SOFTWARE DESIGN & ANALYSIS

ASSIGNMENT #3

Submitted By

Tahawar ihsan, M. Mohsin Raza, Matti Ur Rehman

(19P-0097) (19p-0072) (19p-0048)

Submitted to

USAMA MUSHARAF

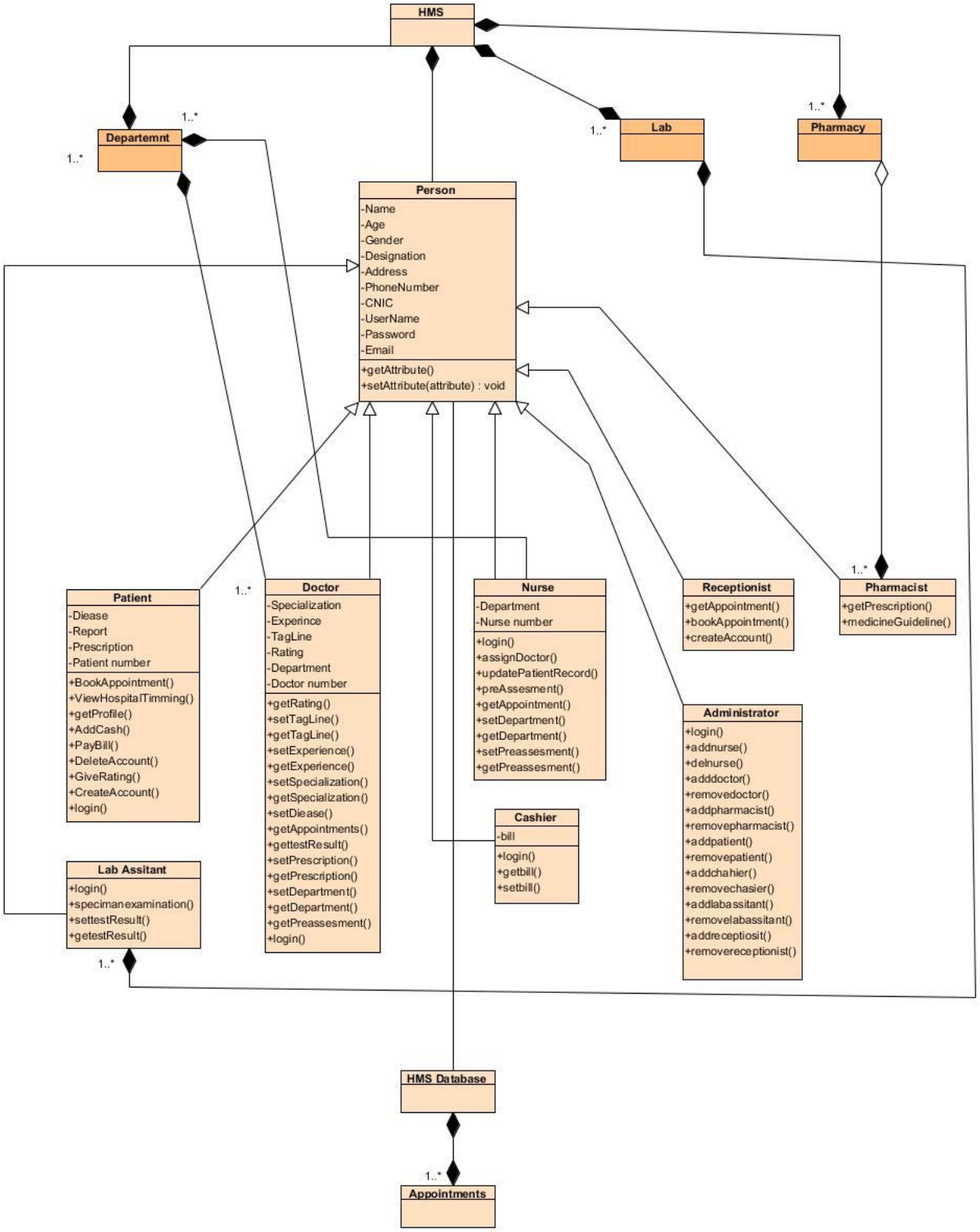
(INSTRUCTOR CS)

DEPARTMENT OF COMPUTER SCIENCE

FAST NATIONAL UNIVERSITY OF COMPUTER AND EMERGING SCINCES, PESHAWAR

Session 2019-2023

Class diagram:



AFTER ANALYSIS AND USING SOLID principles and design patterns.

A) SOLID Principles

1) Open Close Principle:

In our class diagram we apply Open Close Principle in our class diagram we can add new classes with out changing data in other class. You can simply see this Principle in our diagram. For example you add new class inherit with person class without changing into any other class.

2) LISKOV SUBSTITUTION PRINCIPLE:

In our class diagram we apply LISKOV SUBSTITUTION PRINCIPLE by adding this we inherit that class together which have same function under same class and other in different class and at the end we combine these classes with super class. In our case we have function getattribute() and setattribute() which are use with their base class

B) Design pattern

1) Builder Design Pattern

In our class diagram we use Builder Design Pattern. Like for HMS we have different classes and combine them in that way at the end it make for us complete HMS.

2)Bridge Design Pattern

In our class diagram we use Bridge Design Pattern for everything we use sperate class like lab class and pharmacy class we

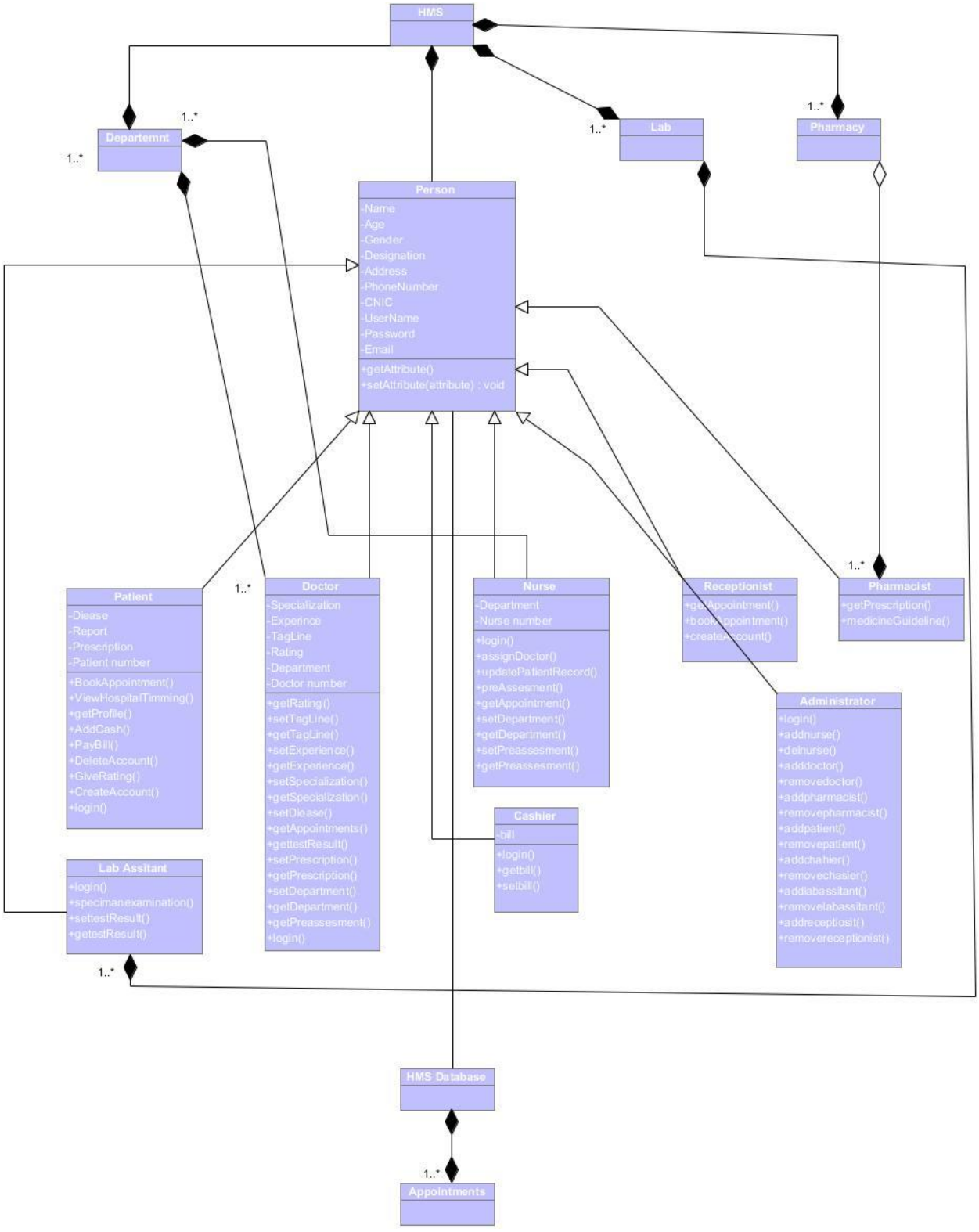
divide every class in that way and connect it with super class that at the time we call and do things.

3) Decorator Design Pattern

We also use Decorator Design Pattern in our class

4) Chain of Responsibility

In our class diagram we also have Chain of Responsibility in that after main class we have different classes after some choice we move to next class and after using different class we move to next class .


HMS
Departemnt
Lab
Pharmacy
Person
-Name
-Age
-Gender
-Designation
-Address
-PhoneNumber
-CNIC
-UserName
-Password
-Email
+getAttribute()
+setAttribute(attribute) : void
Patient
-Diease
-Report
-Prescription
-Patient number
+BookAppointment()
+ViewHospitalTimming()
+getProfile()
+AddCash()
+PayBill()
+DeleteAccount()
+GiveRating()
+CreateAccount()
+login()
Doctor
-Specialization
-Experince
-TagLine
-Rating
-Department
-Doctor number
+getRating()
+setTagLine()
+getTagLine()
+setExperience()
+getExperience()
+setSpecialization()
+getSpecialization()
+setDisease()
+getAppointments()
+gettestResult()
+setPrescription()
+getPrescription()
+setDepartment()
+getDepartment()
+getPreassesment()
+login()
Nurse
-Department
-Nurse number
+login()
+assignDoctor()
+updatePatientRecord()
+preAssesment()
+getAppointment()
+setDepartment()
+getDepartment()
+setPreassesment()
+getPreassesment()
Receptionist
+getAppointment()
+bookAppointment()
+createAccount()
Pharmacist
+getPrescription()
+medicineGuideline()
Cashier
-bill
+login()
+getbill()
+setbill()
Administrator
+login()
+addnurse()
+delnurse()
+adddoctor()
+removedoctor()
+addpharmacist()
+removepharmacist()
+addpatient()
+removepatient()
+addchahier()
+removechasier()
+addlabassitant()
+removelabassitant()
+addreceptiosit()
+removereceptionist()
Lab Assitant
+login()
+specimanexamination()
+settestResult()
+getestResult()
HMS Database
Appointments
1..*