Name: Ahmed sattar
Roll no: p19-1672
Project: SDA

PROJECT CONTENT

Analysis and Design of Hospital Management System.

Phase-1: Requirement Analysis
Requirement Analysis and Specification
Phase-2: Architectural Design
Select an architectural style for your problem statement. (Hospital Management System).
Describe your architecture with multiple views and view-points. (See conceptual model of ISO/IEC/IEEE 42010).
Phase-3: Architectural Evaluation
Architecture Evaluation (Qualitatively)
Architecture Evaluation (Quantitatively by using metrics).
How your proposed architecture address complexity, reusability, maintainability and reliability.
Phase-4: Detail Design
Identify Classes of each component by extracting nouns from the requirements.

Eliminate irrelevant/ vague classes.
Identify possible associations
Identify attributes of classes
Identify functions of classes using sequence diagram. (Sequence diagram would help you to find out the interaction between objects).
Note: Follow design patterns.

Phase-1: Requirement Analysis

in requirement analysis we use story card, in HRM system have different user like Patient, Doctor, nurse, reception, laborarty, pharmcy etc we waite the story cade reladed to all the user requirement.

User stories cards for Requirement analysis:

	Login/signup
• Summary	To access and enter these website
• Dependency	none

• Actors	Patients, Receipts’
• Preconditions	Show home page
• Flow of Events	Activity Diagram
• Alternatives	if the system does not match password then reset
• Post condition	Login successfully

View appointment

• Summary	Check all the appointment history and later appointmemnt
• Dependency	Login/sign up, Book Appointment
• Actors	Patients, Receipts’
• Preconditions	Display the list of appointment.
• Flow of Events	Activity Diagram
Alternatives	If the patients not signup or login and Book the Appointment so patient not see the list
If patients open the list directly without book appointment show the Book Appointment screen show	
• Post condition	See the appointment history, receipts see the all appointment list.

Book appointmnet

• Summary	Booking the appointment with relater diagnose doctor
• Dependency	Sign up/login

- Actors Patients, receipts
- Preconditions Sign up / login ,
- Flow of Events Activity Diagram

Alternatives if the appointment is not book,then complete your info and book apoointment

If book appointment is cancel means the doctor not available that time so your appointment is cancel

- Post condition Confirmation massages display your appointment is book

Receptionist

- Summary The receptionist
- Dependency Sign up/login, view appointment
- Actors Receipts
- Preconditions View appointment list and check which doctor available
- Flow of Events Activity Diagram

Alternatives If the patient does not register these websites the receipts firstly register the patients and the book the appointment than she assigns the number

- Post condition After the view appointment she assign a token no

Pre-assesment examination

- Summary Examine the patient based on problem explain by the patient and according to that explanation or by viewing the patient history and diagnose it and generate a report.
- Dependency Sign up / login, view Appointment

- Actors Nurse, doctor
- Preconditions See the profile of patient and update and add the discerption, and doctor see the
- Flow of Events Activity Diagram

Alternatives If the nurse does not update the profile of patients. its check the

- Post condition Add the discerption for the see the doctor.

laboratory

- Summary process the specimans,analyze and generate the test result
- Dependency Sign up / login,
- Actors lab assistant
- Preconditions Doctor send the medical test of patients to the lab assistant and assistant see the test of patient
- Flow of Events Activity Diagram

Alternatives the lab assistant receives a message from doctor but not see the patient reports .

Lab assistant reopens the browser and see this face same issue,so he has access to see patient report description.

- Post condition Lab assistants send the test report to doctor and also update the patients description.

pharmacist

- Summary See the prescription made by the doctor for patient
- Dependency none
- Actors pharmacist

- Preconditions Doctor send the medicine description the pharmacist see the description and ready the doctor
- Flow of Events Activity Diagram

Alternatives none

- Post condition handover the medicine to the patient

payement

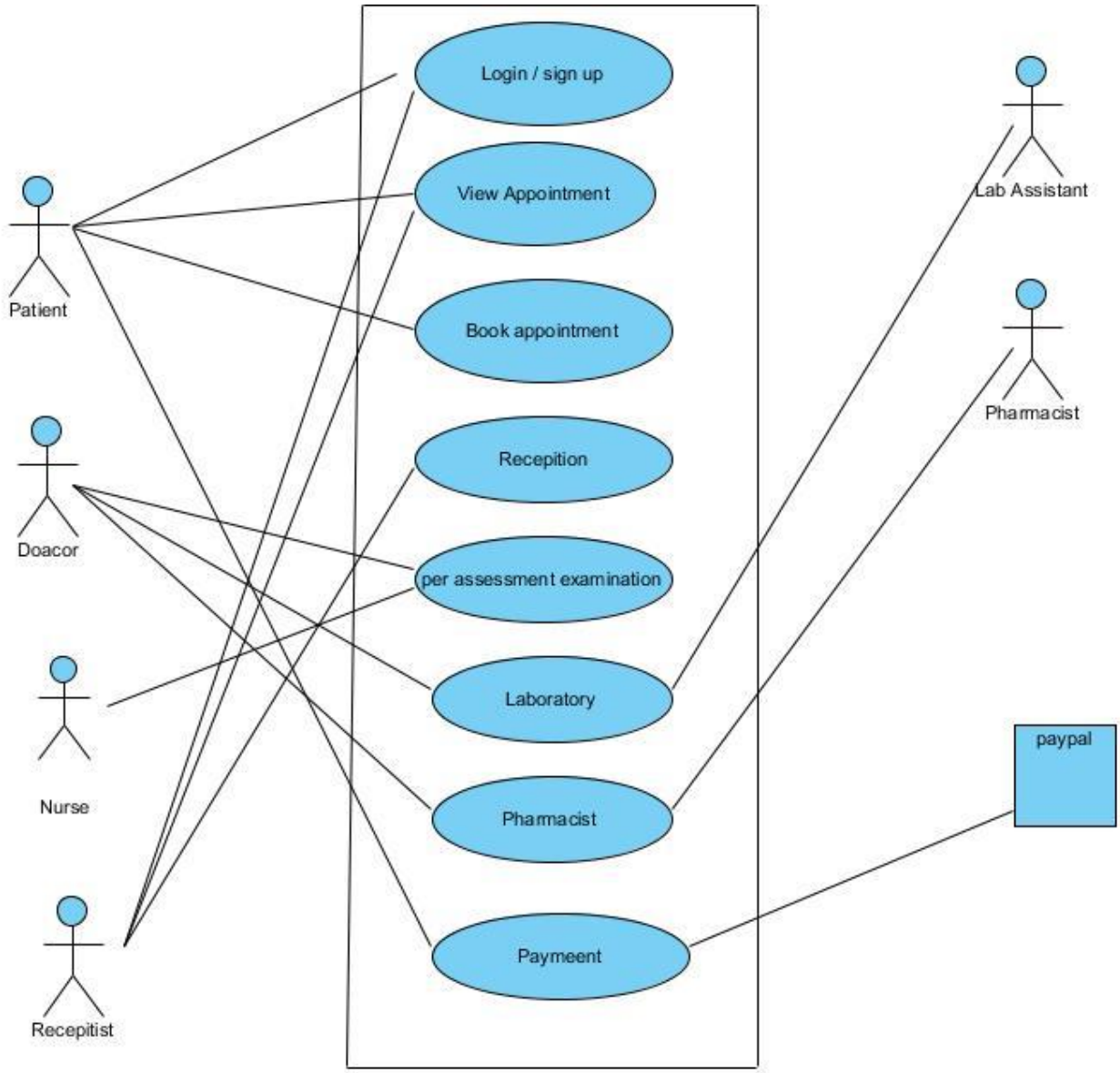
- Summary **to** see the patient payment history and also see If the patients purchase medicine then he /she pay for that if the patients just only checkup with the doctor and not purchase the medicine then he/she pay for medicine
- Dependency paypal
- Actors patient
- Preconditions but the medicine then pay for it
- Flow of Events Activity Diagram

Alternatives If the system paypal not working, the server is down

You pay by your easy-peasy account if the money is present in your account

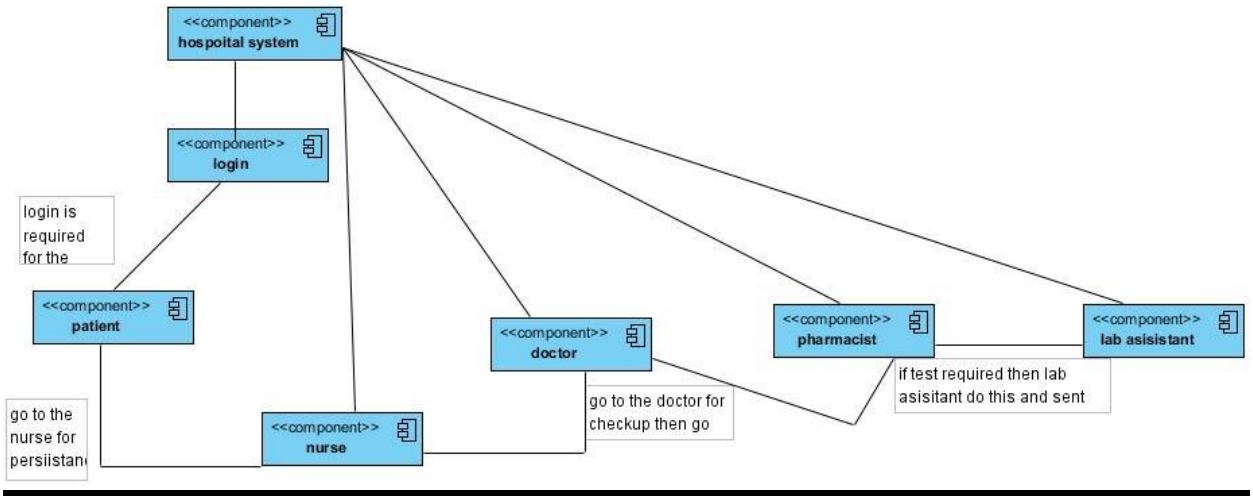
If your all account have no money you can pay with your cash

- Post condition payment is deposit successfully/



Phase-2: Architectural Design

Component based architecture



Views:

	Stakeholder	Data	Function	Time	people
Scope	patient	Data entity view(class model)	Profile view, view appointment, book appointment	To save the patient time	Patient, doctor, lab assistant
	Doctor	Data view,	View the profile of her and update the pre examination of patient,and send the labarorty and pharmacy	Performance view	Lab assistant, pharmacist, patient,
	Nurse	Data view	Can view and update the patient pre examination	Saving and performance of doctor time	Doctor, patient
	Reception	View and create data	View the appointment of patent and also register the patient	Saving the patient time	
	Pharmacy	View the data	View the doctor description		Doctor, patient

System model	Designer	Logical data view	Application to application communication	Process vies, Quality of services view	Usability, complexity

View Point:

Viewpoint Element	Description
Stakeholders	Management staff, staff and patient Information
Concerns	Show the top-level relationships between hospital staff, patients and all functions.
Modeling technique	Class object diagram. Classes =object and main actor; object = main functions.

Phase-3: Architectural Evaluation

Architecture Evaluation (Quantitatively)

After evaluation of **Hospital Management System**

Size Metrics

1) **No of components:**
   No of components (NC) metric is used to count the number of components in a system. The NC metric indicates the system size that is used in estimating the complexity of the system.

$$NC(SyS) = \sum_{C \epsilon SyS}^{n} 1$$

2) **No of Operations in a components**

- NO is a count of all the operations of a component to indicate the complexity of a component.

$$NO(C) = \sum_{O \epsilon C}^{n} 1$$

- To calculate the overall NO of all components in a system.

$$NO(SyS) = \sum_{O \epsilon SyS}^{n} 1$$

3) No of operations in a system

S.NO	No of components	No of Operations in a components	No of operations in a system
1	Patient	5	17
2	Doctor	3	
3	Nurse	1	
4	Pharmacy	3	
5	Lab assistant	3	
6	Receipts	2	

Coupling Metrics:

1) Direct coupling

Direct Coupling

- The direct coupling can be defined as the direct interactions between consumers and providers. Direct coupling can be calculated by counting all the direct consumer calls for a specific provider (component) as shown below.

  DC (p) = C (p)

2) Indirect coupling

Indirect Coupling (IC)

- The indirect coupling was calculated by counting the direct consumers calls for specific providers and then assume consumers as providers and calculate their coupling.

$$IC(p) = DC(p) + \sum_{c(p)\epsilon P} \text{IC(c(p))}$$

3) Total coupling

Total Coupling

- The total coupling can be calculated by the addition of direct and indirect coupling, as shown in the equation below.

  $TCoup = DC(p) + IC(p)$

4) **Coupling factor**

Coupling Factor (CoupF)

- The quantitative result of total coupling could not be interpreted directly, because this number may give a good indication if the system is big or worse if it is small. This means that the size of the system is needed to understand the value of this metric. If we suppose f = NC (SyS) + NO (SyS) then:

$$CopF(SyS) = \frac{TCoup(SyS)}{f2 - f}$$

S.NO	**No of Component**	**Direct coupling**	**Indirect coupling**	
1	Patient	4	1	
2	Doctor	1	1	
3	Nurse	1	1	
4	Pharmacy	2	0	
5	Lab assistant	2	0	
6	Receipts	2	0	
Total coupling		14	7	21

Architecture Evaluation (Quantitatively)

After evaluation of **Hospital Management System**

Cohesion Metrics

4) **Cohesion Metric**
   CM(C)=7
5) **Cohesion Factor**
   F=NC(sys)+NO(sys)=7+22=29

I=

S.NO	No of components	No of Oper in a components	I=NC(c)+NO(c)	CohF(c)=CM(c)*(i^2-i)/(f^2-f)
1	Patient	5	6	0.2586
2	Doctor	3	4	0.10344
3	Nurse	1	2	0.01724
4	Pharmacy	3	4	0.10344
5	Lab assistant	3	4	0.10344
6	Receipts	2	3	0.05172

Coupling Metrics:

5) **Total Complexity Metrics**
6) **Complexity Factor**
7) **Total Complexity Metrics for a system**

S.NO	No of Component	Direct coupling	Indirect coupling	TCM(c)= (Tcoup(c) +NC(c)+ NO(c)) /CM(c)	CohF(c)= CM(c)*(i^2-i)/(f^2-f)	CoupF(c)= Tcoup(c)/ f^2-f	CompF= Coup(c)/Coh(c)	TCM(sys)= TCM(c)*compF(c)
1	Patient	4	1	1.5714	0.2586	0.00615	0.023781	0.3736
2	Doctor	1	1	0.8571	0.10344	0.00246	0.023781	0.02038

3	Nurse	1	1	0.57148	0.01724	0.00246	0.142691	0.08154
4	Pharmacy	2	0	0.4571	0.10344	0.00246	0.023781	0.01087
5	Lab assistant	2	0	0.8571	0.10344	0.00246	0.02378	0.01086
6	Receipts	2	0	0.71428	0.05172	0.00246	0.156488	0.11177
7	Payment	2	4	1.71428	0.2586	0.00738	0.2857	0.4897

➨Reusability Metrics

1. Reusability Metric

- Reusability can be calculated in terms of coupling, by counting the entire direct and indirect consumer calls for a specific provider.

Res = TCoup = DC(p) + IC(p)

2. Reusability Factor:

- The reusability factor can be calculated by comparing reusability values (in terms of coupling) with the system size by measuring the cohesion of operations.

$$ResF = \frac{CM(SyS)}{TCoup(SyS)}$$

Direct Coupling in a System	14
In-Direct Coupling in a System	7
Total Coupling in a System	21
Total Cohesion in a System	10
Coupling Factor in a System	0.0259
Cohesion Factor in a System	0.2512
Total Complexity in a System	399.6
Reusability Factor in a System	0.4762

➨Maintainability Metrics

In the literature, the most common characteristics related to maintainability are coupling, cohesion and complexity.

- **Cohesion**

  [Cohesion] positively impacts (+) → [maintainability]

- **Coupling**

  [Coupling] negatively impacts (-) → [maintainability]

- **No of complex component (NCC)**

  [NCC] negatively impacts (-) → [maintainability]

- **Complexity**

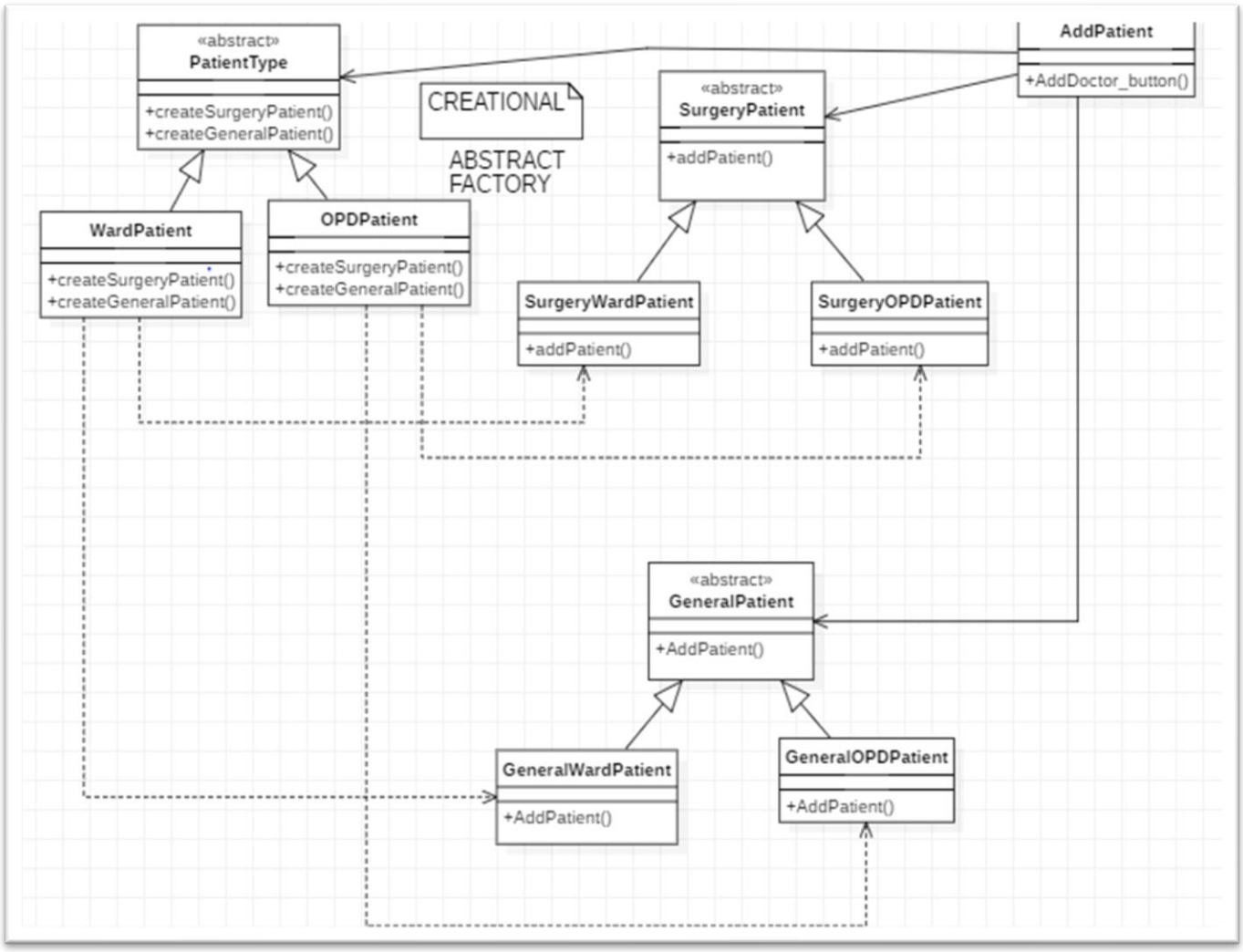
  [Complexity] negatively impacts (-) → [maintainability]

Maintainability Factors	Value	Upper Bound		Final Score
Cohesion (+ve) impacts (25%)	447.3684	100	111.841053	111.841053
Coupling (-ve) impacts (25%)	42.1053	100	10.52631579	14.47368421
No of Complex Services (-ve)(25%)	1	20	1.25	25.00
Complexity (-ve) impacts (25%)	416.7	300	34.72887731	-9.728877315
Total Maintainability Factor in a Sys				140.34

Phase-4: Detail Design

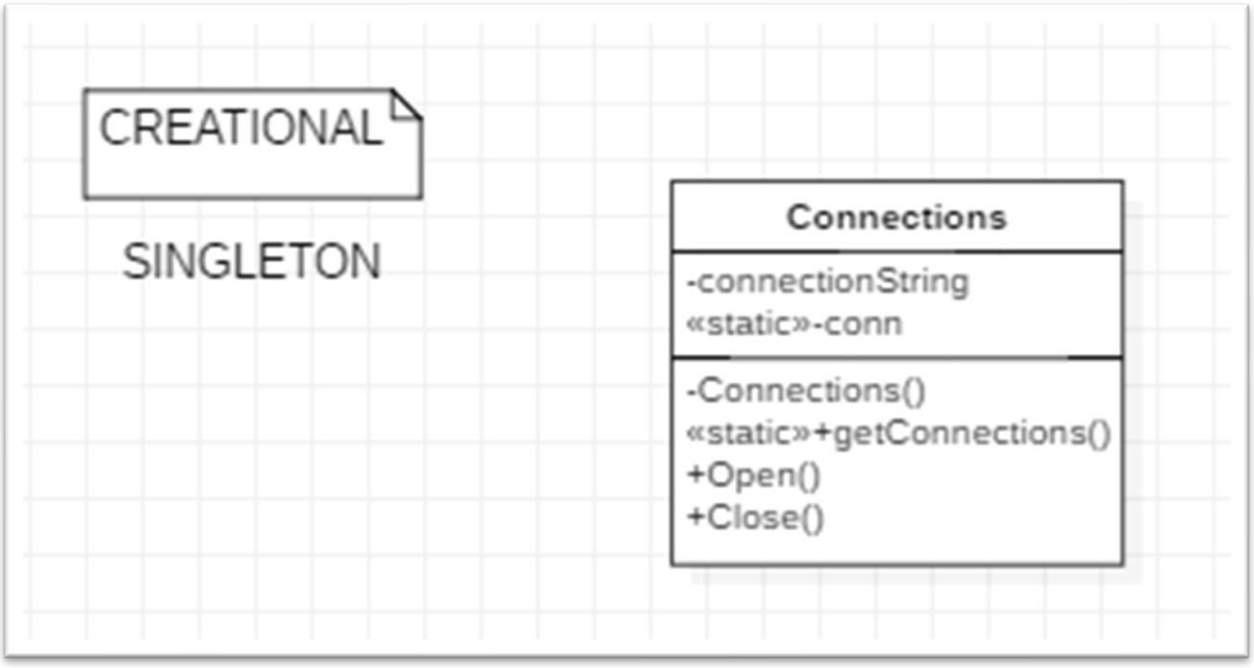
1. **Factory Design Pattern:**

We will have many classes and its objects. To create the objects of related families like Ward and OPD. We will create PatientType Class which will work as factory class and wardpatient, OPDpatient will work as concrete class. The picture is given below:



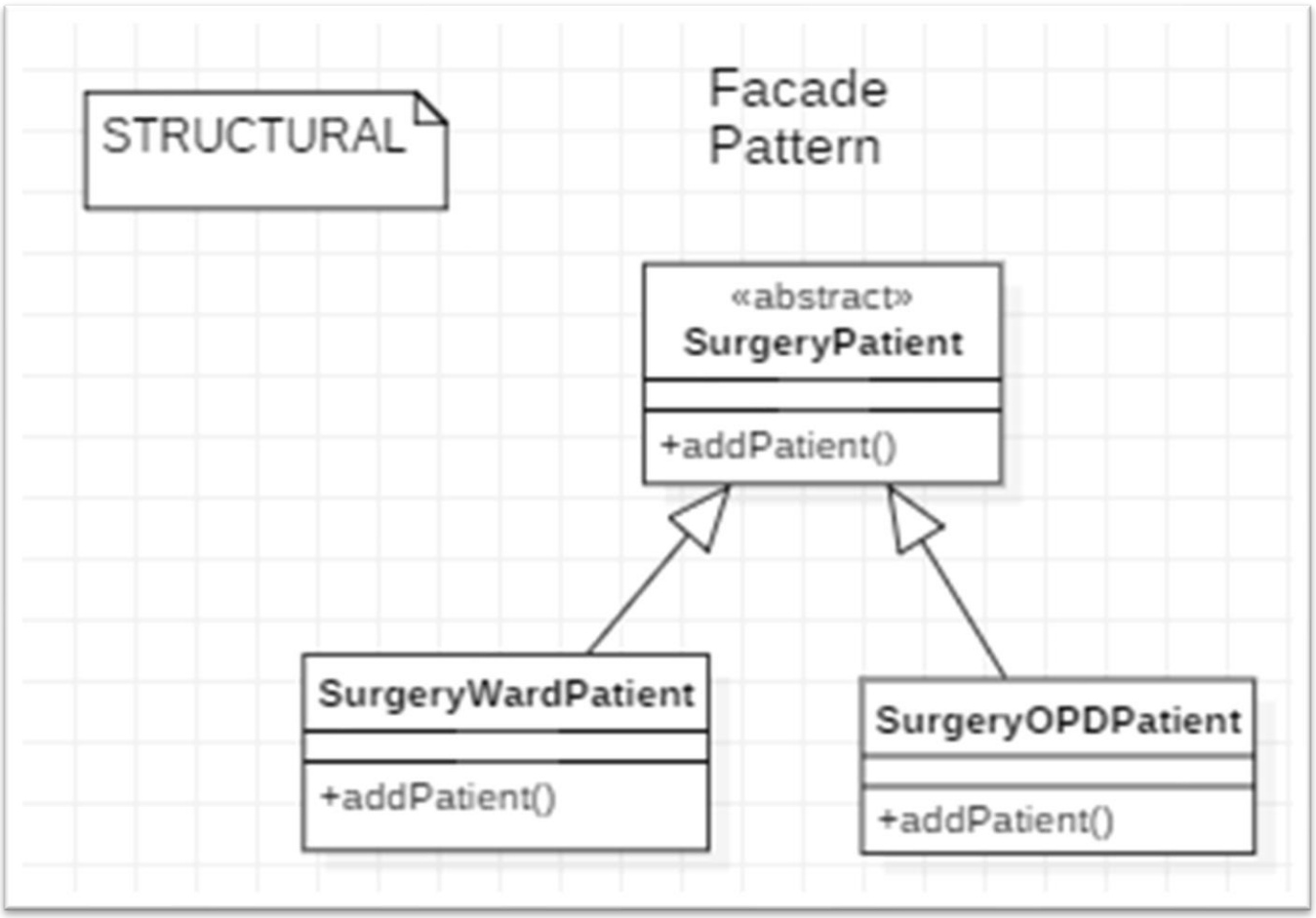
2. Singleton Design Pattern:

In this Design Pattern we will create single instance for a single class.



Structural Design Pattern

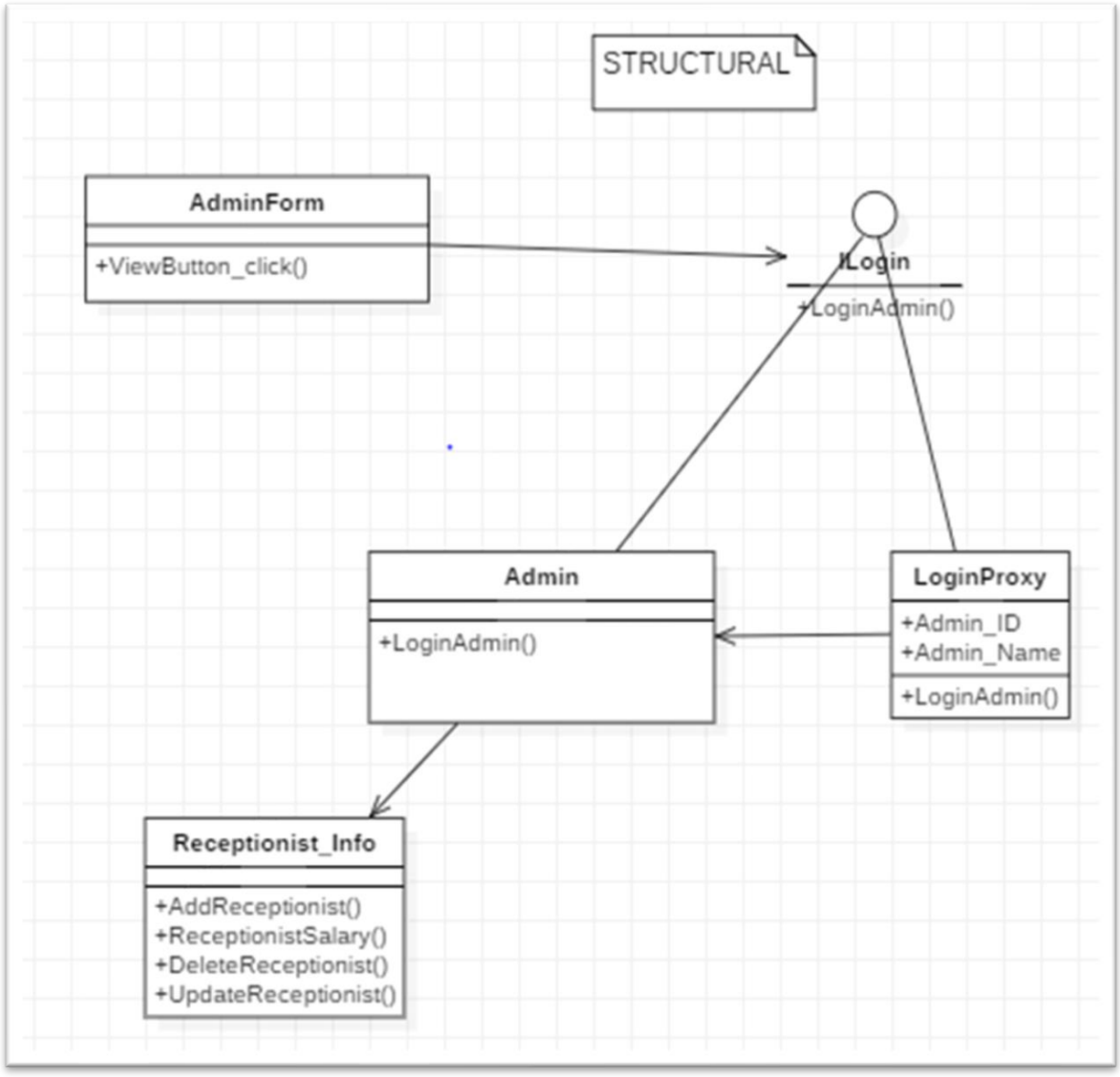
1. **Facade Pattern:** In this interface we will provide one interface for different types of Patients. For example SurgeryPatient class for both SurgeryWardPatients and SurgeryOPDPatients.



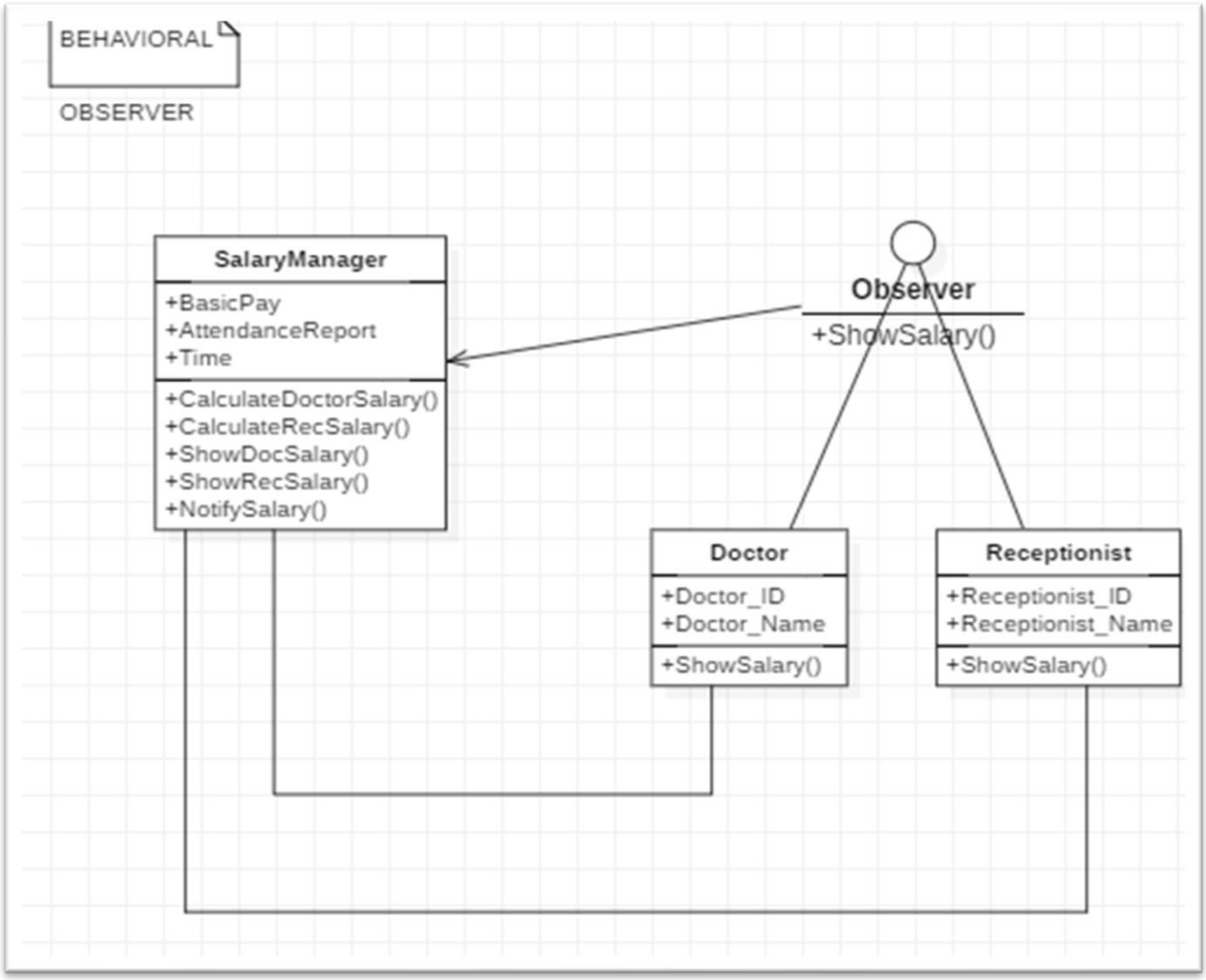
2. **<u>Proxy Pattern:</u>**

In this design pattern we will hide an interface from other classes except those who have the access to it.

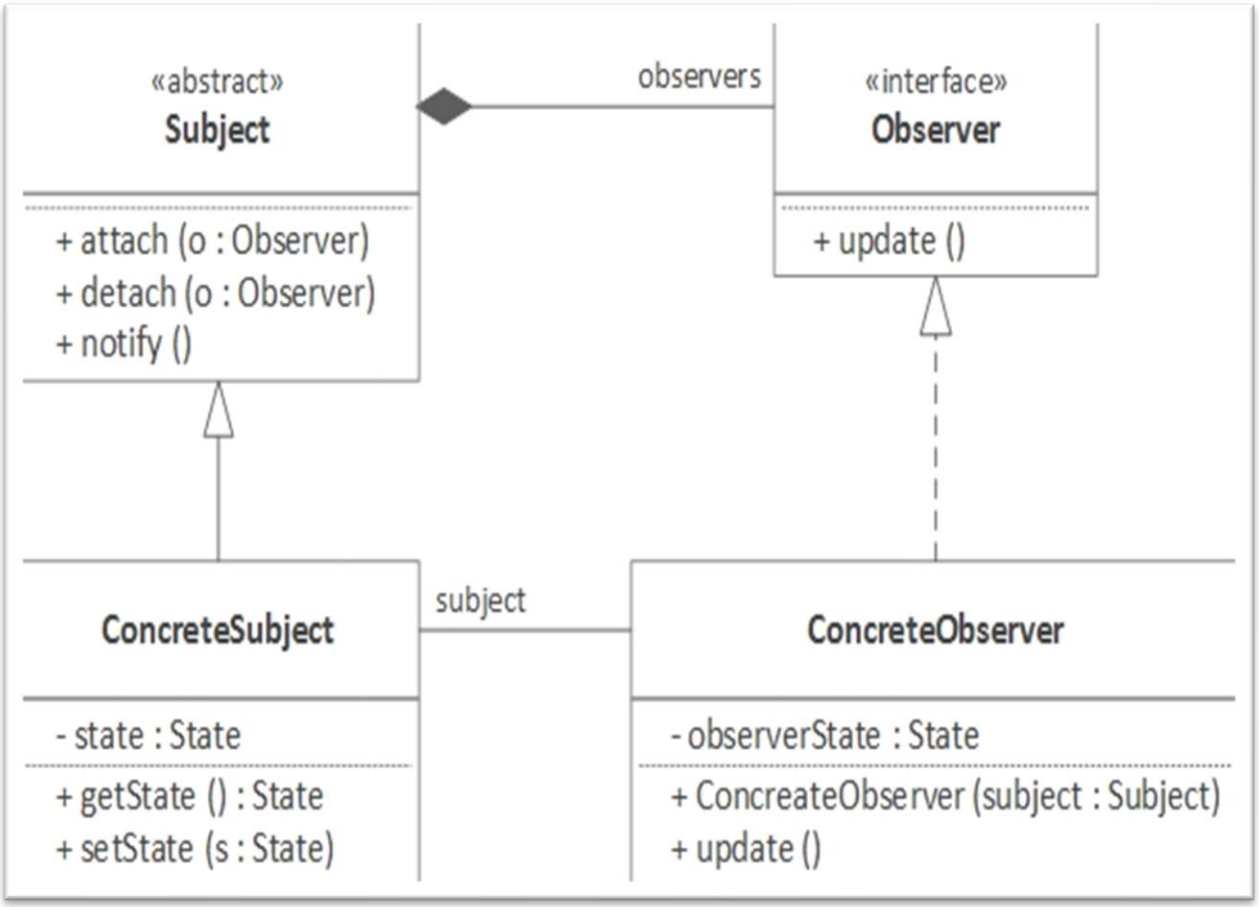
Like we will hide class receptionist from other classes except admin.



Behavioral Pattern



Strategy Design Pattern


«abstract»
Subject
observers
«interface»
Observer
+ attach (o : Observer)
+ detach (o : Observer)
+ notify ()
+ update ()
ConcreteSubject
subject
ConcreteObserver
- state : State
+ getState () : State
+ setState (s : State)
- observerState : State
+ ConcreateObserver (subject : Subject)
+ update ()