

Associate Architect 인증과제

스마트홈 전기 관리 시스템

시스템 개요

□ 스마트홈 전기 관리 시스템

- 스마트홈 전기 관리 시스템은 가정 내의 전기 사용을 실시간으로 모니터링하고, 자동 제어 및 최적화 기능을 통해 에너지 효율성을 높이며 전력 소비를 절감하는 **IoT** 기반 관리 시스템입니다.

■ 목적

- 에너지 절감: 불필요한 전력 사용을 줄이고 전기 요금 절감
- 편의성 향상: 원격 제어 및 자동화 기능으로 생활 편리성 증대
- 안전성 강화: 과부하, 누전, 이상 전류 감지 및 알림
- 친환경 실현: 에너지 효율화를 통한 탄소 배출 저감
- 데이터 기반 최적화: 전력 사용 패턴 분석을 통한 맞춤형 제어 정책 제공

시스템 컨텍스트

□ 사용자

- 전력 사용 현황 조회, 가전기기의 원격제어, 알림수신, 설정 관리 등을 사용하는 일반 가정내 사용자

□ 전력 공급사

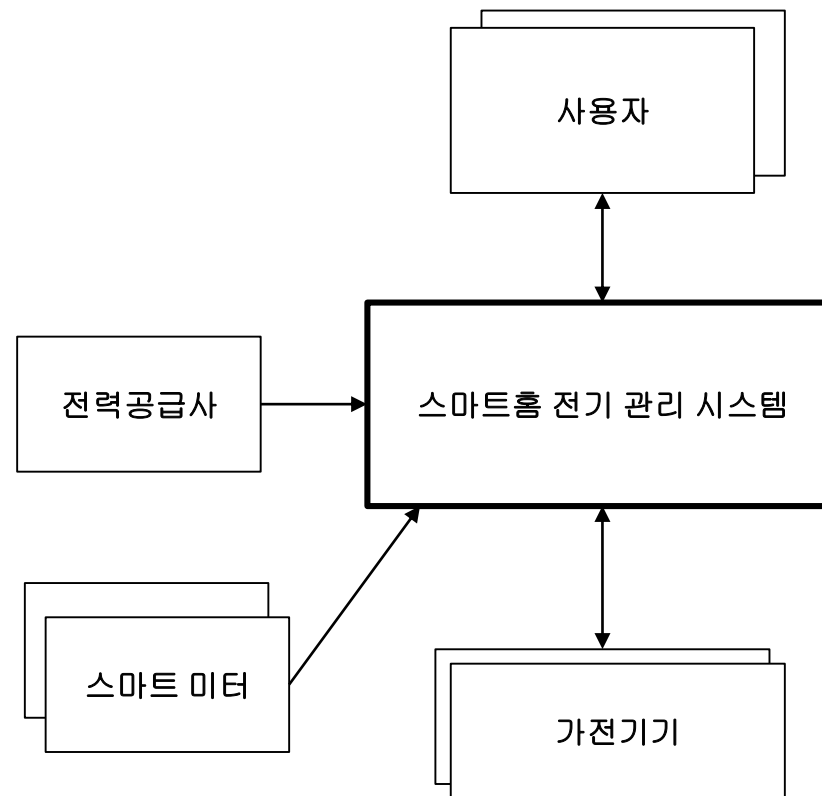
- 전력 요금제 및 단가 정보 제공
- 실시간 또는 일정 주기별 전력 사용량 데이터 제공 (스마트미터 연계)
- 정전, 전력 수급 상황, 긴급 공지 등의 정보 전달

□ 스마트 미터

- 가정 전체의 전력 사용량, 전압, 전류, 주파수 등 측정
- 이상 전력 패턴 감지 및 전송

□ 가전기기

- 원격 제어 명령(전원 **On/Off**, 모드 변경 등) 실행
- 사용 상태(동작 중, 대기 중, 꺼짐) 및 소비 전력 정보 전송



주요기능 요구사항

- 실시간 전력 사용량 모니터링
 - 가정 전체 및 개별 기기별 사용량 측정
 - 시간대별, 일별, 월별 그래프 제공
- 원격 제어
 - 스마트폰/웹을 통한 가전기기 전원 **On/Off**
 - 스케줄 기반 자동 제어 설정
- 에너지 절감 모드
 - 사용 패턴 분석을 통한 절전 제안
 - 대기 전력 차단 기능
- 이상 상태 감지 및 알림
 - 과부하, 누전, 과열 등의 이상 감지
 - 앱/문자/푸시 알림 발송
- 전기 요금 예측
 - 현재 사용량 기반 월말 요금 예측
 - 시간대별 요금제 반영한 최적 사용 가이드 제공

주요 비기능 요구사항

□ 성능(Performance)

- 실시간 데이터 업데이트 주기: 5초 이내
- 최대 100대의 IoT 기기 동시 연결 지원

□ 신뢰성(Reliability)

- 전력 데이터 수집 성공률 99% 이상
- 장애 발생 시 자동 복구(재시도) 기능

□ 보안(Security)

- 기기 제어 및 데이터 전송 시 암호화(SSL/TLS)
- 사용자 인증(2단계 인증) 지원

□ 확장성(Scalability)

- 새로운 IoT 기기 및 프로토콜 추가 가능

□ 가용성(Availability)

- 연간 시스템 가동률 99.5% 이상

인증과제

- 목표 시스템의 아키텍처 설계 보고서를 단계별로 작성하시오
 - Interim Report
 - Business Driver, System context analysis, architectural drivers
 - Prefinal Report
 - Architectural drivers(수정), High level Structural design description
 - Final report
 - High level Structural design description(수정)
 - Detailed Component design description, Architectural evaluation
- 시스템의 구체적인 비즈니스 환경, 기능 및 품질 요구사항, 제약사항 등은 제시된 기술서를 참고하여 각자 설정한다.
 - 요구사항 중 최소 1건 이상의 AI 기능을 적용
- 보고서 작성 분량
 - 100페이지 이내 (부록 포함)

평가 범위 및 배점

Part	Chapter	평가비중	Interim	Pre-final	Final	챕터별 배점
Architectural Requirements	1장. Business drivers	7%	40	10		50
	2장. System context	6%	30	10		40
	3장. Architectural drivers	16%	70	40		110
Architectural Design	4장. Top level structural design description	34%		150	90	240
	5장. Detailed component design description	29%			200	200
Architectural Evaluation	6장. Architectural evaluation	9%			60	60
배점(누계)			140	210	350	700

보고서 평가 기준

□ 명확성

- 보고서의 내용이 명확하게 전달되고 이해하기 쉬운 정도로 기술되어 있는지 평가
- 평가 포인트:
 - 문장 구조와 표현이 명확하게 이해되는가?
 - 개념과 아이디어가 혼란없이 전달되는가?
 - 필요한 용어와 개념을 명확히 정의하고 설명하는가?
 - 그림, 표 등이 명확하게 해석되는가?

□ 정확성

- 요구사항과 설계의 표현이 제시된 표현 방식에 정확하게 부합하는지 평가
- 평가 포인트:
 - 요구사항 기술 방식 (**Use case**, **QA scenario**, **Constraint**)이 정확한가?
 - 아키텍처 뷰 표현 방식 (**C&C** 뷰, **Allocation**뷰, **Module** 뷰)이 정확한가?

보고서 평가 기준

□ 타당성

- 보고서의 내용이 합리적이고 타당한지를 평가
- 평가 포인트:
 - 요구사항을 만족시키는 아키텍처 설계가 타당한가?
 - 후보 아키텍처 설계 중에 선택된 아키텍처 설계 의사결정이 합리적인가?
 - 아키텍처 분석 및 평가가 논리적으로 설득력이 있는가?

□ 일관성

- 보고서의 내용이 일관성 있게 전개되는 정도를 평가
- 평가 포인트:
 - 보고서에 사용된 용어, 개념, 이름 등이 일관되게 사용되는가?
 - 요구사항과 아키텍처 설계 사이에 모순되는 부분이 존재하는가?
 - 제출된 아키텍처 뷰 간의 일관성이 있는가?

설계 문서 작성시 생성형 AI 활용법 안내

- 생성형 AI는 아키텍처 설계 문서 작성 과정에서 자동화, 최적화, 품질 유지를 지원하는 강력한 도구입니다.
 - AI를 활용하여 시간을 절약하고, 일관된 문서를 유지하며, 더 나은 설계를 도출할 수 있습니다.
 - 단, 오류 (부정확한 용어 사용, Hallucination 등)가 있을 수 있으니, 결과를 반드시 재검증 해야 합니다.
 - 단순 Copy 등으로 오류 발견시 감점 진행됨
- 🔥 가장 중요한 것은 AI를 보조 도구로 활용하고, 최종 검토와 의사결정은 설계자 책임하에 수행되어야 한다는 점입니다.

요구사항 작성시

□ 활용 방법

- 핵심 키워드 혹은 간단한 설명을 바탕으로 요구사항 (예, 유스케이스 시나리오, 품질 속성 시나리오) 작성을 요청
- 생성된 출력을 검토하여 부적절한 부분에 대한 추가 프롬프트 제시를 통해 요구사항을 정제

✓ 프롬프트 예시:

다음은 도로 위험 관리 시스템의 기능이다.

위험 상황 감지

- 도로에 설치된 센서 및 CCTV 등을 통해 교통 흐름, 차량 속도, 도로 파손 여부 등을 실시간으로 수집
- AI 기반 영상 인식 기술을 활용하여 노면의 위험 상황 (침수, 포트홀 등)을 실시간 감지

이 기능을 유스케이스 시나리오 명세로 구체화하시오. 시나리오 명세기 Precondition, Post Condition, Basic Flow, Alternative Flow 관점에서 서술하시오

설계 대안 탐색시

□ 활용 방법

- 특정 요구사항에 맞는 아키텍처 패턴 추천 (예: **CQRS**, 이벤트 기반, 계층형 아키텍처)

✓ 프롬프트 예시:

“사용자가 많고 확장성이 중요한 **SaaS** 애플리케이션을 위한 아키텍처 설계를 도와줘. 적합한 패턴과 고려해야 할 요소들을 정리해 줘.”



사용자가 많고 확장성이 중요한 SaaS 애플리케이션 아키텍처 설계

SaaS 애플리케이션은 **멀티 테넌시, 확장성(Scalability), 고가용성(High Availability), 보안(Security)**이 핵심 요건입니다. 이를 고려한 적절한 아키텍처 패턴과 설계 요소를 정리합니다.

1. 적합한 아키텍처 패턴

1 마이크로서비스 아키텍처 (Microservices Architecture)

✓ 이유:

- 개별 서비스 단위로 확장이 가능하여 트래픽 증가에 유연하게 대응 가능
- 특정 기능(예: 결제, 인증 등)만 독립적으로 배포 및 운영 가능
- 장애 발생 시 일부 서비스만 영향을 받아 전체 시스템이 중단되지 않음



시스템 구성도 및 다이어그램 생성

□ 활용 방법

- 생성형 **AI**를 활용하여 시스템 아키텍처의 설명 초안을 작성
- 다이어그램 툴(예: **Draw.io**, **Lucidchart**, **PlantUML**)과 연계해 다이어그램 생성 자동화

✓ 프롬프트 예시:

"마이크로서비스 아키텍처를 기반으로, **API** 게이트웨이, 인증 서버, 서비스 레이어, 데이터베이스를 포함한 다이어그램을 설명해 줘.

- ✦ **팁:** AI가 다이어그램을 직접 생성할 수는 없지만, **PlantUML** 코드를 생성하여 시각적 작업을 쉽게 할 수 있음

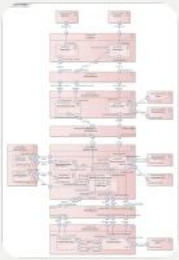
다이어그램에 대한 설명 추가시

□ 활용 방법

- 다이어그램 이미지를 첨부하고, 다이어그램에 대한 전반적인 설명을 요청

✓ 프롬프트 예시:

“다음 다이어그램에 대한 전반적인 설명을 제시하시오”



다음 다이어그램에 대한 전반적인 설명을 제시하시오

주어진 다이어그램은 **UML 컴포넌트 다이어그램**으로, 시스템의 주요 구성 요소와 그들 간의 상호작용, 데이터 흐름 및 통신 방식 등을 시각적으로 나타냅니다. 이 다이어그램은 로봇 배송 시스템의 아키텍처를 모델링한 것으로 보이며, 주요 구성 요소를 다음과 같이 설명할 수 있습니다:

1. 상위 구조

- 다이어그램은 여러 서브시스템과 외부 시스템, 데이터 저장소, 사용자 인터페이스(UI), 서버 간의 관계를 계층적으로 표현하고 있습니다.
- 주요 서브시스템으로는 **UserClient**, **RequestControlServer**, **ApplicationServer**, **RobotControlServer**, **MasterDBServer** 등이 있습니다.

2. 주요 구성 요소 설명

↓