1. (a) What is UML? How can it help your class project?

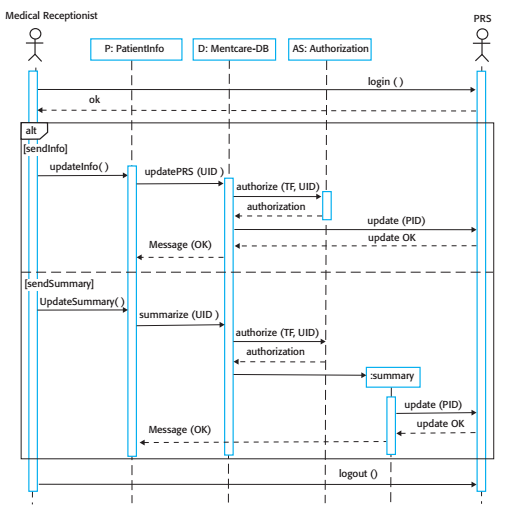
* UML이란, Unified Modeling Language의 줄임말로 소프트웨어 공학에서 사용되는 표준화된 범용 모델링 언어이다. UML은 객체 지향 소프트웨어 집약 시스템을 개발할 때 산출물을 명세화, 시각화, 문서화할 때 사용한다. UML을 사용하면 프로젝트의 전체적인 구성을 시각화 하여 한 눈에 파악할 수 있게 해주어 잘못 설계된 부분이나 모호하게 설계된 부분을 찾아 수정하는 데에 용이하고 결과적으로 더 완벽한 소프트웨어를 설계하기 위한 검증에 사용된다. 대부분 실제로 소프트웨어를 만드는 것보다 더 적은 비용이 들어 경제적인 효율이 높다.

1. Explain the following diagram (what it is, why important, how to draw)
   1. Use case diagram

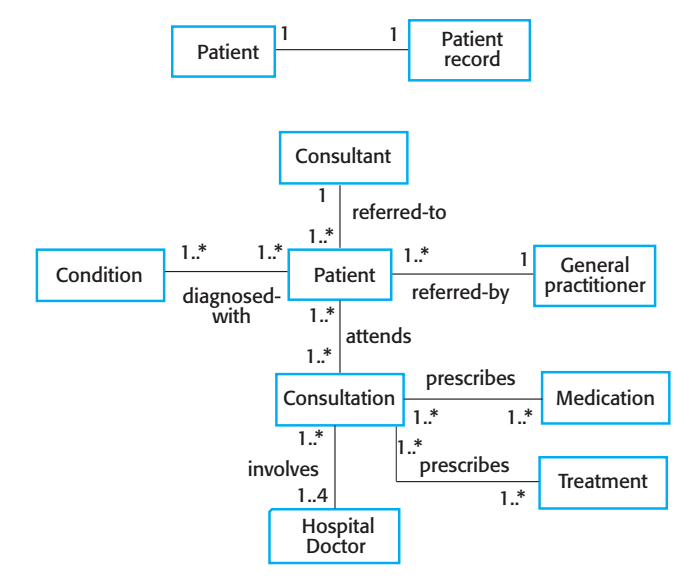
* Use case diagram은 유저와 시스템 간의 상호작용을 도식화한 그림이다. 설계하려는 시스템의 High Level View를 제공하여 실제로 그 시스템이 어떤 일을 하는지를 간단 명료하게 보여준다. 상호작용의 주체가 되는 User가 들어가고, 여러가지 원과 타원, 직선으로 구성된다.
  1. Sequence diagram
* Sequence Diagram은 시스템 내의 object들 간의 상호작용을 시간 순서에 따라 도식화한 그림이다. 기능의 종류에 따라 다양한 형태의 결과물이 나올 수 있다. 상단에 object들을 나열하고 좌우방향의 화살표를 통해 상호작용을 표현하고, 아래로 내려가면서 시간순서로 배치하게 된다. 어떤 function의 흐름을 표현하는 그림이다. 구성한 기능이 작동하는 흐름을 따라 그리면서 중간에 빠트린 부분이 없고 잘 작동하는지 점검 및 예측할 수 있다.
  1. Class diagram
* Class diagram은 클래스 내부의 정적인 내용이나 클래스 사이의 관계를 표기하는 다이어그램으로 시스템의 일부 또는 전체의 구조를 나타낼 수 있다. Class diagram은 의존관계(Dependencies)를 명확히 보게 해주며, 순환의존(Cycle, Circular Dependencies)이 발생하는 지점을 찾아내서 어떻게 이 순환 고리를 깨는 것이 가장 좋은 지 결정할 수 있게 해준다. 클래스와 스테레오 타입으로 구성되어 여러 개의 상자와 화살표로 그린다.
  1. State diagram
* State diagram은 프로그램의 상태와 서로간의 상태천이(천이 오타아님)를 표현하기 위해 도식화한 그림이다. 프로그램이 가질 수 있는 상태들을 노드로 표현하고 상태들 간에 발생할 수 있는 천이를 화살표로 표현하여 그 원인이 되는 기능을 적는다. State diagram을 통해 프로그램 실행이 진행하는 과정을 표현하고 코딩을 수행하면 오류를 줄일 수 있다.

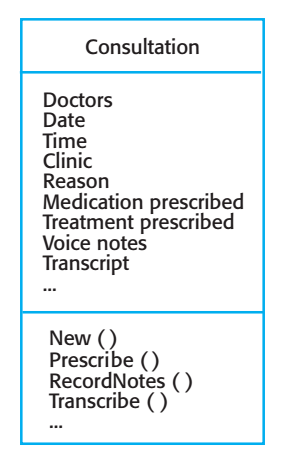
1. Draw sequence diagram and class diagram bsed on the following requirements

*“A receptionist may transfer data from the Mentcase system to a general patient record database that is maintained by a health authority. The information transferred may either be updated personal information (address, phone number, etc.) or a summary of the patient’s diagnosis and treatment.”*



<Sequence Diagram>



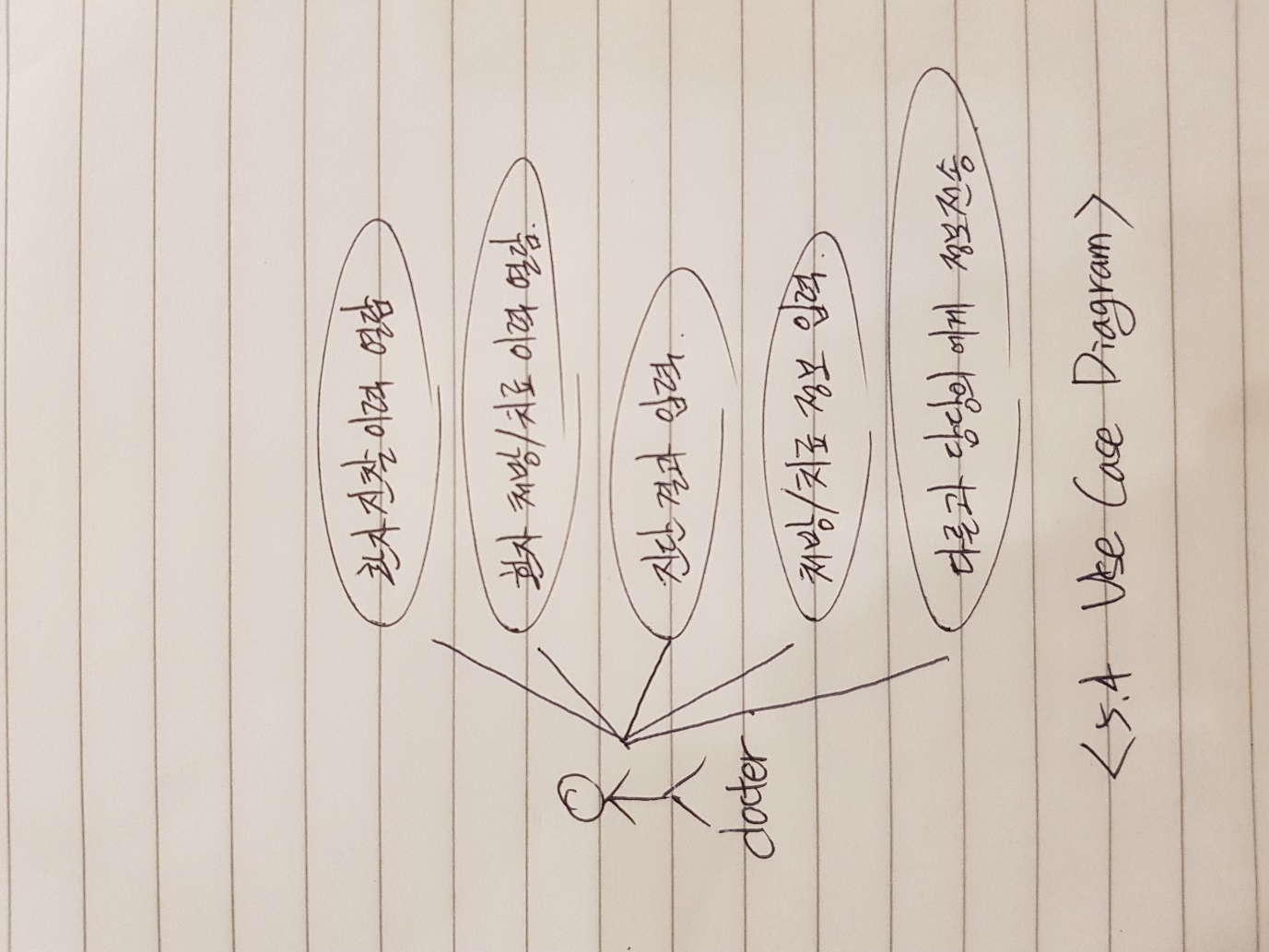


<위 : Class Diagram / 아래 : Consultation Class Description>

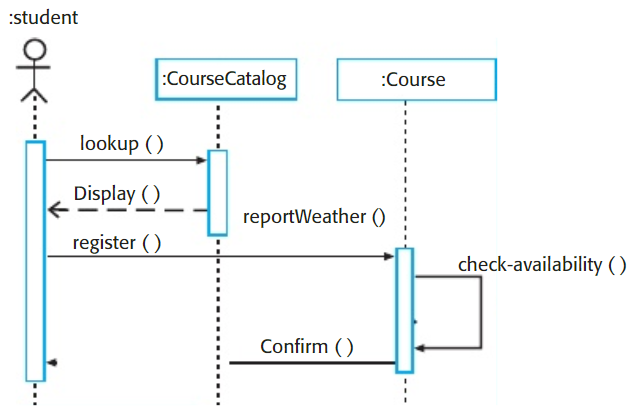
* 교과서에 이렇게 나와있는데, 이게 저 위에 그림 각각의 모든 클래스에 대해 아래가 하나씩 있어야 되는거에요. 근데 어차피 시간 모자라니까 예시로 하나만 적었다고 설명하시면 될 듯 합니다.

1. Textbook Exercises 5.4 ~ 5.8

5.4

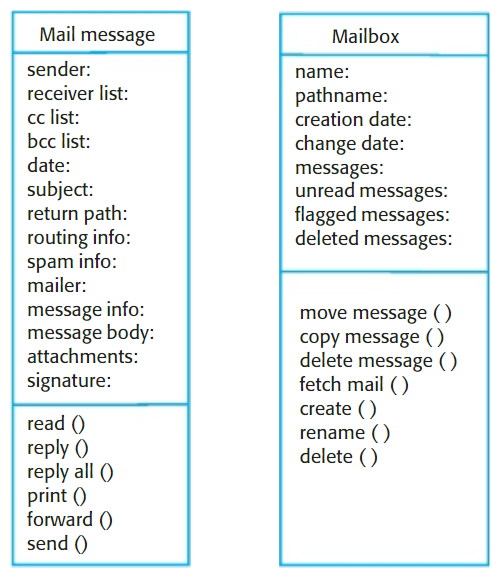


5.5



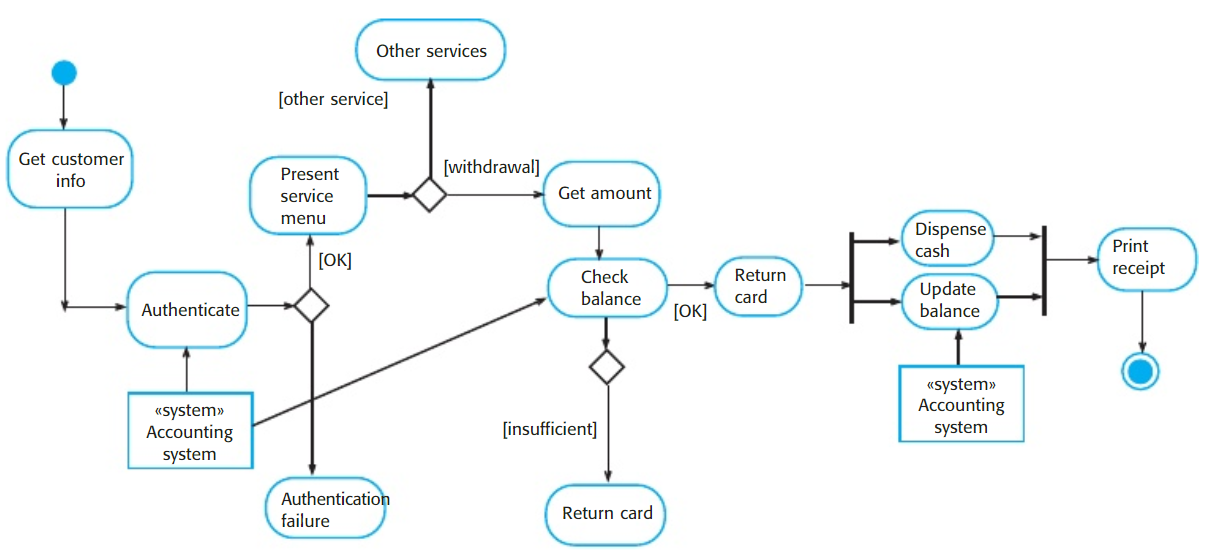
<5.5 Sequence Diagram>

5.6



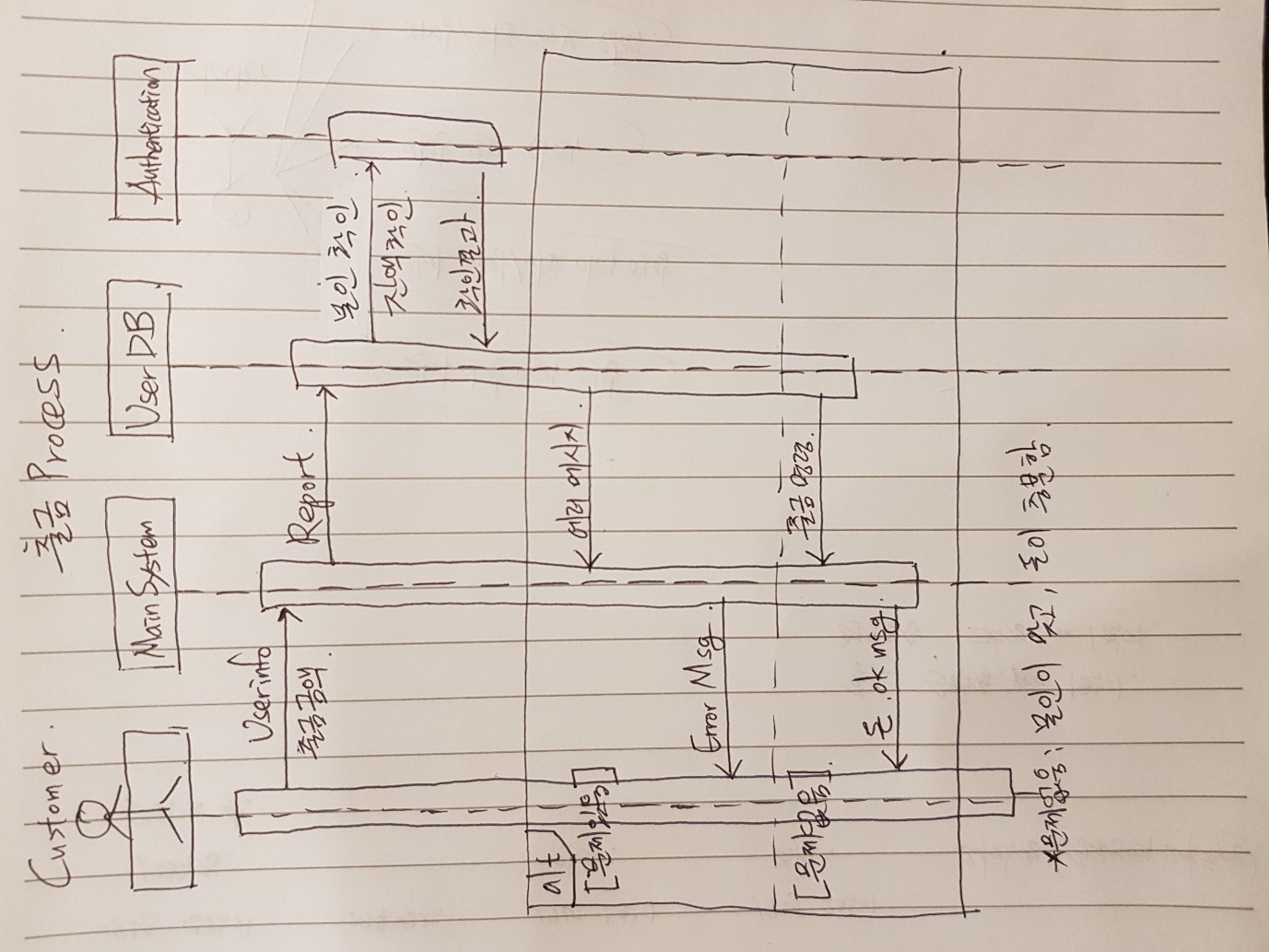
<5.6 Class Description>

5.7



<5.7 Activity Diagram>

5.8



<5.8 Sequence Diagram>

+왜 Activity Diagram과 Sequence Diagram을 둘 다 그리고 싶은 지 서술하랬음. 그리기 싫은데.

* Activity Diagram의 경우, 시스템의 전체적인 기능들의 배치와 구성을 알 수 있다. 어떤 분기점에서 다른 기능이 수행되는지, 사용자가 어느 시점에서 여러 개의 선택지를 마주하게 되는지 등 전체적인 구성을 파악하는데 용이하다. 반면에 Sequence Diagram은 하나의 기능에 대해 해당 기능이 어떤식으로 수행되고, Activity Diagram에서는 하나로 표현된 과정이 기능별로 어떻게 다르게 적용되어야 하는지 구체적으로 파악할 수 있다. 따라서 두 Diagram모두 다른 방식으로 프로세스의 완성도를 높이는 데 기여한다.