

# 한의원 관리자

제작기간 :: 23.01.31-23.02.04

제작 : 이 와 이



# 목차

1. 기획의도
2. 소프트웨어 구성
3. 목업 디자인



# 기획의도

“

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

”

# 소프트웨어 구성



```
graph LR; DATA((DATA)) --- SERVICE((SERVICE)); SERVICE --- CONTROLLER((CONTROLLER));
```

**DATA**

**SERVICE**

**CONTROLLER**

# 구성

## <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 LocalDateTime 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;
}
```

# 구성

## < 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);
}
```

HospitalPatientBuilder

```
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;
}
```

# 구성

## < service >

setPrescribeHistory도  
비슷한 방법으로 build()를 적용한다

```
@Service
@RequiredArgsConstructor
public class PrescribeHistoryService {
    private final PrescribeHistoryRepository prescribeHistoryRepository;
    public void setPrescribeHistory(HospitalPatient hospitalPatient, 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.getIsInsured() ? request.getTreatmentList().getInsuredPrice() :
request.getTreatmentList().getNonInsuredPrice();
    this.isInsurance = request.getIsInsured();
    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.isInsured = origindata.getIsInsured() ? "예" : "아니오";
    this.treatmentDate = origindata.getTreatmentDate();
    this.treatmentTime = origindata.getTreatmentTime();
    this.isPaid = origindata.getIsPaid() ? "예" : "아니오";
}
```

# 목업 디자인

고객관리 버튼을 누르고,  
이동하여 DB를 등록할 수 있도록 적용함

