




# 소프트웨어공학: 프로젝트 계획서

기술 개발 과제	AI 음성인식 Mobile Robot Controller (영문 : Mobile Robot Controller with AI Voice Recognition)			
과제 팀 이름	커널		지도교수	이병정
개발 기간	2023년 9월 25일 ~ 2023년 11월 24일 (총 2개월)			
개발 소요 비용	총액	₩ 10,084,940	학교부담금	₩ 9,000,000
			과제팀부담금	₩ 1,084,940
과제 팀 구성원	이름	이성호	김민교	이정우
	사진			
	학번	2019920037	2019920010	2021920043
	연락처	010-4195-2626	010-8982-6576	010-5103-8541

소프트웨어공학 프로젝트를 성실히 수행하고자 계획서를 제출합니다.

2023년 10월 16일

과제 수행자 1 : 이성호 (이성호)  
 과제 수행자 2 : 김민교 (김민교)  
 과제 수행자 3 : 이정우 (이정우)

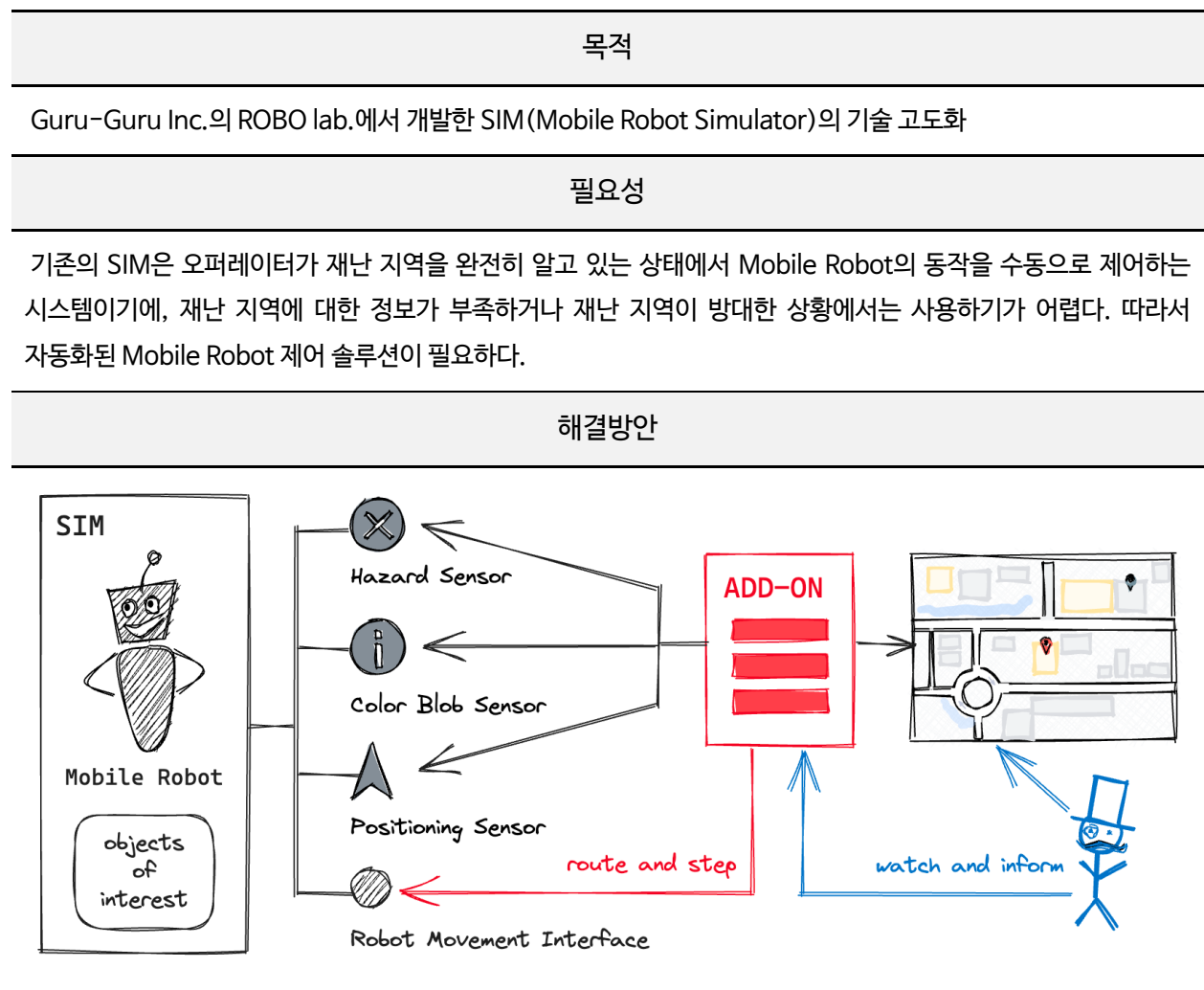
서울시립대학교 컴퓨터과학부 귀중

# 1. 개발 과제의 내용

## 가. 개발 과제 요약

Guru-Guru Inc.의 ROBO lab.에서 개발한 SIM(Mobile Robot Simulator) 기술을 고도화하는 프로젝트이다. 수동으로 제어하던 기존의 SIM의 Mobile Robot에 컴퓨터 알고리즘을 활용한 자동 경로 탐색 기능을 추가하며, AI 기술을 통한 음성 인식 기능을 도입하여 재난 상황을 실시간으로 보고하고 및 갱신한다. 프로젝트는 MVC(Model-View-Controller) 패턴을 기반으로 개발되며, 세부적으로는 SIM 프레임워크, UI 개발, ADD-ON 개발(안전 경로 탐색 알고리즘 및 음성 인식 AI 기술 개발)로 구성된다. 성공적인 구현을 통해 재난 지역에서의 자동화된 Mobile Robot 동작 제어와 위험 지점에 대한 신속한 대응이 가능할 것으로 예상된다.

## 나. 개발 과제의 배경 및 효과



SIM에 ADD-ON(추가 기능)을 등록할 수 있도록 한다. 이미 알고 있는 위험 지역 정보와 Mobile Robot의 센서 정보를 취합 및 가공하여 Mobile Robot이 목표 지점까지 안전하게 이동할 수 있는 경로를 제공하는 ADD-ON을 개발한다. 그리고 Mobile Robot이 미처 감지하지 못했거나, 사람에 의해 발견한 위험 지점을 SIM에 전파하기 위해 AI 기반 음성 인식 및 위험 지점 전파 ADD-ON을 개발한다. 해당 ADD-ON은 “위험 지점 x y 발견”과 같은 사람의 음성을 인식하여 새로운 위험 지점을 사전에 알리고, 안전 경로를 새롭게 구성할 수 있도록 한다.

#### 효과 및 우수성

커널 팀의 ADD-ON 도입 솔루션은 로봇의 자동화된 동작 제어를 가능하게 하며, 재난 지역의 정보가 부족하거나 방대한 경우에도 효과적으로 대응할 수 있게 한다. 무엇보다 새로운 기능을 추가하기 위하여 핵심 기능을 크게 수정하지 않고도 새로운 요구 사항을 만족시킬 수 있게 한다. 이러한 기술의 성공적인 구현은 화재, 지진의 발생, 폭발, 독성 물질 유출 등과 같은 여러 종류의 재난에 대해 정확하고 신속하게 대응할 수 있을 것으로 예상되며, 이는 결국 사람들의 생명을 구하는데 큰 도움이 될 것이다.

### 다. 요구 분석

개발 과제를 수행하기에 앞서 유스케이스 분석을 통해 어떤 모델 및 기능들이 필요한지 살펴본다.

#### 안전 경로로 로봇을 이동

- 액터 : 오퍼레이터
- 시작 조건 : 로봇과 시뮬레이터가 초기화되어 작동 가능하며 로봇이 아무런 동작을 하지 않는 상태이다.
- 기본 흐름
  1. “오퍼레이터”가 “STEP” 버튼을 클릭한다.
  2. “로봇”과 통신하여 “positioning sensor”를 통한 현재 위치 정보, “hazard sensor”를 통한 전방 1칸 앞의 위험 지점 유무, “color blob sensor”를 통한 사방의 중요 지점 유무를 파악한다.
  3. 전방 1칸 앞의 위험 지점과 사방의 중요 지점 유무를 통해 “재난 지역(지도)” 정보를 갱신한다.
  4. 사전에 정의된 “재난 지역 모델(지도)” 정보와 로봇의 현재 위치를 기반으로 로봇이 “중요 지점” 또는 “목표 지점”까지 안전하게 이동할 수 있는 다음 위치를 탐색한다.
  5. 다음 위치로 가기 위해 “로봇”을 시계 방향으로 90도 회전하거나 해당 방향으로의 1칸 전진하여 한 “STEP”을 진행한다.
- 종료 조건 : 지정된 위치 또는 방향으로 로봇을 안전하게 움직인다.

#### AI 음성 인식을 기반으로 위험 지점을 추가

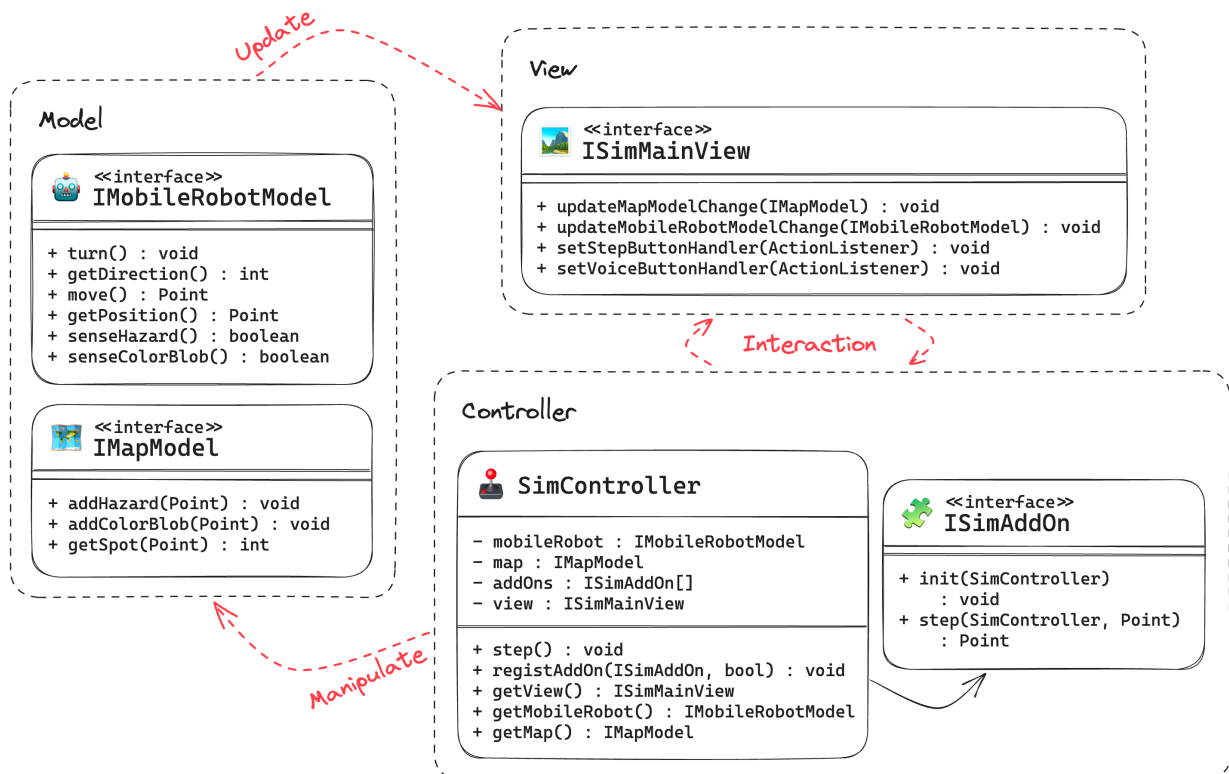
- 액터 : 오퍼레이터
- 시작 조건 : 로봇과 시뮬레이터가 초기화되어 작동 가능하며 로봇이 아무런 동작을 하지 않는 상태이다.

- 기본 흐름
  1. “오퍼레이터”가 “VOICE” 버튼을 클릭한다.
  2. “시뮬레이터”에 연결된 마이크를 이용하여 오퍼레이터의 음성 “중요/위험 지점 x y 발견”을(를) 녹음한다.
  3. STT 기술을 활용하여 “오퍼레이터”의 음성을 문자열로 변환한다.
  4. 문자열로 변환된 “오퍼레이터”의 음성으로부터 “위험 지점 좌표”를 추출한다.
  5. 추출된 “위험 지점 좌표”를 현재 지도에 갱신한다.
- 종료 조건 : 오퍼레이터가 전파한 위험 지점 위치를 성공적으로 지도 위에 갱신한다.

## 라. 개발 과제의 목표와 내용

MVC 패턴을 기반으로 SIM을 체계적으로 모듈화하여 각 컴포넌트를 독립시키고 개발 의존성을 크게 낮춘다. 그리고 지속적인 통합 및 테스팅을 통해 솔루션을 성공적으로 개발한다.

### SIM ADD-ON 프레임워크 개발



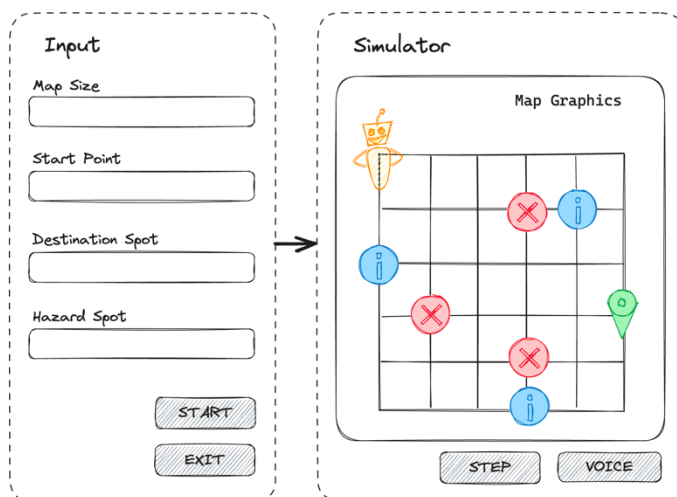
MVC 패턴에 따라, Model을 정의한다. Model은 애플리케이션의 정보(데이터)를 나타내며, Mobile Robot을 나타내는 **IMobileRobotModel 인터페이스**, 재난 지역(지도)을 나타내는 **IMapModel 인터페이스**를 설계하여 이를 구현한다. **IMobileRobotModel** 인터페이스에는 로봇이 한 칸 앞으로 가며, 한번에 시계 방향으로 90도만 회전

가능하다는 제약조건을 토대로, 각각 move와 turn 메서드를 정의한다. 또한, 전방의 위험 지점 또는 사방의 중요 지점을 파악할 수 있는 senseHazard, senseColorBlob 메서드를 구현해야 한다. 한편, 오퍼레이터로부터 지도 정보를 입력받아 재난 지역 모델을 구성하기 위해 IMapModel을 구현해야 하며, 위험 지점 또는 중요 지점이 새로 갱신될 수 있는 요구사항을 만족하기 위하여 addHazard와 addColorBlob이라는 메서드가 존재한다. 한편, Model이 변경되는 경우 다음으로 소개할 View를 갱신하도록 한다.

View는 Model의 상태 및 정보를 받아 시각화를 담당하는 부분이며, 이는 ISimMainView 인터페이스를 통해 구현하도록 한다. 앞에서 살펴봤던 Model은 자신의 상태가 변경되었을 때 상황에 맞게 ISimMainView의 updateMapModelChange와 updateMobileRobotModelChange 메서드를 호출하여 View에 변경된 상태를 전파한다. 이로써 View는 Model의 변경 사항을 감지하고 화면을 갱신할 수 있게 된다. 또한, ISimMainView에는 유스케이스 분석을 통해 알아낸 STEP 및 VOICE 버튼에 대한 이벤트 처리를 하여야 하는데, setStepButtonHandler 및 setVoiceButtonHandler라는 메서드를 통해 이벤트 처리를 위임한다. 예를 들어, 로봇의 한 STEP을 수행하는 버튼이 눌렸을 때 다른 클래스에서 정의한 ActionListener 인스턴스를 받아 대신 이벤트를 처리한다. 이를 통해 View는 출력에만 집중하게 되고, 다른 작업은 별도의 클래스가 처리하는 형태가 되어 유지보수가 용이해진다.

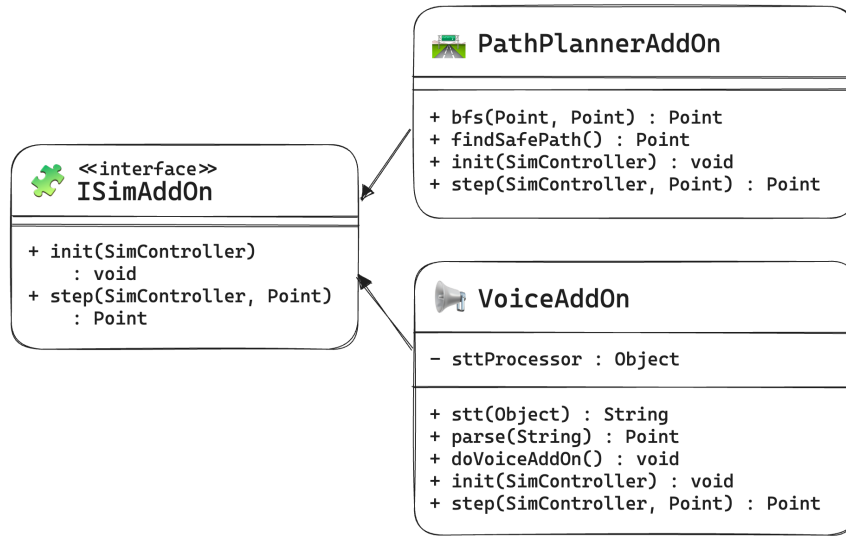
Controller는 Model을 조작하는 역할을 수행한다. 이에 SimController라는 클래스를 정의하며, 이는 Model과 View를 모두 통합함과 동시에 ISimAddOn이라는 인터페이스 구현체의 인스턴스를 리스트로 등록받는다. ISimAddOn 인터페이스는 step 메서드를 구현하게 되어 있는데, 해당 인터페이스의 step 메서드들은 SimController의 로봇을 한 번 움직이게 하는 메서드를 호출할 때 순차적으로 호출되도록 한다. 이렇게 되면, 추가로 개발할 안전한 경로 탐색, 음성 인식 기능 등의 ADD-ON들을 효과적으로 통합할 수 있게 된다. 그리고 각 ADD-ON 개발 담당자들은 다른 ADD-ON에 독립적인 개발을 보장받을 수 있다.

## UI 개발



총 2개의 품을 구현한다. 하나는 Input 품이고, 다른 하나는 Simulator 품이다. 두 UI 모두 Java Swing 프레임워크를 사용하여 개발한다. Input 품은 지도의 크기, 로봇의 시작 위치, 탐색(중요 지점)의 위치, 위험 지점의 위치를 TextField를 통해 입력 받는다. 그리고 입력 받은 정보를 바탕으로 IMapModel 및 IMobileRobotModel 인터페이스를 구현한 클래스의 인스턴스를 초기화한다. 다음으로 Simulator 품은 초기화된 정보를 바탕으로 Graphics 클래스를 이용하여 품 위에 지도 정보와 로봇의 위치를 표시한다.

그리고 ISimMainView 인터페이스를 구현하여 View로서의 기능을 수행하도록 한다.



#### (1) PathPlannerAddOn 클래스 : 안전한 경로를 탐색하는 알고리즘 기술 개발

ISimAddOn 인터페이스를 구현하여 주어진 재난 지역 모델(IMapModel)로부터 안전한 경로를 탐색하는 알고리즘 기술을 개발한다. 우선 시작점과 끝점, 그리고 지도의 정보를 매개변수로 받는다. 그 후 지도의 정보를 통해 시작점부터 끝점까지의 최단 경로를 BFS 및 역추적 알고리즘을 이용하여 찾고, 다음 이동할 점을 반환하도록 한다. 의사 코드는 다음과 같다.

```

bfs(stPoint: Point, edPoint: Point, map: MapModel) -> Point:
    q := Queue<Point>
    visited := Set<Point>
    prev := map<Point, Point>
    route := list<Point>
    visited.add(stPoint)

    while q is not empty:
        curP := q.pop()
        for each nxtP adjacent to curP:
            if not nxtP in visited and nxtP is not hazard:
                visited.add(nxtP)
                prev[nxtP] := curP

    for (Point p := endPoint; p != stPoint; p := prev[p])
        route.push_back(p)
    reverse(route)

    return route[0]
    
```

다른 알고리즘으로 A\*나 CCH를 이용할 수도 있다.

## (2) VoiceAddOn 클래스 : 위험 지점 음성 인식 AI 기술 개발

ISimAddOn 인터페이스를 구현하여 오퍼레이터의 위험 지점 전파 음성 메시지를 현재 위험 지역 정보(지도)에 갱신하는 기능을 개발한다. 우선 오퍼레이터의 장치 내에 있는 마이크를 활용하여 음성을 녹음한다. 다음으로 Naver에서 제공하는 AI 기반 CLOVA Speech Recognition(CSR)을 이용하여 녹음된 음성을 문자열로 변환한 후, 이를 적절히 파싱하여 원하는 위험 지점의 위치 정보를 반환한다. 녹음될 음성 형식은 “위험 지점 x y 발견”과 같은 식이다. 그리고 해당 클래스가 SimController에게 주입될 때, ISimMainView의 setVoiceButtonHandler를 호출하여 VOICE 버튼의 클릭 이벤트를 등록한다.

## 2. 완료 작품의 평가 방법

번호	평가 항목	평가 방법	비중 (%)
1	지도 및 로봇 초기화 기능	<ul style="list-style-type: none"><li>- 시뮬레이터 시작 시 처음 입력하는 지도 정보(지도 크기, 시작 위치, 탐색 위치, 위험 지점)가 제대로 SIM에 전달되는가?</li><li>- 로봇이 정상적으로 초기화되어 전진, 회전, 센서 값 취득 기능을 수행하는가?</li></ul>	100%
2	재난 지역(지도 정보) 시각화 기능	<ul style="list-style-type: none"><li>- 중요, 위험 지점 그리고 로봇의 위치 및 방향이 오퍼레이터의 화면 상에 그래픽으로 제대로 표시되는가?</li><li>- 로봇이 움직이거나 중요 및 위험 지점이 추가되었을 때 지도 그래픽이 갱신되는가?</li></ul>	100%
3	안전 경로 탐색 기능	<ul style="list-style-type: none"><li>- 로봇이 위험 지점을 피해 안전한 경로를 탐색하여 최종 목적 지점에 도착하는가? → <u>검증 데이터셋 32개를 만들어 성공률을 계산한다.</u></li></ul>	100%
4	사전 정의된 문자열 형식에 대한 음성 인식 및 위험 지점 파악 기능	<ul style="list-style-type: none"><li>- 사전 정의된 문자열 형식 “중요/위험 지점 (x, y) 발견”에 대한 사람의 음성을 인식하여 위험 지점의 위치를 판단할 수 있는가? → <u>시행을 총 100번 수행하여 성공률을 계산한다.</u></li></ul>	100%
5	다양한 문자열 형식 집합에 대한 위험 지점 파악 기능	<ul style="list-style-type: none"><li>- 사전 정의된 문자열 형식이 아닌 “위험지점 발견! 3, 3”이나 “3, 3에 중요 지점 발견”과 같은 다양한 문자열 형식에 대해 위험 지점을 판단할 수 있는가? → <u>다양한 문자열 형식으로 이루어져 있는 검증 데이터셋 32개를 만들어 성공률을 계산한다.</u></li></ul>	50%

### 3. 기술개발 일정 및 추진체계

#### 가. 기술개발 일정



개발하고자 하는 과제와 목표에 따라 아래와 같은 Gantt Chart를 제시한다. 10월 1주차부터 11월 4주차까지의 기간 중 수행해야 할 작업들을 나열하고 그 담당자를 명시하였다. 그리고 작업 별 목표 일정에 따라 해당 주차에 색칠하였다.

번호	작업	담당자	10-1	10-2	10-3	10-4	11-1	11-2	11-3	11-4
1	프로젝트 계획서 작성	모두								
2	SIM 모델링	모두								
3	SIM 프레임워크 개발	이성호								
4	중간 발표 준비	모두								
5	UI 개발	김민교								
6	IMobileRobotModel 구현체 개발	이정우								
7	IMapModel 구현체 개발	이정우								
8	PathPlannerAddOn 구현	이정우								
9	VoiceAddOn 구현	김민교								
10	통합 및 테스트	이성호								
11	최종 발표 준비	모두								

#### 나. 추진 체계

개발 인력	<p><b>[이성호, 팀장, JAVA 초급 개발자, 정보처리기능사 보유]</b></p> <ul style="list-style-type: none"> <li>- 프로젝트 문서 관리 및 계획, 일정 및 리스크 관리</li> <li>- SIM 프레임워크 및 세부 인터페이스 구현, 통합 및 테스트</li> </ul> <p><b>[김민교, 팀원, JAVA 초급 개발자]</b></p> <ul style="list-style-type: none"> <li>- UI 및 AI 음성 인식 기반 위험 지점 전파 기술 개발</li> </ul> <p><b>[이정우, 팀원, JAVA 초급 개발자]</b></p> <ul style="list-style-type: none"> <li>- Mobile Robot 및 Map 모델 개발, 안전 경로 탐색 알고리즘 개발</li> </ul>
-------	---



소스 코드 형상 관리  	<ul style="list-style-type: none"> <li>- Git을 이용하여 소스 코드 형상 관리를 진행한다. 그리고 GitHub를 원격 저장소로 두어 모든 구성원이 인터넷으로 소스 코드를 공유할 수 있도록 한다.</li> <li>- SIM 프레임워크 개발 담당자는 Framework 브랜치에서 개발을 수행한다.</li> <li>- SIM 프레임워크를 활용하여 ADD-ON을 개발하는 인원은 Framework 브랜치로부터 새로운 브랜치로 분기하여 개발을 진행한다.</li> <li>- 유스케이스 별로 개발이 완료되면 main 브랜치에 병합한다.</li> </ul>
개발 문서 형상 관리  	<p>GitHub Issue 기능을 이용하여 개발 문서의 형상 관리를 진행한다. GitHub Issue는 프로젝트 관리와 협업을 간편하고 효율적으로 할 수 있도록 도와주는 도구이다. 이 기능을 이용하면 다음과 같은 작업을 수행할 수 있다.</p> <ul style="list-style-type: none"> <li>- 문제 추적(Issue Tracking) : 프로젝트에서 발생하는 문제, 버그, 기능 요청 등을 Issue로 생성하고 추적할 수 있다. 각 Issue는 고유한 번호를 가지며, 제목, 설명, 댓글, 레이블 등의 정보를 포함할 수 있다.</li> <li>- 작업 할당과 협업 : Issue를 개발 담당자에게 할당하고, Issue에 댓글을 남기며 토론하고 협업할 수 있다.</li> <li>- 우선순위 관리 : Issue에 우선순위를 할당하고, Milestone과 연결하여 작업의 진행 상황을 추적할 수 있다.</li> <li>- 통합 : GitHub Issue는 Git 저장소와 통합되어 있어, commit message나 pull request와 연결하여 어떤 변경 사항이 어떤 Issue와 연관되어 있는지 추적할 수 있다.</li> </ul>

## 4. 위험 분석

### 1. 중간 시험 기간

10월 3주차에서 4주차 사이는 구성원들의 중간 시험 기간이므로 프로젝트에 시간을 많이 투자하기 힘들다. 따라서 개발하려는 솔루션에 대해 면밀한 모델링 및 이해를 위주로 프로젝트를 진행한다.

### 2. 소프트웨어 아키텍처의 결함 가능성

11월부터 본격적으로 개발을 진행하는 데 있어, 10월에 설계했던 소프트웨어 아키텍처 및 모델들이 잘못된 방향으로 이루어져 있을 수 있다. 예를 들어, 클래스별 의존 관계가 너무 복잡하거나, 유스케이스 분석을 잘못된 경우가 이에 해당될 것이다. 이런 경우를 미연에 방지하기 위해 개발에만 몰두하는 것이 아닌 구성원 전원이 로버트 C. 마틴의 “클린 아키텍처”라는 책을 읽으며 소프트웨어 공학 지식을 향상 공부할 수 있도록 한다.

### 3. AI 음성 인식 모델 성능 기준 미달 가능성

Naver에서 개발한 CLOVA Speech Recognition(CSR) 기술의 성능이 해당 솔루션의 요구 사항을 만족하지 못할 경우, Google에서 제공하는 Speech-To-Text 기술을 사용한다.

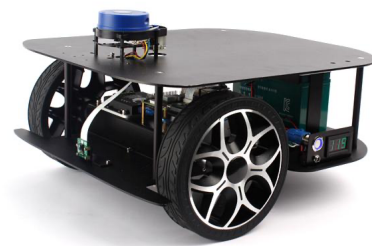
## 5. 개발 환경

### 가. SW 명세

JDK	OpenJDK 21
운영 체제	OpenJDK 21이 구동되는 모든 운영체제 (이하 Windows, Linux, Mac)

### 나. HW 명세

개발용 PC	MacBook Air M2 13형 - Apple M2 (8코어 CPU, 8코어 GPU), 8GB 통합 메모리, 256GB SSD 저장 장치
개발용 Mobile Robot	STELLA N1 - 크기 : 380 X 440 X 253 mm, 무게 : 7.18 kg, 적재 용량 : 5 kg, 재질 : 알루미늄 (플랫폼 및 바퀴) - 모터 : 10W DC Motor (감속 27 : 1), 속도 : 9.5 km/h (max) - 엔코더 해상도 : 500 PPR, 바퀴 둘레 : 56.52 cm, 두 바퀴 간의 거리 : 33.7cm - SoC : Quad-core Cortex-A73(up to 2.4 GHz) & Dual-core Cortex-A53(up to 2 GHz) - GPU : Mali-G52 GPU with 6 x Execution Engines (800 MHz) - RAM : 4GB LPDDR4 - 인터페이스 : RS232 통신 제어, 배터리 : 12V(12Ah) 연축전지 - 동작 온도 : 0 ~ 50도, 습도 : 85% 이하  개발용 Mobile Robot을 구할 수 없는 경우에는 개발용 PC위에서 로봇 시뮬레이션을 진행한다.
통신용 무선 라우터	EFM ipTIME A1004 유무선공유기 - 기가비트, 4개의 LAN 포트, Wi-Fi 5 지원, 듀얼 밴드(2.4 GHz, 5 GHz) - AI 기반 음성 인식을 위해 인터넷에 연결되어 있어야 함



## 6. 개발 사업비 및 산정 내역서

	항목 (품명, 규격)	수량	단가	금액	비고
직 접 개 발 비	개발자 인건비	3명 * 60시간 =180시간	₩ 20,000 / 1시간	₩ 3,600,000	- 초급 개발자 기준 - 프로젝트 기간 60일 1시간씩
	개발용 PC MacBook Air M2 13형	3대	₩ 1,590,000	₩ 4,770,000	
	개발용 Mobile Robot STELLA N1	1대	₩ 1,650,000	₩ 1,650,000	
	통신용 무선 라우터 EFM ipTIME A1004	1대	₩ 41,900	₩ 41,900	
	NAVER CLOVA Speech Recognition(CSR) 서비스 이용료	24시간	₩ 4 / 15초	₩ 23,040	음성 인식 이용 시간(음성 인식 요청~종료), 15초당 4원 부과
	합계			₩ 10,084,940	

## 7. 보고 및 감시 체계

- 정기 회의 : 매주 일요일 19시, 1시간 동안 Google Meet 화상 회의로 진행한다.
- 보고 내용 : 개발 목표치 달성 여부 보고, 개발 중 발생한 어려움 및 시행 착오 공유, 코드 리뷰
- 수시 보고 : 정기 회의 외의 보고 내용은 카카오톡 및 GitHub Issue를 이용하여 진행한다.
- 감시 체계 : GitHub Issue를 통하여 개발 문서에 대한 형상 관리를 진행하며, 산출물은 Google Drive를 통해 공유 및 관리한다. 각자의 개발 내용에 대해 git commit 기록을 확인하여 개발이 원활히 진행되고 있는지 확인한다.

끝.