

객체지향개발방법론

팀프로젝트#1_v2

Functional Requirements

Ref.#	Functional Requirements	Use-Case Number & Name	Category	Visibillity
R1.1	전원이 켜지면 로봇 청소기 시스템을 시작한다.	1.Power On System	Secondary	Evident
R2.1	청소 중에는 전진 이동한다. (이동 중 바닥 청소 및 물걸레 청소를 수행한다.)	2.Perform Automatic Cleaning	Primary	Evident
R3.1	전방, 좌측, 우측 센서를 통해 장애물을 감지한다.	3.Detect Obstacle	Primary	Hidden
R3.2	장애물 감지 시 전진 이동 및 청소를 중지한다.	3.Detect Obstacle	Primary	Evident
R3.3	전방이 막혀 있고 좌측이 비어 있으면 좌회전한다.	4.Avoid Obstacle – Turn Left	Primary	Evident
R3.4	전방과 좌측이 막혀 있고 우측이 비어 있으면 우회전한다	5.Avoid Obstacle – Turn Right	Primary	Evident
R3.5	전방, 좌측, 우측이 모두 막혀 있으면 후진 후 방향 전환한다.	6.Avoid Obstacle – Move Backward & Turn	Primary	Evident
R4.1	먼지 센서를 통해 먼지를 감지한다.	2.Perform Automatic Cleaning	Secondary	Hidden
R4.2	먼지 감지 시 청소 강도를 높인다.	7.Perform Boost Cleaning	Primary	Evident
R4.3	강화된 청소 강도를 일정 시간 유지한다.	7.Perform Boost Cleaning	Secondary	Evident
R5.1	전원 종료 시 모든 동작을 중지한다.	8.Power Off System	Secondary	Evident
R6.1	예외 상황 혹은 오류 발생 시 전원을 종료한다.	9.Power Off System – Exceptional	Secondary	Hidden

Use Case Descriptions

Use Case	1. Power On System
Actors	User (사용자)
Description	<ul style="list-style-type: none">• 이 Use Case는 사용자가 로봇청소기의 전원을 켜면서 시작된다.<ul style="list-style-type: none">◦ 이 단계에서 시스템을 초기화한다.
Use Case	2. Perform Automatic Cleaning
Actors	FrontSensor, DustSensor, Cleaner, Motor
Description	<ul style="list-style-type: none">• 이 Use Case는 로봇청소기의 자동 청소 모드가 비활성화된 상태에서 수행된다.• 시스템은 청소모드를 활성화하고 로봇청소기를 전진 이동 상태로 설정한다.<ul style="list-style-type: none">◦ 더불어, 바닥 청소 및 물걸레 청소를 수행하며 자동 청소를 시작한다.• 청소 도중 장애물이나 먼지가 감지되면 관련 Use Case가 수행된다.

Use Case Descriptions

Use Case	3. Detect Obstacle
Actors	Left Sensor, Right Sensor, Cleaner, Motor
Description	<ul style="list-style-type: none">• 이 Use Case는 전방 장애물이 감지된 상태에서 수행된다.• 좌측 및 우측 장애물을 감지하고 적절한 Avoid Obstacle - Use Case 가 수행된다.
Use Case	4. Avoid Obstacle - Turn Left
Actors	Motor
Description	<ul style="list-style-type: none">• 이 Use Case는 전방이 막혀 있고 좌측이 비어 있을 때 수행된다.• 시스템은 현재 이동을 중지한 뒤 로봇청소기를 좌측으로 회전시킨다.• 회전 완료 후 Perform Automatic Cleaning - Use Case를 수행한다.

Use Case Descriptions

Use Case	5. Avoid Obstacle – Turn Right
Actors	Motor
Description	<ul style="list-style-type: none">• 이 Use Case는 전방과 좌측이 막혀 있고 우측이 비어 있을 때 수행된다.• 시스템은 로봇청소기를 우측으로 회전시킨다.• 회전 완료 후 Perform Automatic Cleaning – Use Case를 수행한다.
Use Case	6. Avoid Obstacle – Move Backward & Turn
Actors	Left Sensor, Right Sensor, Motor, Cleaner
Description	<ul style="list-style-type: none">• 이 Use Case는 전방, 좌측, 우측이 모두 막혀 있을 때 수행된다.• 시스템은 로봇청소기를 일정 거리 후진시킨다.• 후진 완료 후, 가능한 방향으로 회전을 실행하고 회전 완료 후 Perform Automatic Cleaning – Use Case를 수행한다.

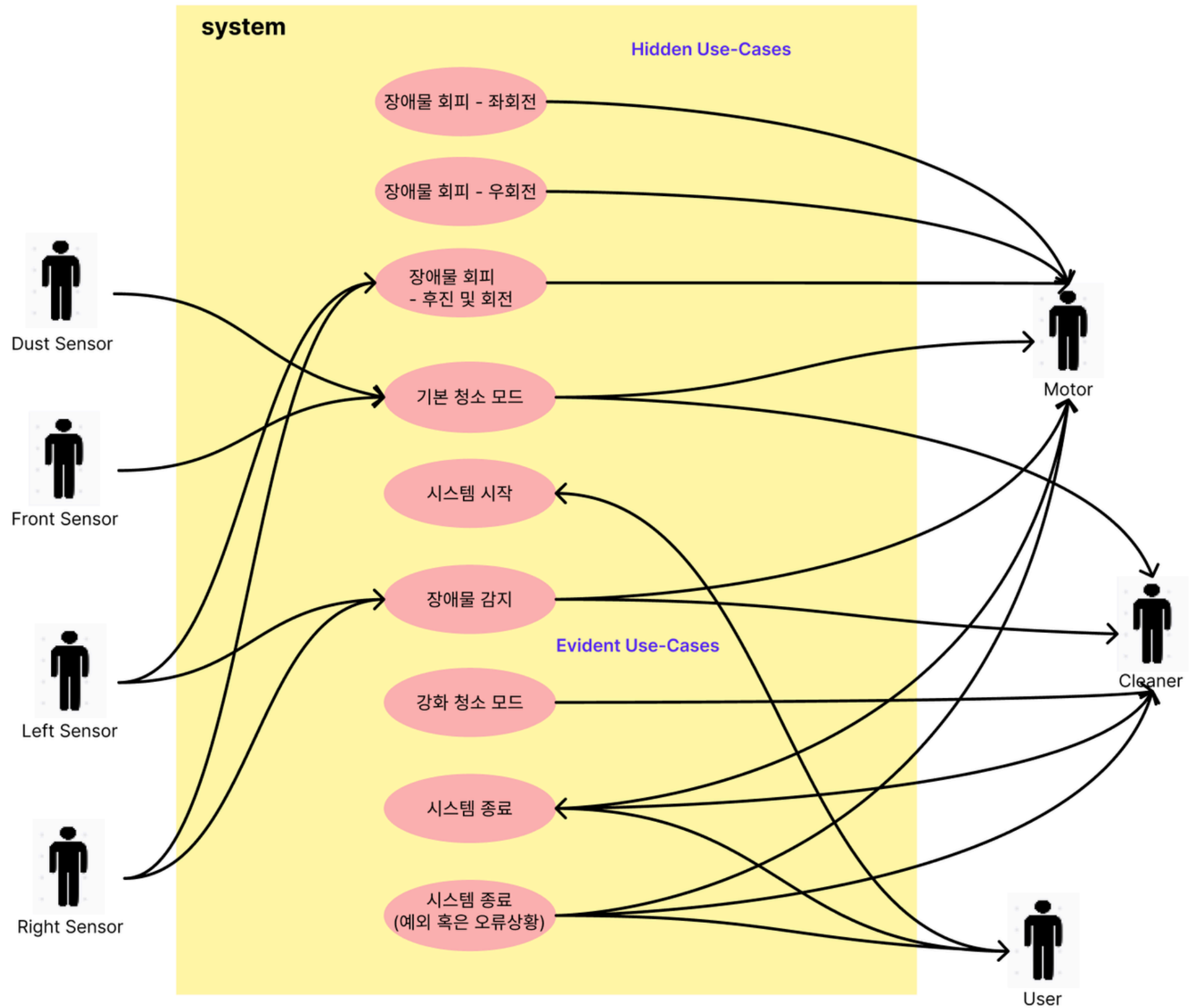
Use Case Descriptions

Use Case	7. Perform Boost Cleaning
Actors	Cleaner
Description	<ul style="list-style-type: none">• 이 Use Case는 먼지 센서가 먼지를 감지했을 때 수행된다.• 시스템은 청소 강도를 높여 강화 청소 모드를 활성화한다.<ul style="list-style-type: none">◦ 단, 현재강도가 이미 최댓값인 경우 현재 상태를 유지한다.• 강화된 청소 모드는 일정 시간 동안 유지된다.• 일정 시간이 지나면 시스템은 기본 청소 모드로 복귀한다.
Use Case	8. Power Off System
Actors	User (사용자), Motor, Cleaner
Description	<ul style="list-style-type: none">• 이 Use Case는 사용자가 로봇청소기의 전원을 끌 때를 가리킨다.• 시스템은 현재 수행 중인 동작을 중지한다.<ul style="list-style-type: none">◦ 모든 모터와 청소 장치를 정지시키고 시스템을 안전한 종료 상태로 전환한다.

Use Case Descriptions

Use Case	9. Power Off System – Exceptional
Actors	User (사용자), Cleaner, Motor
Description	<ul style="list-style-type: none">• 이 Use Case는 시스템이 예외 또는 오류를 감지해 전원을 끌 때를 가리킨다.• 시스템은 현재 예외 또는 오류를 사용자에게 알리고 수행 중인 동작을 중지한다.<ul style="list-style-type: none">◦ 모든 모터와 청소 장치를 정지시키고 시스템을 안전한 종료 상태로 전환한다.

Use Case Diagram



Non-Functional Requirements

Performance Requirements (Quality)

NFR-P-01	전원 켜짐 시 청소 상태가 되어 첫 이동을 시작하기까지의 초기화 시간은 5초 이내여야 한다.
NFR-P-02	전방 센서로부터 장애물 감지 시 정지 상태로 전환 시간은 50ms 이내여야 한다.
NFR-P-03	좌회전, 우회전 결정 시 판단 및 모터 제어 신호 발생까지의 지연 시간은 500ms 이내여야 한다.
NFR-P-04	전/좌/우 방향이 모두 장애물로 막힌 경우 정지 후 100ms 이내에 후진을 시작하고, 후진 완료 후 회전 방향을 500ms 이내에 결정해야 하며, 결정된 후 500ms 이내에 모터 제어 신호가 발생해야 한다.
NFR-P-05	먼지 센서로부터 먼지 감지 시 터보 청소 모드 상태로 전환 시간은 100ms 이내여야 한다.
NFR-P-06	터보 청소 모드는 5초간 유지되고, 이후 터보 청소 모드를 해제한다.

Non-Functional Requirements

Other requirements (Quality)

NFR-0-01	시스템의 주요 구성 요소는 추상 클래스로 정의하여 구현 간 결합도를 낮추고 변경에 유연하도록 설계되어야 한다.
NFR-0-02	실제 외부 의존성 없이도 단위 테스트가 가능하도록 Stub 객체를 구현해야 한다.
NFR-0-03	시스템은 정상 동작 중 오류가 발생하지 않도록 설계되어야 하며 예외 상황 발생 시 안전한 종료 상태로 전환하는 예외 처리 로직을 포함해야 한다.
NFR-0-04	시스템은 입력 데이터에 대해 유효성 검증을 수행해야 하며 검증되지 않은 입력은 처리하지 않아야 한다.
NFR-0-05	시스템은 민감한 인증 정보를 소스 코드에 하드코딩하지 않아야 하며 안전한 저장 방식을 통해 보호되어야 한다.
NFR-0-06	회전 동작 중 센서 입력이 들어오더라도 시스템은 입력을 무시하고 회전 동작을 완수한다.
NFR-0-07	터보 청소 모드 중 먼지 센서 입력이 들어오더라도 시스템은 입력을 무시하고 터보 청소 모드를 완수한다.

Non-Functional Requirements

Other requirements (Quality)

NFR-0-08	터보 청소 모드 중 전방 및 좌우센서 입력에 따른 후진 및 회전 동작은 실행되어야 한다.
NFR-0-09	장애물 감지와 먼지 감지 중 항상 전자를 우선한다.

Non-Functional Requirements

Operating Environment

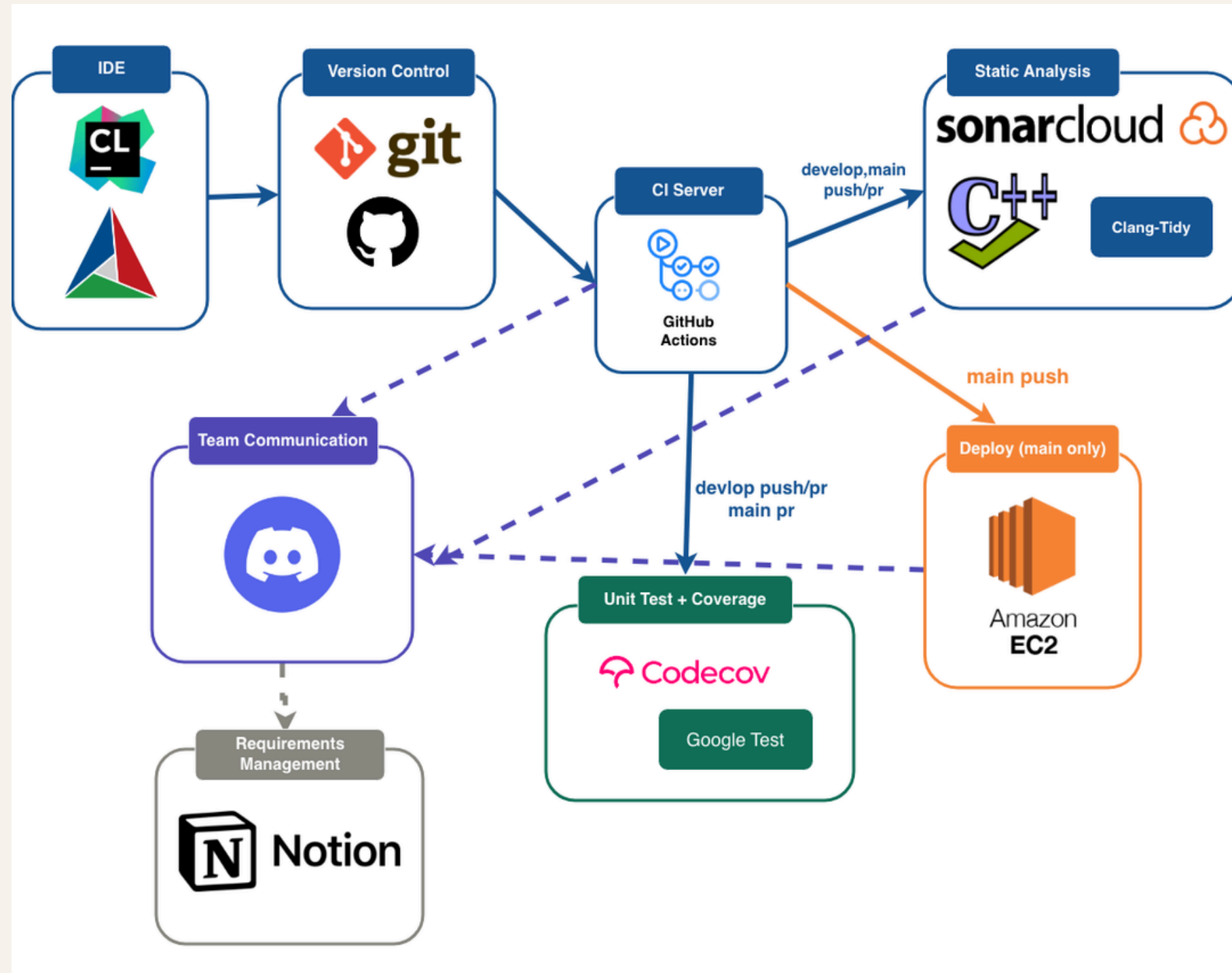
NFR-OE-01	시스템은 C++17 표준으로 구현하며 디 빌드 환경과 로컬 개발 환경에서 동일한 CMake 3.14 버전으로 빌드가 성공해야 한다.
-----------	---------------------------------------------------------------------------

Non-Functional Requirements

Interface Requirements

NFR-I-01	로봇 청소기의 방향 전환, 청소 모드와 같은 상태는 터미널 화면에 실시간 텍스트 로그 형태로 출력되어야 한다.
NFR-I-02	사용자는 터미널 기반 CLI를 통해 전원 ON/OFF 명령을 입력할 수 있어야 한다.

CI/CD Environment



Version Control & CI/CD Server

Git/GitHub 기반 버저닝 및 GitHub Actions를 통한 통합 파이프라인 구축

Unit Test & Coverage

Google Test를 활용한 단위 테스트 자동화 및 Codecov를 통한 커버리지 시각화

Static Code Analysis

SonarCloud, Cppcheck, Clang-Tidy
3종의 정적 분석 도구를 병렬로 실행하여 코드 품질 보장

Deploy & Team Communication

AWS EC2 자동 배포 및 Discord Webhook을 통한 실시간 빌드/배포 상태 공유

브랜치 전략

main : 보호 브랜치 / PR → CI / Merge → CD

develop : 통합 브랜치 / PR 및 Merge → CI

CI/CD Environment

관련 규칙

- main/develop 브랜치로의 모든 Merge는 자동화 검사 통과와 최소 1명 이상의 Peer Review 승인을 거쳐야 한다.
- SonarCloud, Clang-Tidy, Cppcheck 분석 결과 중 'Blocker' 및 'Critical' 이슈는 0건이어야 배포 가능하다.
- Google Test를 활용하여 핵심 비즈니스 로직을 검증하며 Codecov 기준 라인 커버리지 80% 이상을 상시 유지한다.
- 모든 CI/CD 파이프라인의 성공/실패 여부는 Discord Webhook을 통해 실시간으로 공유되어야 한다.
- AWS EC2 접속 및 서버 통신에 필요한 인증 정보는 GitHub Secrets에 암호화하여 관리하며, 소스 코드 내 하드코딩을 금지한다.

감사합니다.