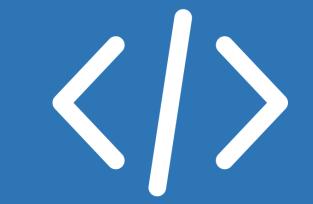
한의원관리자

제작기간:: 23.01.31-23.02.04

제작:이 와이



上大

- 1. 기획의도
- 2. 소프트웨어 구성
- 3. 목업 디자인



7 2 9 5

DB수업에 대한 응용을 위해, 사용자의 접근이 익숙한 한의원의 환자 관리 차트를 제작하였습니다.

소프트웨어구성



TA TO <DATA>

```
@Entity
@Getter
@NoArgsConstructor(access = AccessLevel.PROTECTED)
public class PrescribeHistory {
   @Id
   @GeneratedValue(strategy = GenerationType.IDENTITY)
   private Long id;
   @ManyToOne(fetch = FetchType.LAZY)
   @JoinColumn(name = "hospitalPatientId", nullable = false)
   private HospitalPatient hospitalPatient;
   @Column(nullable = false, length = 20)
   @Enumerated(value = EnumType.STRING)
   private TreatmentList treatmentList;
   @Column(nullable = false)
   private Integer price;
   @Column(nullable = false)
   private Boolean isInsuranced;
   @Column(nullable = false)
   private LocalTime treatmentTime;
   @Column(nullable = false)
   private LocalDate treatmentDate;
   @Column(nullable = false)
   private Boolean isPaid ;
```

```
@Entity
@Getter
@NoArgsConstructor(access = AccessLevel.PROTECTED)
public class HospitalPatient {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;
    @Column(nullable = false,length = 15)
    private String patientName;
    @Column(nullable = false, length = 14)
    private String idNumber;
    @Column(nullable = false,length = 13)
    private String phoneNum;
    @Column(nullable = false)
    private String address;
    @Column(nullable = false , columnDefinition = "TEXT")
    private String purpose;
    @Column(nullable = false)
    private LocalDate enrollDate;
```

了人 < service >

.bulid() 를 이용하여 for문을 이용하지 않고 ———— HospitalPatientBuilder를 통해 값을 받아오도록 함

```
@Service
@RequiredArgsConstructor
public class HospitalPatientService {
    private final HospitalPatientRepository hospitalPatientRepository;

    public void setHospitalPatient(HospitalPatientRequest request) {
        HospitalPatient addData = new HospitalPatient.HospitalPatientBuilder(request).build();
        hospitalPatientRepository.save(addData);
    }

    public HospitalPatient getData(long id) {
        return hospitalPatientRepository.findById(id).orElseThrow();
    }
}
```

```
public HospitalPatientBuilder(HospitalPatientRequest request) {
    this.patientName = request.getPatientName();
    this.idNumber = request.getIdNumber();
    this.phoneNum = request.getPhoneNum();
    this.address = request.getAddress();
    this.purpose = request.getPurpose();
    this.enrollDate = LocalDate.now();
}

@Override
public HospitalPatient build() {
    return new HospitalPatient(this);
}
```

```
private HospitalPatient(HospitalPatientBuilder builder) {
    this.patientName = builder.patientName;
    this.idNumber = builder.idNumber;
    this.phoneNum = builder.phoneNum;
    this.address = builder.address;
    this.purpose = builder.purpose;
    this.enrollDate = builder.enrollDate;
}
```

HospitalPatientBuilder

了人 < service >

setPrescribeHistroy도 비슷한 방법으로 build()를 적용한다

```
@Service
@RequiredArgsConstructor
public class PrescribeHistoryService {
    private final PrescribeHistoryRepository prescribeHistoryRepository;
    public void setPrescribeHistory(Hosnital Patient hosnital Patient HistoryRequest request) {
        PrescribeHistory addData = new PrescribeHistory.PrescribeHistoryBuilder(hospitalPatient,request).build();
        prescribeHistoryRepository.save(addData);
    }
}
```

```
public PrescribeHistoryBuilder(HospitalPatient hospitalPatient, HistoryRequest request) {
    this.hospitalPatient = hospitalPatient;
    this.treatmentList = request.getTreatmentList();
    this.price = request.getIsInsuranced() ? request.getTreatmentList().getInsurancedPrice():

re uest.getTreatmentList().getNonInsurancedPrice();
    this.isInsurance = request.getIsInsuranced();
    this.isInsurance = request.getIsInsuranced();
    this.treatmentDate = LocalDate.now();
    this.treatmentTime = LocalTime.now();
    this.isPaid = false;
}

@Override
public PrescribeHistory build() {
    return new PrescribeHistory(this);
}
}
```

```
public HistoryItemBuilder (PrescribeHistory origindata){
    this.historyId = origindata.getId();
    this.patientId = origindata.getHospitalPatient().getId();
    this.patientName = origindata.getHospitalPatient().getPatientName();
    this.phoneNum = origindata.getHospitalPatient().getPhoneNum();
    this.idNumber = origindata.getHospitalPatient().getIdNumber();
    this.treatmentName = origindata.getTreatmentList().getTreatmentName();
    this.price = origindata.getPrice();
    this.isInsuranced = origindata.getIsInsuranced()? "OH" : "OFLY";
    this.treatmentDate = origindata.getTreatmentDate();
    this.treatmentTime = origindata.getTreatmentTime();
    this.isPaid = origindata.getIsPaid()? "OH" : "OFLY";
}
```

목업 디자인

고객관리 버튼을 누르고, 이동하여 DB를 등록할 수 있도록 적용함



고객리스트

고객등록 메인으로

ld	이름	주민등록번호	주소	전화번호	방문사유	비고
1	흥길동	97***-*****	경기도 ******	010-***-***	물리치료	
2	이길동	86***-****	경기도 ******	010-***-***	교통사고	자동차보험 처리
3	강길동	58***-****	서울특별시 *****	010-***-***	물리치료/부항	
4	박길동	00****-*****	경기도*****	010-***-***	물리치료	
5	한길동	88***-****	서울특별시 ******	010-***-***	추나치료	