1. Name: Xiaotong Diao
2. Title: Healthcare Service
3. Project Description:A healthcare service website that patients can schedule appointments and pay bills, and doctors can send prescriptions to patients. We will set up for several patients and one doctor.
4. Features that were implemented:

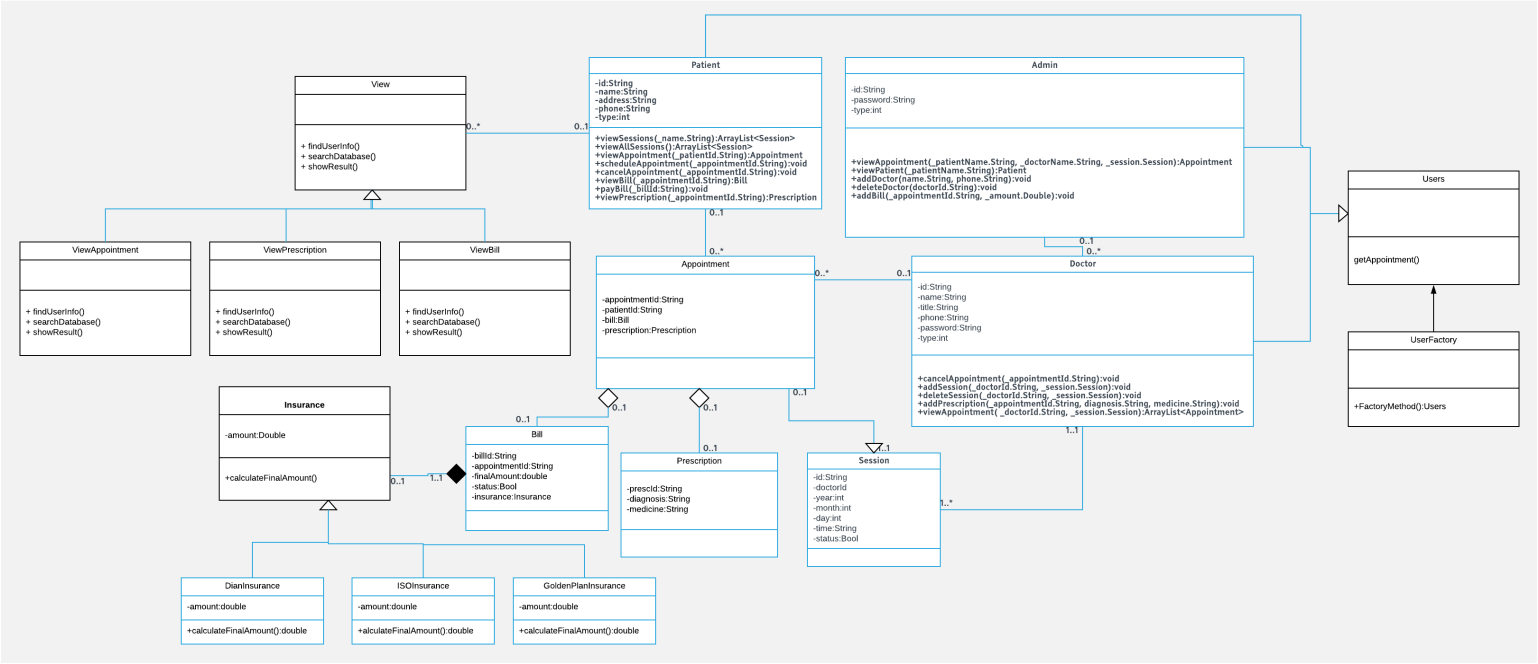
Requirement	ID
patients can signup	UR-01
patients, doctors, admin can log in	UR-02
Patients can search for available scheduling session by doctor's name	UR-03
Patients can view all available sessions	UR-04
Patients can schedule an appointment	UR-05
Patients can cancel an appointment	UR-06
Patients can view bill	UR-07
Patients can pay bill	UR-08
Patients can choose insurance	UR-09
Patients can view prescription form	UR-10
Admin can search a patient by name	UR-11
Admin can add a doctor	UR-12
Admin can delete a doctor	UR-13
Admin can view a patent's appointment	UR-14
Admin can add bill information	UR-15
Doctors can cancel appointment	UR-16
Doctors can add available session	UR-17
Doctors can delete available session	UR-18
Doctor can add prescription	UR-19

5. Features that were not implemented:

None

6. Class Diagram:

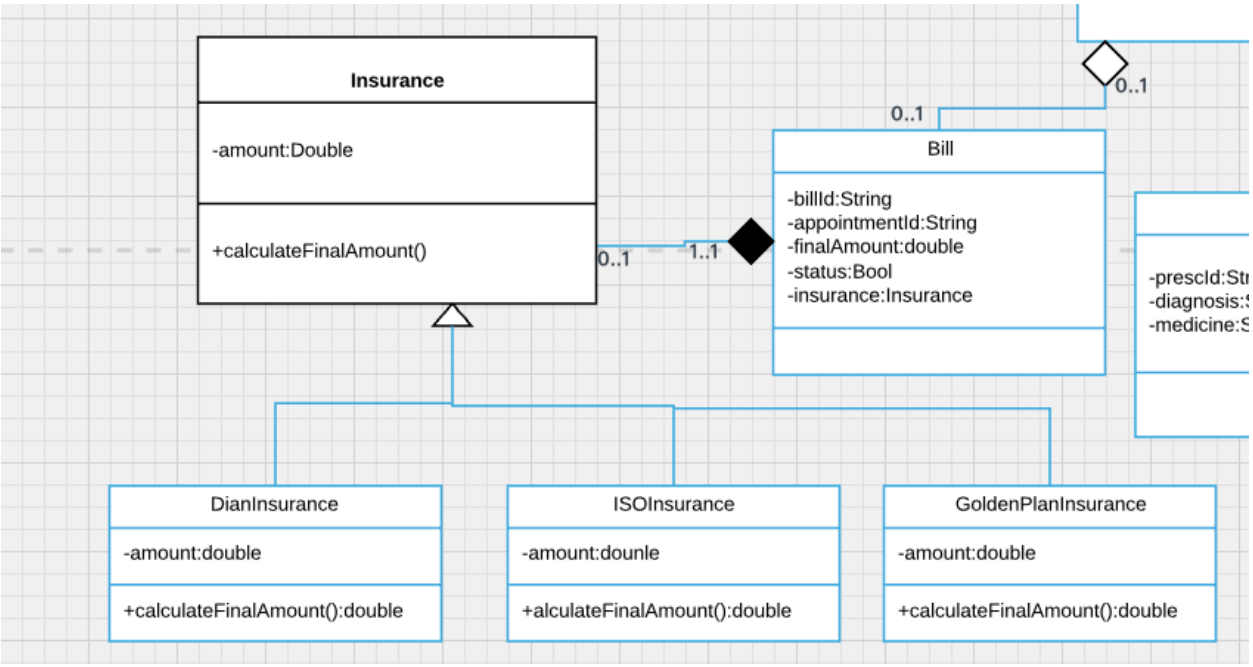
The mainly change in the class diagram is design patterns. I add 2 more design pattern to the class diagram: Template and Factory. To implement factory design pattern, I add 2 more classes(Users and UserFactory). To implement template, I add 4 more classes(View, ViewBill, ViewAppointment and ViewPrescription). So there are more classes and more design patterns in the final class diagram.



7. Design Patterns:

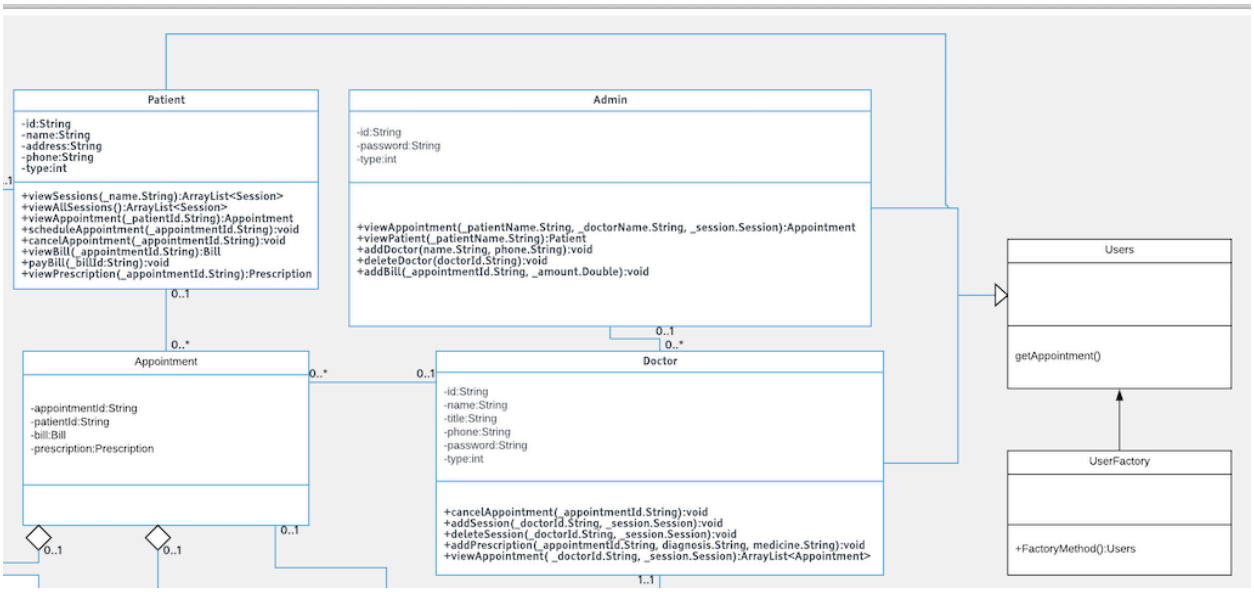
1.Strategy

When we compute the amount for each insurance, the algorithms are all different in reality. So I use Strategy design pattern to encapsulate the algorithm for this procedure. As the class diagram below shows, Insurance can be a composition of Bill, all different types of insurance including DianInsurance, ISOInsurance and GoldenPlanInsurance is inherited from the class Insurance. In this way, we could easily extend our insurance system without changing the client code. For example, if we want to add another insurance type called SafetyInsurance, we just need to add a class SafetyInsurance which extends Insurance and define the algorithm inside it. This exactly follows the close-open rule.



2. Factory

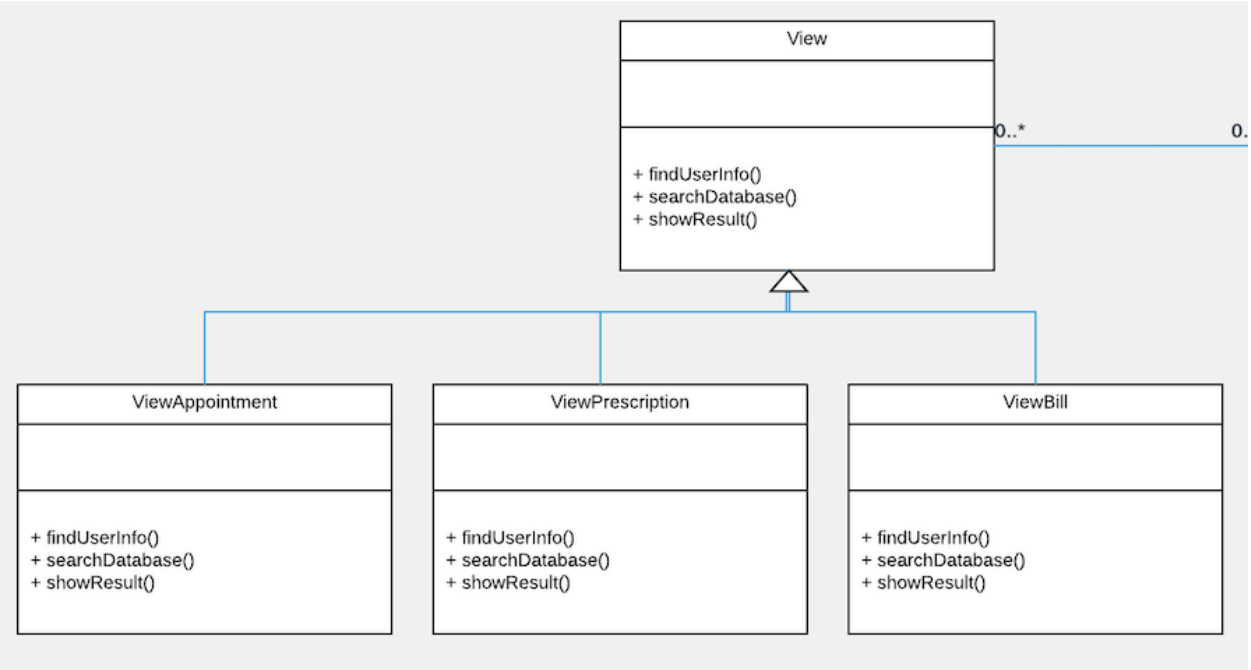
The Patient, Admin and Doctor classes are all child class of Users Class. Every time we initiate them, we need to tell the system what type of Users we create. In order to simplify our code, the Factory design pattern was used. I created a class called UserFactory to decide which type of Users to initiate by a method called FactoryMethod(). So we don't need to write a lot of if condition code and it also makes our system easier to extend.



3.Template

There are several method related to view information that have similar procedure to implement in the healthcare system, so I use Template design pattern to make my code more simple. Here I create a parent class called View which includes the steps that a view action should follow.

ViewAppointment, ViewPrescription and ViewBill are 3 classes that is inherited from View. They need to override the method in their parent class View to achieve their own goal.



8. What I Have Learned:

By implementing 3 design pattern in my project, I have a deeper understanding about design pattern and object-oriented analysis. Although sometimes we need to write more code and add more classes to implement a certain design pattern, this kind of skill can make the whole system more stable and easier to be extended. When I collaborate with other people, these code will be easier to modify. So, the project really let me learn a lot about what should I do in the industry and get a better coding habit.

Also, by creating a website with the Spring Boot frame, I learned how to do both front end developing and back end developing. Also I'm more familiar with database system. Those are skills that are really useful in my future career.