# 응급상황대응프로젝트

Java 웹 풀스택 웹 개발 미니 프로젝트

### 목차



요구사항 정의서



기능 구조



코드 구현

조원 소개 및 업무 분담



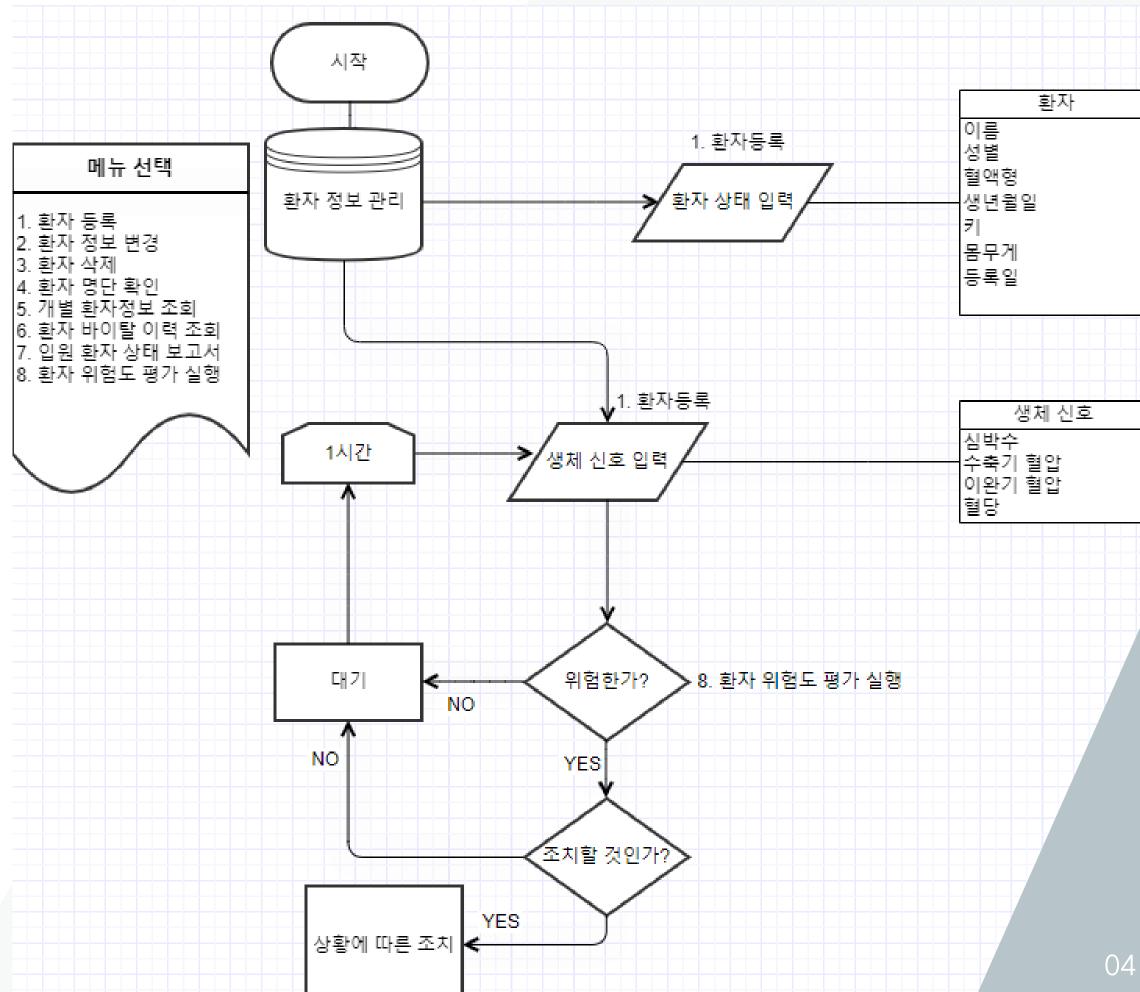
Q&A

## 요구사항정의서

기능	세부 기능	내용	필수 입력
환자 정보 관리	환자 등록	- 새로운 환자의 이름, 생년월일, 혈액형, 생일, 키, 몸무게, 등록일 정보 입력 및 기록	이름, 생년월일, 혈액형, 생일, 키, 몸무게
	환자 삭제	- 등록된 환자 정보를 이름으로 검색하여 삭제	OI름
	환자 정보 변경	- 등록된 환자 정보를 이름으로 검색하여 정보 변경	이름, 변경할 정보 선택
	환자 명단 조회	- 현재 입원 환자의 명단과 번호 출력	-
	환자 개인별 생체정보 이력 확인	- 환자 이름으로 검색하여 생체정보 이력과 위험도 상태, 시간 정보 출력	이름
	환자 위험도 관리 보고서 출력	- 입원 환자의 이름, 위험 상태 출력 - 총 인원, 안전/위험/매우 위험/비상 인원 수 및 명단 출력 - 미조치 위험/매우 위험 인원 수 및 명단 출력 - 콘솔 출력 및 "요청한 시간report.txt" 파일 저장	_
환자 위험 상태 관리	생체 정보 측정값 입력	- <mark>매시 정각마다 모든 환자의 생체 정보값(심박수, 혈압, 혈당)</mark> 입력 메시지 출력 - 즉시 입력 불가 시 5/10/30분 후 입력 설정 가능	심박수, 혈압, 혈당
	생체 정보 측정값 분석, 판단	- 입력받은 환자의 생체 정보값을 분석하여 위험도 상태 설정	_
	환자 위험도에 따른 조치상황 확인	- 위험/매우 위험 상태일 경우 의사 조치 여부 확인 및 권고 - 미조치 시 보고서 출력	조치 여부 판단 (Y/N)
프로그램 데이터 관리	저장된 정보 불러오기	- 환자 정보 전체 파일과 바이탈 기록 파일에서 정보 불러오기	-
	프로그램 종료 시 정보 기록	- 환자 정보 전체 파일과 각 환자의 바이탈 기록 파일 저장	-







### C Cancelable Scanner

- △ subCallable: Callable<String>
- △ futureTask: FutureTask<String>
- o readLine(): String
- o cancel(): void

### C CancelThread

- △ cancelableScanner: CancelableScanner
- o run(): void
- CancelThread(cancelableScanner: CancelableScanner)

### C HopitalManagemet <T>

- △ patients: List<T>
- △ hospitalName: String
- HospitalManagement(hospitalName: String): void
- saveWholeFile(): int
- o saveInfoFile(patient: T): int
- o saveVitalInfoFile(patient: T, vitalinfo: VitalInfo): int
- o deleteInfo(patient: T): int
- deleteInfoFile(path: String): int
- o readInfoFile(path: String): T
- o readVitalInfoFile(path: String): VitalInfo
- search(name: String): Patient
- initHospital(): int
- registerPatient(): int
- modifyPatient(): void
- deletePatient(): void
- o vitalCheck(): int
- patientList(): void
- evaluatePatientStatus(): int
- o notifyToDoctor(): void
- o patientReport(): void
- hospitalReport(): void

### C Patient

- □ name:String
- □ gender: String
- □ bloodType: String
- □ birthDay: String
  □ height: int
- □ weight: int
- □ registerdDate: String
- □ isNotified: boolean
- o ststus: PatientStatus
- o vitalInfo: ArrayList<VitalInfo>
- o setRegisterdDate(re
- △ Patient()
- o equals(obj: Object): boolean
- getName(): String
- o setName(name: String): void
- o getGender(): String
- o setGender(gender: String): void
- getBloodType(): String
- getBloodType(): String
- getBirthDay(): String
- o setBirthDay(birthDay: String): void
- getHeight(): int
- setHeight(height: int): void
- getWeight(): int
- setWeight(weight: int): void
- o getRegisterdDate(): String
- setRegisterdDate(registerdDate: Date): void
- isNotified(): boolean
- o setNotified(isNotified: boolean): void
- getStatus(): PatientStatus
- o setStatus(status: PatientStatus): void
- Patient(name: String, gender: String, bloodType: String, birthDay: String, height: int, weight: int)

### C VitalInfo

- BPM: int
- □ SBP: int
- □ DBP: int
- □ bloodSugar: int
- checkTime: String
- getCheckTime(): String
- setCheckTime(checkTime: String): void
- VitalInfo(bpm: int, sbp: int, dbp: int, bloodSugar: int)
- getBpm(): int
- o setBpm(bpm: int): void
- getSbp(): int
- setSbp(sbp: int): void
- getDbp(): int
- o setDbp(dbp: int): void
- o getBloodSugar(): int
- setBloodSugar(bloodSugar: int): void

### C Main

△ main(args: String[]): void

### E Patient Status

Safe: PatientStatus
Dangerous: PatientStatus
Very\_Dangerous: PatientStatus

### C LoopThread

- △ hospital: HospitalManagement
- △ LoopThread(hospital: HospitalManagement<Patient>)
- o printMenu(); void
- o run(): void

### C VitalThread

- △ vitalCheck: int
- △ evaluate: int
- △ loop: boolean
- △ hospital: HospitalManagement<Patient>
- △ scanner: Scanner
- VitalThread(hospital: HospitalManagement<Patient>)
- o run(): void

**Patient** 

- 환자의 개인 정보와 생체 정보, 위험 상태 등을 관리
- 환자의 위험 상태를 저장하는 속성을 갖고 있음
- 의사 조치 여부 속성을 가지고 있음
- LOOPTHREAD
   1시간마다 반복하게 만드는 CLASS.

VITAL THREAD 객체 생성, 실행

VITALTHREAD
 바이탈 정보 입력 및 상태 판단

Hospital

Management

- 환자 관리 기능 제공
- 생체 정보 입력, 분석, 판단 기능 제공
- 위험도 관리 보고서 출력 기능 제공
- 프로그램 실행 / 종료 시 데이터 관리 기능 제공

Thread

• BPM : 심박수

● SBP : 이완기 혈압

● DBP : 수축기 혈압

• bloogsugar : 혈당

Patient

• 환자의 개인 정보와 생체 정보, 위험 상태 등을 관리

- 환자의 위험 상태를 저장하는 속성을 갖고 있음
- 의사 조치 여부 속성을 가지고 있
- 개별 생체 정보를 저장하는 역할
- 현상이 발생한 원인에 대한 여러가지 가설을 제시해 보세요. 가능한 원인을 명시하고 각각에 대한 근거 또한 제시해보세요.

Hospital Management

- 환자 관리 기능 제공
- 생체 정보 입력, 분석, 판단 기능 제공
- 위험도 관리 보고서 출력 기능 제공
- 프로그램 실행 / 종료 시 데이터 관리 기능 제공
- LOOPTHREAD
   1시간마다 반복하게 만드는 CLASS.
   VITAL THREAD 객체 생성, 실행
- VITALTHREAD
   바이탈 정보 입력 및 상태 판단

항목	생체 정보	기호	범위	비고
SAFE	심박수		80	
	수축기 혈압		140	早干
	이완기 혈압		90	
	혈당		120	
DENGEROUS	심박수		80	
	수축기 혈압		140	17世 〇   人
	이완기 혈압		90	1개 이상
	혈당		120	
VERY DENG ROUS	심박수		80	
	수축기 혈압		140	2개 이사
	이완기 혈압		90	3개 이상
	혈당		120	

Vitalinfo )

MENU STRUCUTRE

### 코드구현



## 코드는 이클립스를 통해 보여드리겠습니다.

## 조원 소개 및 업무 분담

### 이성연

-리더 -기술 총괄

### 김승희

-환자 위험도 분류

### 김현석

-기획서 작성 -ppt 작성 및 발표

### 성준모

-환자 정보 관리

### 이무원

-아이디어 기획 -위험도 평가 기준

### 최진서

-환자 위험도 전달

# 감사합니다.

Q&A