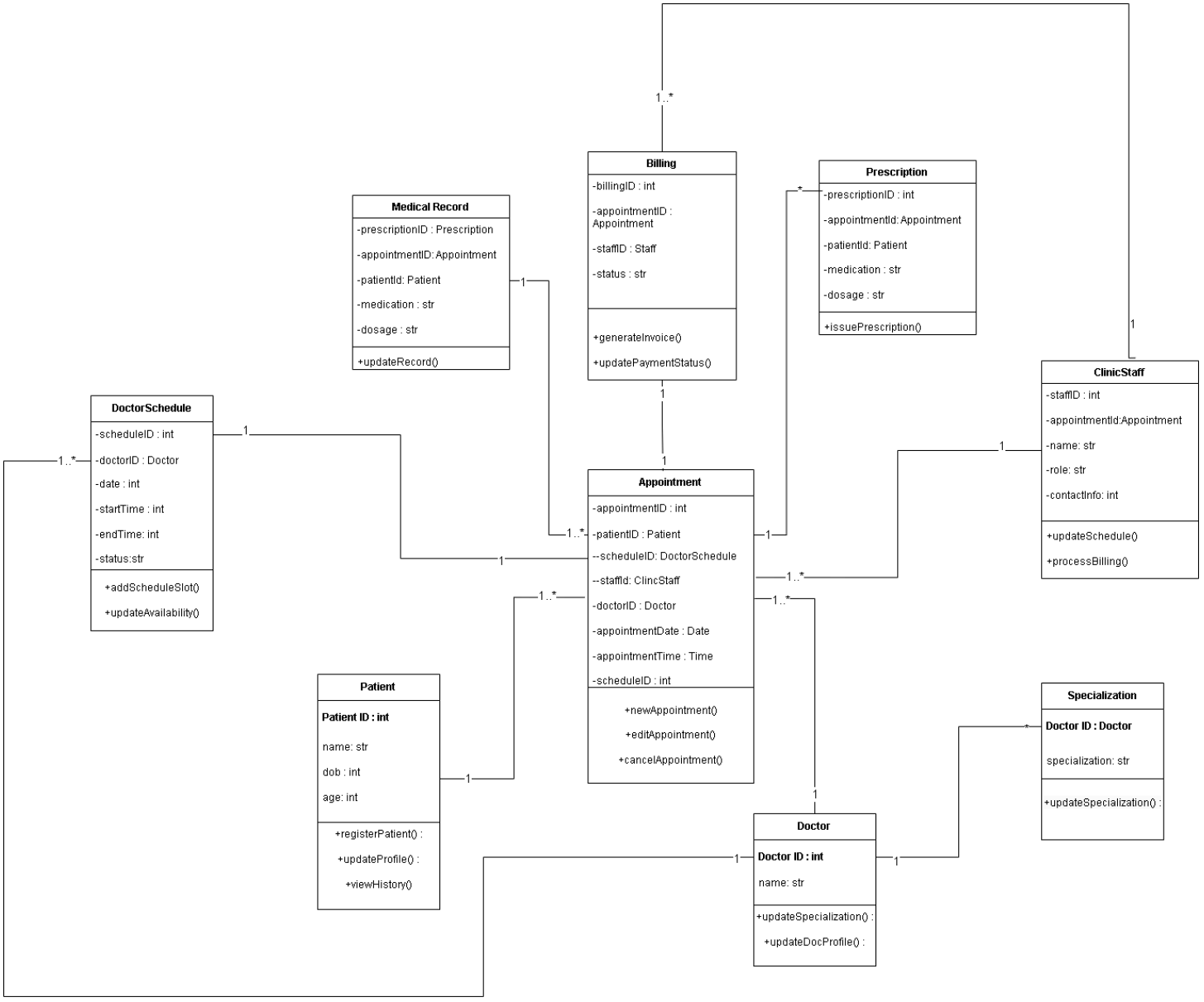
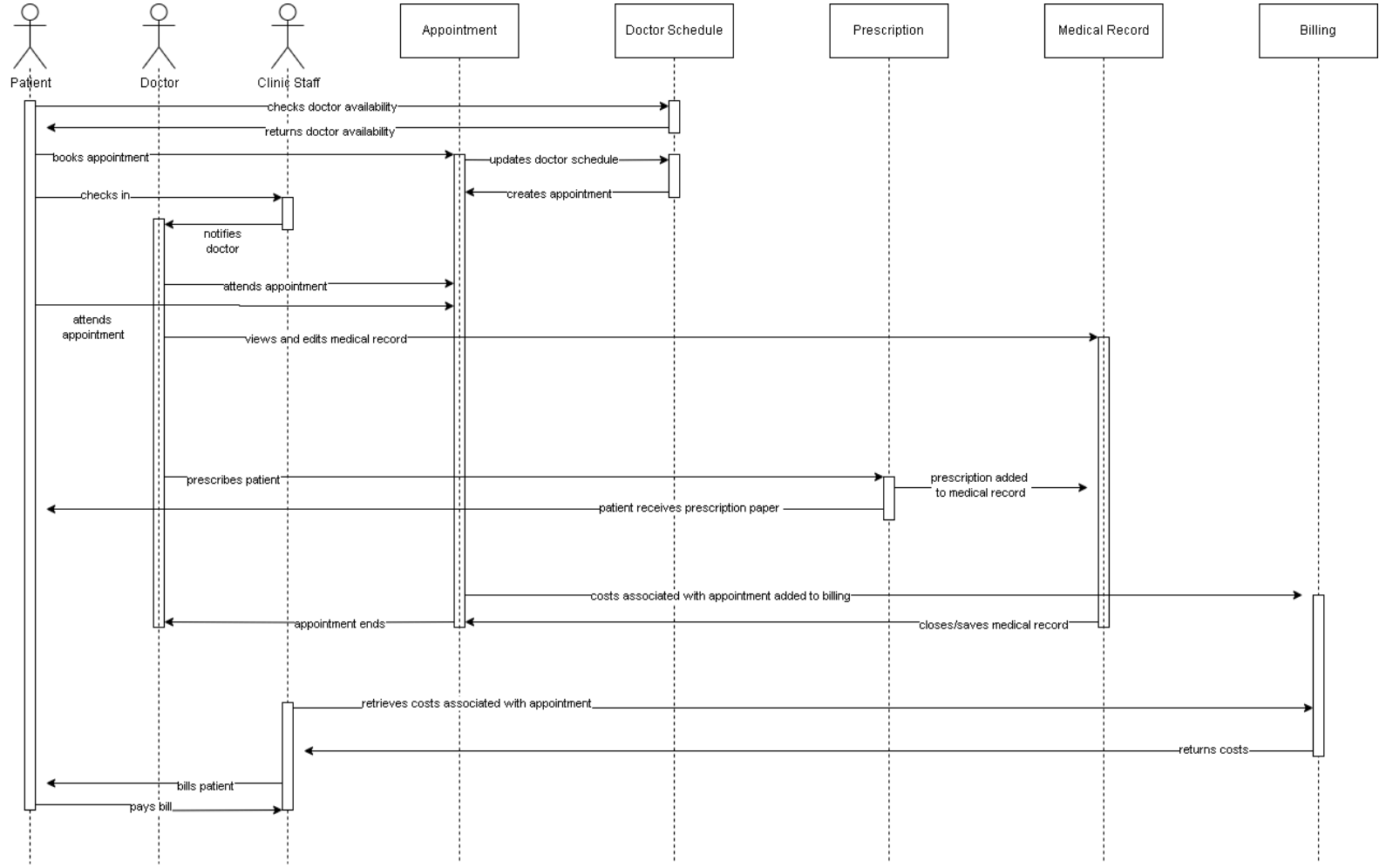
Intermediate Project Report

Danielle Park, Gaurav Gulati, Rhett Bramfield
Faculty of Science, University of Calgary
CPSC 471: Database Management System
Prof. Reda Alhajj
March 17, 2025

UML Diagram


Billing
-billingID : int
-appointmentID : Appointment
-staffID : Staff
-status : str
+generateInvoice()
+updatePaymentStatus()
Prescription
-prescriptionID : int
-appointmentId: Appointment
-patientId: Patient
-medication : str
-dosage : str
+issuePrescription()
Medical Record
-prescriptionID : Prescription
-appointmentID: Appointment
-patientId: Patient
-medication : str
-dosage : str
+updateRecord()
ClinicStaff
-staffID : int
-appointmentId:Appointment
-name: str
-role: str
-contactInfo: int
+updateSchedule()
+processBilling()
DoctorSchedule
-scheduleID : int
-doctorID : Doctor
-date : int
-startTime : int
-endTime: int
-status:str
+addScheduleSlot()
+updateAvailability()
Appointment
-appointmentID : int
-patientID : Patient
--scheduleID: DoctorSchedule
--staffId: ClincStaff
-doctorID : Doctor
-appointmentDate : Date
-appointmentTime : Time
-scheduleID : int
+newAppointment()
+editAppointment()
+cancelAppointment()
Patient
Patient ID : int
name: str
dob : int
age: int
+registerPatient() :
+updateProfile() :
+viewHistory()
Specialization
Doctor ID : Doctor
specialization: str
+updateSpecialization() :
Doctor
Doctor ID : int
name: str
+updateSpecialization() :
+updateDocProfile() :
1..*
1
*

Sequence Diagram



Description

1. **Patient Books an Appointment Online**
    - The patient initiates an online booking request.
    - The Appointment system checks the Doctor Schedule for availability before creating an appointment.
    - The Doctor Schedule returns available time slots to the patient.
    - The patient selects a suitable time slot from the available options.
    - The Appointment system confirms the booking and updates the Doctor Schedule accordingly.
2. **Patient Checks In at the Clinic**
    - The patient arrives at the clinic and checks in.
    - The clinic system notifies the doctor that the patient has arrived.
3. **Patient and Doctor Attend the Appointment**
    - The patient and doctor enter the consultation room for the appointment.
    - The doctor accesses the patient's medical record and updates it if needed.
4. **Doctor Prescribes Medication**
    - The doctor prescribes medication to the patient.
    - The Prescription system generates the prescription.
    - The patient receives the prescription (either paper or digital).

- The prescription is added to the patient's medical record.

5. **Appointment Ends and Billing is Processed**
   - The appointment concludes.
   - The system calculates the costs associated with the appointment and sends them to the Billing system.
   - The Medical Record system saves and closes the patient's file.
6. **Patient Pays the Bill**
   - The clinic staff retrieves the cost details of the appointment from the Billing system.
   - The clinic staff generates the bill via the Billing system for the patient.
   - The patient pays the bill and completes the process.

Structured Query Language (SQL)

All create Table and create Database

```
CREATE DATABASE clinic_db;

CREATE TABLE Patient (
   patientID   SERIAL PRIMARY KEY,
   name        VARCHAR(100) NOT NULL,
   dob         DATE,
   age         INT
);

CREATE TABLE Doctor (
   doctorID        SERIAL PRIMARY KEY,
   name            VARCHAR(100) NOT NULL,
   specialization  VARCHAR(100)
);

CREATE TABLE ClinicStaff (
   staffID     SERIAL PRIMARY KEY,
   name        VARCHAR(100) NOT NULL,
   role        VARCHAR(50),
   contactInfo VARCHAR(200)
);

CREATE TABLE DoctorSchedule (
   scheduleID  SERIAL PRIMARY KEY,
   doctorID    INT NOT NULL,
   scheduleDate DATE NOT NULL,
   startTime   TIME NOT NULL,
   endTime     TIME NOT NULL,
   status      VARCHAR(50),
   CONSTRAINT fk_schedule_doctor
```

```
    FOREIGN KEY (doctorID)
    REFERENCES Doctor(doctorID)
    ON DELETE CASCADE
);

CREATE TABLE Appointment (
  patientID      INT  NOT NULL,
  appointmentID  INT  NOT NULL,
  appointmentDate DATE NOT NULL,
  appointmentTime TIME NOT NULL,
  doctorID       INT  NOT NULL,
  staffID        INT  NOT NULL,
  scheduleID     INT,
  PRIMARY KEY (patientID, appointmentID),
  CONSTRAINT fk_appointment_patient
    FOREIGN KEY (patientID)
    REFERENCES Patient(patientID)
    ON DELETE CASCADE,

  CONSTRAINT fk_appointment_doctor
    FOREIGN KEY (doctorID)
    REFERENCES Doctor(doctorID)
    ON DELETE CASCADE,

  CONSTRAINT fk_appointment_staff
    FOREIGN KEY (staffID)
    REFERENCES ClinicStaff(staffID)
    ON DELETE CASCADE,

  CONSTRAINT fk_appointment_schedule
    FOREIGN KEY (scheduleID)
    REFERENCES DoctorSchedule(scheduleID)
    ON DELETE SET NULL
);

CREATE TABLE Prescription (
  prescriptionID  SERIAL PRIMARY KEY,
  medication      VARCHAR(100) NOT NULL,
  dosage          VARCHAR(50),
  -- references Appointment's composite key
  patientID       INT NOT NULL,
  appointmentID   INT NOT NULL,

  CONSTRAINT fk_prescription_appointment
    FOREIGN KEY (patientID, appointmentID)
    REFERENCES Appointment(patientID, appointmentID)
    ON DELETE CASCADE
);
```

```
CREATE TABLE MedicalRecord (
    recordID      SERIAL PRIMARY KEY,
    diagnosis     TEXT,
    treatment     TEXT,
    patientID     INT NOT NULL,
    appointmentID INT NOT NULL,

    CONSTRAINT fk_mrecord_appointment
        FOREIGN KEY (patientID, appointmentID)
        REFERENCES Appointment(patientID, appointmentID)
        ON DELETE CASCADE
);

CREATE TABLE Billing (
    billingID     SERIAL PRIMARY KEY,
    amount        DECIMAL(10,2),
    paymentStatus VARCHAR(50),
    patientID     INT NOT NULL,
    appointmentID INT NOT NULL,
    staffID       INT,

    CONSTRAINT fk_billing_appointment
        FOREIGN KEY (patientID, appointmentID)
        REFERENCES Appointment(patientID, appointmentID)
        ON DELETE CASCADE,

    CONSTRAINT fk_billing_staff
        FOREIGN KEY (staffID)
        REFERENCES ClinicStaff(staffID)
        ON DELETE SET NULL
);
```

Other SQL Statements

INSERT INTO Example

```
INSERT INTO Patient (name, dob, age)
VALUES
  ('Alice Smith', '1990-01-15', 34),
  ('Bob Johnson', '1985-09-10', 39);

INSERT INTO Doctor (name, specialization)
VALUES
  ('Dr. Carter', 'General Practitioner'),
  ('Dr. Lin', 'Pediatrics');
```

```
INSERT INTO ClinicStaff (name, role, contactInfo)
VALUES
  ('Sarah Gray', 'Receptionist', 'sarah@clinic.com'),
  ('Michael Brown', 'Billing Specialist', 'michael@clinic.com');

INSERT INTO DoctorSchedule (doctorID, scheduleDate, startTime, endTime, status)
VALUES
  (1, '2025-03-21', '08:00', '12:00', 'Open'),
  (1, '2025-03-21', '13:00', '16:00', 'Open'),
  (2, '2025-03-21', '09:00', '12:00', 'Open');

INSERT INTO Appointment (
  patientID, appointmentID, appointmentDate, appointmentTime, doctorID, staffID,
scheduleID
)
VALUES
  (1, 1, '2025-03-21', '10:00', 1, 1, 1);

INSERT INTO Prescription (medication, dosage, patientID, appointmentID)
VALUES
  ('Amoxicillin', '500mg', 1, 1),
  ('Cough Syrup', '10ml', 1, 1);

INSERT INTO MedicalRecord (diagnosis, treatment, patientID, appointmentID)
VALUES
  ('Common Cold', 'Bed rest, plenty of fluids', 1, 1);

INSERT INTO Billing (amount, paymentStatus, patientID, appointmentID, staffID)
VALUES
  (120.00, 'Pending', 1, 1, 2);
```

SELECT Example

```
SELECT patientID, name, dob, age
FROM Patient;

SELECT
    A.patientID,
    A.appointmentID,
    A.appointmentDate,
    D.name     AS doctor_name,
    D.specialization
FROM Appointment A
JOIN Doctor D
  ON A.doctorID = D.doctorID
```

```
ORDER BY A.appointmentDate, A.appointmentTime;

SELECT
   B.billingID,
   B.amount,
   B.paymentStatus,
   B.patientID,
   B.appointmentID,
   C.name AS processedByStaff
FROM Billing B
LEFT JOIN ClinicStaff C
  ON B.staffID = C.staffID
WHERE B.patientID = 1;
```

UPDATE Example

```
UPDATE Patient
SET age = 35
WHERE patientID = 1;

UPDATE Billing
SET paymentStatus = 'Paid'
WHERE billingID = 1;
```

DELETE Example

```
DELETE FROM Prescription
WHERE prescriptionID = 2;


DELETE FROM Patient
WHERE patientID = 2;
```