>>> System Admin

As a System Admin I want to:

- Login to the server, so I can control the system

- Have search box, so I can search records easily find patient, doctor or any record from server.

- View all appointments, so I can manage the appointments.

- Create doctors or secretary accounts, so I can manage all users' accounts in the system.

>>>Patient

As a Patient I want to:

- Be able to create account, so I can Login to clinic system.

- Login to system, so I can make or view appointments.

- Have personal login account, so I can manage my personal information.

- Make appointments, so I can Visit Clinic.

>>> Doctor

As a Doctor I want to:

- Login to system, so I can View my patient appointments.

- Have personal login account, so I can manage my personal information.

- View my patient information, so I can evaluate patient status.

- Add annotations to patient status, so I can make patient updated with medicine history.

>>> Secretary

As a secretary I want to:

- Login to system, so I can view all patients' appointments.

- Have personal login account, so I can manage my personal information.

- View doctor's appointments, so I can easily create new appointments for patients.

- Add comments to patient page, so I can make doctors have more information about patient.

2. Plan

2.1 Vision

For doctors and patients in all clinics, Who need to adjust their appointments and exports effectively, and do not want to waste time waiting for doctors to be ready for appointment, The Clinic Management is a Web and mobile based application, That booking and viewing the appointments in effective way, and provide a chat between the doctor and patient for extra information after the examination. Unlike other Clinic Management application, our product provides excellent services with cheap cost and high security.

2.2 Release Plan

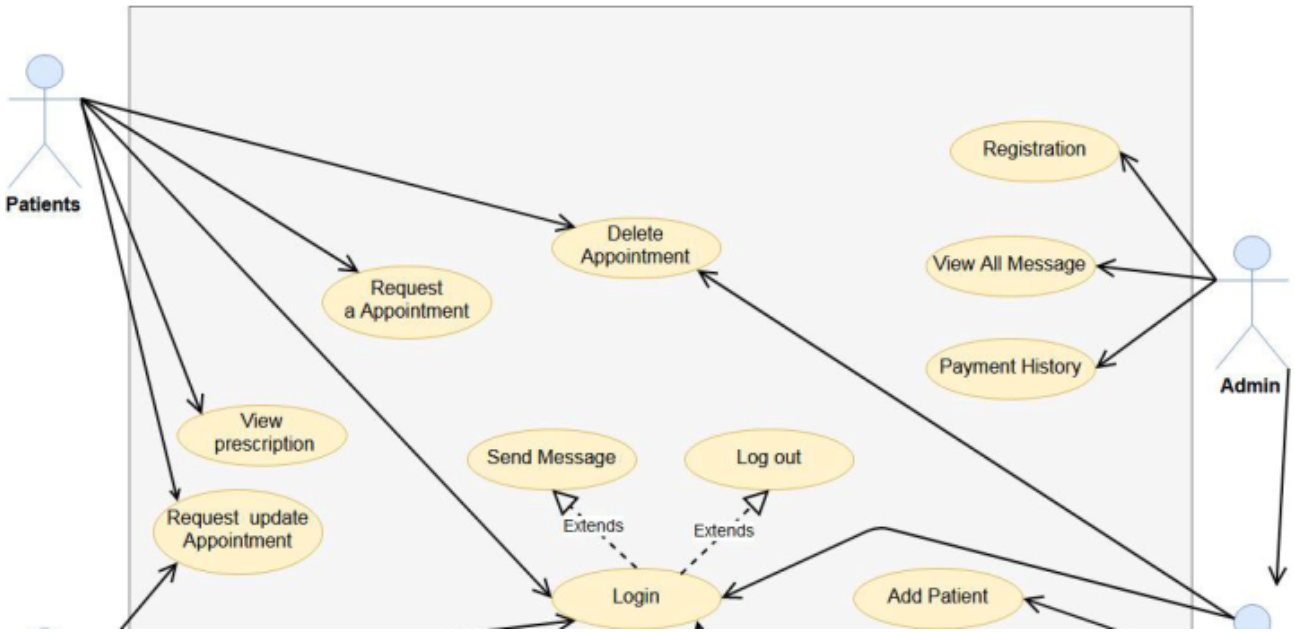
Iteration	Story	Estimated Time
Iteration 1	Login Operation	4-8 hours
Iteration 2	Create Accounts	8-12 hours
Iteration 3	View Appointments	4-6 hours
Iteration 4	Request Appointments	4-6 hours
Iteration 5	Chat System	12-16 hours

2.3 Iteration Plan

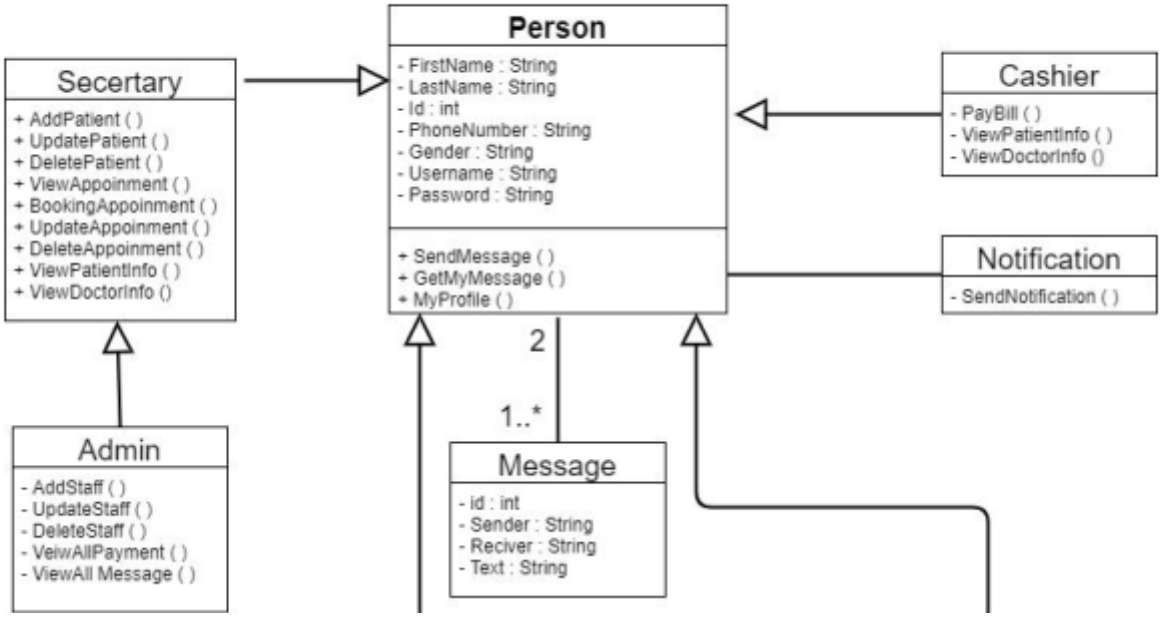
Task	Estimated time (weeks)
Make the fields and UI that user need for his request.	1
Making actions in client side for request	1
Make the part of API that user can request	2
Receive the request in web side and handle it.	2
Arrange table of appointments from various users	1

3. Design

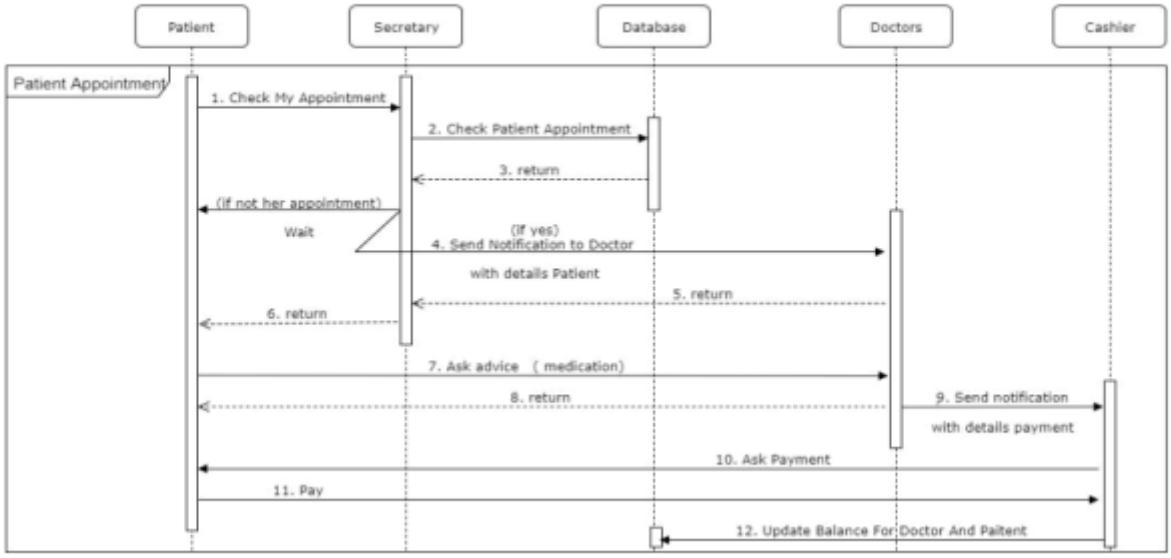
3.1 Use Case Diagram:



3.2 Class Diagram



3.3 Sequence diagram



4. Code (Code Samples -- Method from Doctor Class

```
public class Doctor {

    public ArrayList<Medicine> writeMedicine(Patient patient) {

        ArrayList<Medicine> medicine = new ArrayList<>();
        medicine.add(new Medicine("Utrogestan 100mg", 10));
        medicine.add(new Medicine("Diovan 500mg", 10));

        return medicine;
    }
}
```

- Method from Main Class:

```
public static void writeMedicalPrescription(Doctor doctor, Patient patient){
    Date dateOfWritingPrescription = new Date();
    String datePattern = "dd-MM-yyyy";
    String timePattern = "HH:MM";

    SimpleDateFormat simpleDateFormat = new SimpleDateFormat(datePattern);
    SimpleDateFormat simpleDateFormat1 = new SimpleDateFormat(timePattern);

    System.out.println(simpleDateFormat.format(dateOfWritingPrescription));
    System.out.println(simpleDateFormat1.format(dateOfWritingPrescription));

    ArrayList<Medicine> medicine = doctor.writeMedicine(patient);

    patient.getPrescriptionAsSoftcopy(medicine, patient.getName());

}
```

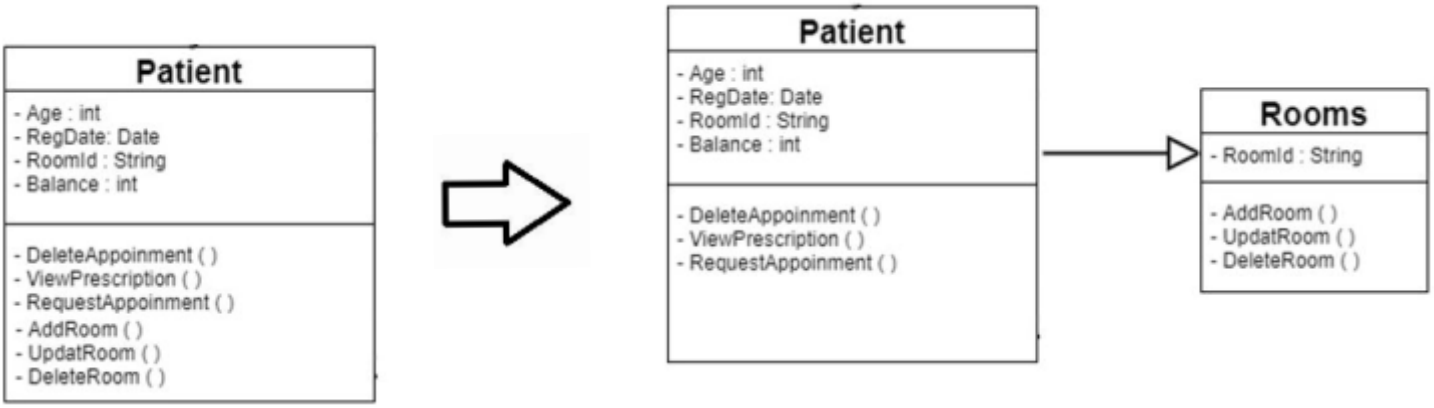
4.2 Code Refectories Example #1: Extract Method

```
public static void writeMedicalPrescription(Doctor doctor, Patient patient){
    Date dateOfWritingPrescription = new Date();
    String datePattern = "dd-MM-yyyy";
    String timePattern = "HH:MM";

    SimpleDateFormat simpleDateFormat = new SimpleDateFormat(datePattern);
    SimpleDateFormat simpleDateFormat1 = new SimpleDateFormat(timePattern);

    System.out.println(simpleDateFormat.format(dateOfWritingPrescription));
    System.out.println(simpleDateFormat1.format(dateOfWritingPrescription));

    ArrayList<Medicine> medicine = doctor.writeMedicine(patient);

    patient.getPrescriptionAsSoftcopy(medicine, patient.getName());
}
```

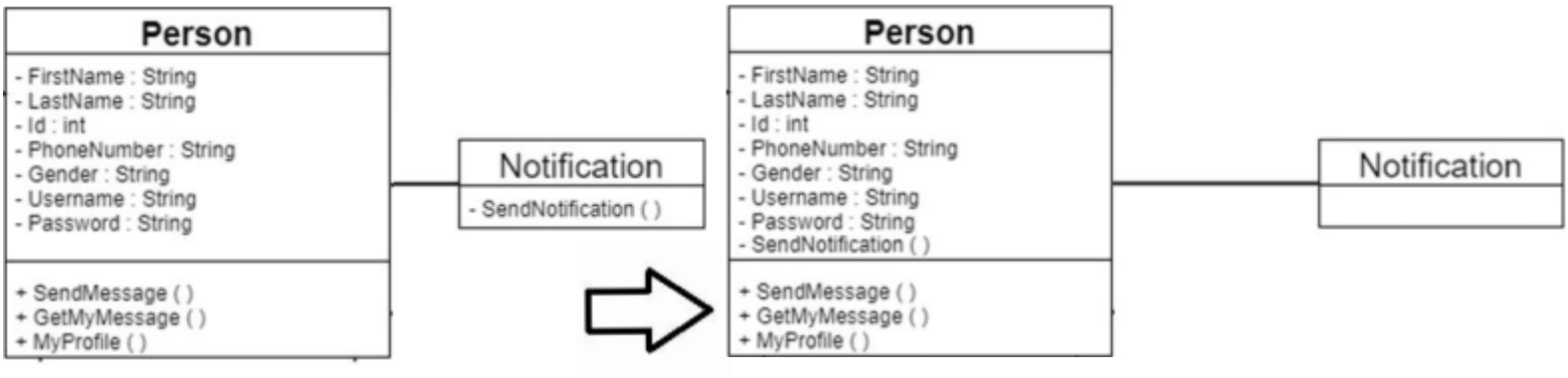
It will be :

```
public static void writeMedicalPrescription(Doctor doctor, Patient patient){
    ArrayList<Medicine> medicine = doctor.writeMedicine(patient);
    patient.getPrescriptionAsSoftcopy(medicine, patient.getName());
}

public static void printTimeAndDate() {
  Date dateOfWritingPrescription = new Date();
  String timePattern = "dd-MM-yyyy";
  String datePattern = "HH:MM";

  SimpleDateFormat simpleDateFormat1 = new SimpleDateFormat(timePattern);
  SimpleDateFormat simpleDateFormat2 = new SimpleDateFormat(datePattern);

  System.out.println(simpleDateFormat1.format(dateOfWritingPrescription));
  System.out.println(simpleDateFormat2.format(dateOfWritingPrescription));
}
```

Example #3: Extract class



Example #4: Move Method



5 Unit Test with JUnit

```
public class Test {

  private Doctor jone;
  private Patient alice;
  private DbConnection db;

  @Before
  public void setup() {
    Doctor jone = new Doctor();
    Patient alice = new Patient();
    db.connect();
    assertNotNull(db);
    db.save(jone);
    db.save(alice);
  }

  @Test
  public void test0() {
    assertNotNull(jone);
    assertNotNull(alice);
  }
```