

소프트웨어 개발 계획서

(Mobile Robot Controller)

2007920010 김영훈

2011920035 안주영

2012920056 조한주

1. 개요

화재, 지진, 폭발, 독성 물질 유출과 같은 심각한 재난 상황 발생 가능성에서 완전히 벗어날 수 있는 방법은 없다. 이러한 재난 상황이 발생한다면 상황의 악화를 막기 위해 재발의 위험이 있는 곳을 파악하고 빠른 구조 작업을 실시하는 것이 중요할 것이다. 하지만 재난 현장이 위치 파악이 힘들거나 구조대가 투입되지 못하는 상태로 악화된다면 대안을 필요로 하게 될 것이다. 따라서 Guru-Guru Inc.의 ROBO lab에서는 위의 작업을 수행할 수 있는 이동형 로봇을 제어 하기 위한 **Mobile Robot Controller(이하 MOROCON)**를 만들고자 한다.

1.1 프로젝트 개요

본 프로젝트의 제어 시스템은 서버, 클라이언트, DB로 구성된다. 서버는 DB를 포함하고 있으며, 클라이언트에서 요청 시 정보를 보내고, 클라이언트에서 보내는 정보를 DB에 저장하고, UI를 통해 사용자에게 정보를 보여준다. 클라이언트는 서버에서 정보를 받아 로봇을 이동시키고, 로봇이 이동할 때 마다 로봇의 현재 위치와 탐색 정보를 서버로 보낸다. 데이터베이스는 이 시스템에서 필요한 정보(재난 상황, 지도 + 위험/중요 지점, 로봇 정보 + 현재 위치)를 가지고 있다. 사용자는 웹 어플리케이션을 통해 로봇의 탐색 범위를 정할 수 있다. 로봇이 이동을 시작하면 어플리케이션에서 로봇의 이동 경로, 로봇의 현재 위치, 로봇이 탐색하고 있는 현장의 지도 정보를 실시간으로 확인 할 수 있다.

1.2 프로젝트 산출물

항목	내용	기타
프로그램	MOROCON 프로그램	
문서	소프트웨어 개발 계획서	1부
	요구 분석서	1부
	설계서	1부
	소프트웨어 사용 매뉴얼	1부

1.3 용어 정의

- **UI** : 일반 사용자들이 컴퓨터 시스템 또는 프로그램에서 데이터 입력이나 동작을 제어하기 위하여 사용하는 명령어 또는 기법을 말한다. 사용자가 컴퓨터나 프로그램과 의사소통을 하고 쉽고 편리하

게 사용할 수 있도록 하는 것이 목적이다.

- **서버** : 일반적으로 서버 프로그램이 실행되고 있는 컴퓨터 하드웨어를 서버라고 부르며 다른 프로그램에게 서비스를 제공하는 컴퓨터 프로그램을 말하기도 한다.
- **구글 앱 엔진** : 구글 앱 엔진은 구글에서 개발한 웹 애플리케이션의 개발과 호스팅을 위한 PaaS(Platform as a Service) 클라우드 컴퓨팅 플랫폼이다.
- **PaaS(Platform as a Service)** : 서비스로서의 플랫폼이라는 뜻으로, 표준화된 플랫폼을 제공하는 서비스다. 이는 가상화된 하드웨어와 소프트웨어 등을 필요에 따라 제공하며, 개발과 관련된 환경 및 프로세스를 제공한다.
- **데이터 스토어** : PaaS 환경에서 데이터가 저장되어 관리되는 곳을 뜻한다.
- **GQL** : GQL은 SQL과 비슷한 언어로써 구글 앱 엔진 데이터스토어에 엔티티와 키를 통해 접근하기 위한 언어이다. GQL은 전통적인 데이터베이스의 쿼리 언어와 다른 점이 있으나 전반적인 문법은 SQL과 비슷하다.
- **이클립스(Eclipse)** : 이클립스는 다양한 플랫폼에서 쓸 수 있으며, 자바를 비롯한 다양한 언어를 지원하는 프로그래밍 통합 개발 환경을 목적으로 시작하였으나, 현재는 OSGi를 도입하여, 범용 응용 소프트웨어 플랫폼으로 진화하였다.
- **파이썬** : 파이썬(Python)은 1991년 귀도 반 로섬(Guido van Rossum)이 발표한 고급 프로그래밍 언어로, 플랫폼 독립적이며 인터프리터식, 객체지향적, 동적 타이핑(dynamically typed) 대화형 언어이다.
- **Git hub** : 깃허브는 깃을 사용하는 프로젝트를 지원하는 웹 기반의 호스팅 서비스이다. 루비 온 레일스로 작성되었다. GitHub는 영리적인 서비스와 오픈소스를 위한 무상 서비스를 모두 제공한다.
- **Git** : 깃은 프로그램 등의 소스 코드 관리를 위한 분산 버전 관리 시스템이다. 빠른 수행 속도에 중점을 두고 있는 것이 특징이다. 최초에는 리누스 토르발스가 리눅스 커널 개발에 이용하려고 개발하였으며, 현재는 다른 곳에도 널리 사용되고 있다.
- **아키텍처** : 아키텍처는 비즈니스 요구 사항을 만족하는 시스템을 구축하기 위해서 전체 시스템에 대한 구조를 정의한 문서로, 시스템을 구성하는 컴포넌트와, 그 컴포넌트간의 관계, 그리고, 컴포넌트가 다루는 정보를 정의한다.

2. 자원 및 조직 구성

2.1 자원

가. 인력

- 서울시립대학교 컴퓨터과학부생 3명 -김영훈, 안주영, 조한주

나. 비용

- 소요 기간 : 총 7주
- 주당 작업일 수 : 5일
- 직접 인건비 = (118,732*2+190,787*1)원 * 35일 = 14,988,785 원

※ 한국소프트웨어 산업협회가 공표한 2015년 9월 1일 적용 SW기술자 평균임금에 의거하여 초급기능사 118,732원(2명), 초급기술자 190,787원(1명) 적용

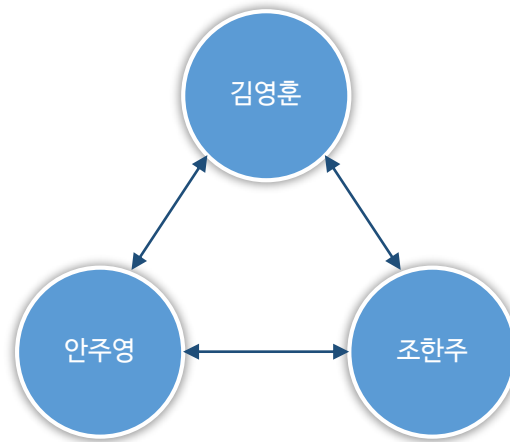
【단위 : 명, 원】

구분	2015년 조사인원	평균임금(일평균)		평균임금(월평균)	평균임금(시간평균)
		2014년	2015년		
기술사	379	408,995	411,642	8,644,482	51,455
특급기술자	17,276	376,262	373,593	7,845,453	46,699
고급기술자	9,361	272,075	276,160	5,799,360	34,520
중급기술자	10,267	221,371	221,375	4,648,875	27,672
초급기술자	15,385	189,174	190,787	4,006,527	23,848
고급기능사	172	172,384	177,337	3,724,077	22,167
중급기능사	320	140,531	141,168	2,964,528	17,646
초급기능사	229	116,756	118,732	2,493,372	14,842
자료입력원	146	111,487	112,570	2,363,970	14,071

〈2015년도 SW기술자 평균임금〉

2.2 조직 구성

본 프로젝트의 팀 구성은 **에고레스(Egoless)** 팀 구성으로 한다. 에고레스 팀 구성은 작업 만족도를 높이고 의사 교류를 원활히 하여 본 프로젝트와 같은 소규모 프로젝트에서의 생산성을 높일 수 있다. 또 전체 구성원이 동등한 레벨에 있어 다른 사람의 일을 함께 검토하고 이해할 수 있는 민주적 팀 구성이다.



〈조직 구성도〉

2.3 작업 분할

각 팀원들은 아래와 같이 업무를 분담하여 수행한다. 그러나 에고레스 형 팀 구성에 맞춰 다른 팀원의 업무도 공동의 일로 간주하여 활발히 의사소통하고 대략적인 업무를 숙지한다.

구성원	업무 내용
김영훈	서버 핵심 기능(로봇의 이동 경로 제어) 구현
안주영	DB 구축, 서버와 클라이언트의 통신부 구현
조한주	웹 서버 UI 설계 및 구현, 클라이언트 기능 구현

3. 위험 관리

위험 내용	심각성	발생 확률	위험 감시 및 조치
일정의 지연	심각	높음	일정을 계획할 때 실현 가능하도록 계획하고, 최대한 일정에 맞게 프로젝트를 진행하되, 만일 일정이 지연된다면 일정을 조정한다.
중간/기말 시험 준비	보통	낮음	시험 일정에 맞추어 미리 시험을 준비하여 프로젝트에 차질이 생기지 않도록 한다.
팀원의 질병	심각	낮음	팀원이 아플 경우 해당 임무를 대체 할 수 있도록 평상시에 다른 팀원의 역할도 이해하도록 한다.
팀원의 프로그래밍 미숙	심각	높음	경험 있는 프로그래머에게 자문을 구하거나 전문 서적을 참고한다.
하드웨어의 극심한 손상	매우 심각	매우 낮음	만약에 있을 천재지변이나 기기의 파손을 대비하여 수시로 진행상황을 백업 해둔다.

4. 하드웨어와 소프트웨어 요구사항

4.1 하드웨어(권장)

범주	환경
OS	Windows XP 이상
CPU	Intel i3 이상
RAM	2GB 이상

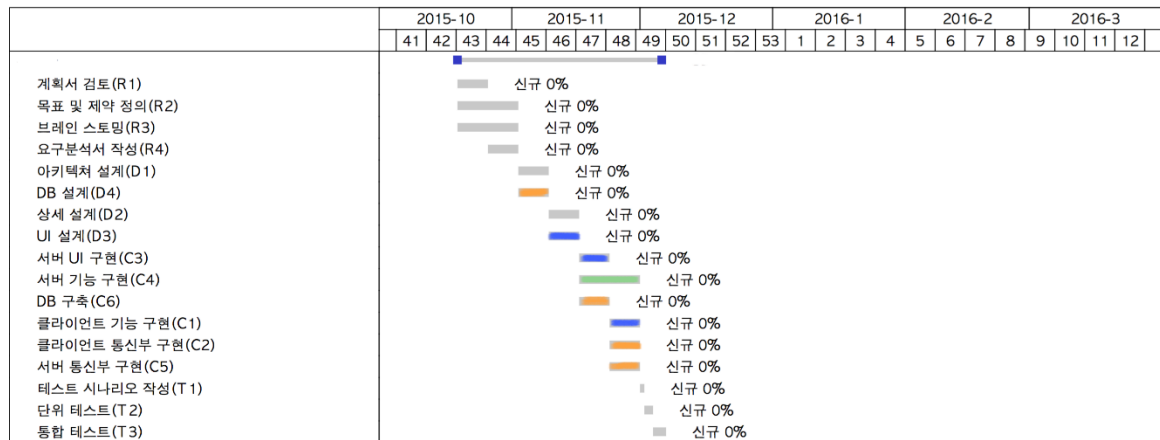
4.2 소프트웨어

범주	환경
Server	Python - Eclipse - Google App Engine
Client	Python - Eclipse
Database	Python - Eclipse - Google App Engine Data Store
Web UI	HTML5

5. WBS



6. 일정 관리



7. 보고 및 감시 체계

7.1 회의 일정

- 매주 목요일 4시에 정보 기술관이나 중앙 도서관 스터디룸에서 실시하는 것을 원칙으로 한다.
- 계획에 따라 단계별 작업을 마친 후 전원 참석 하에 실시한다.
- 만약 계획에 비해 일정이 지연될 시에는 유동적으로 추가 회의를 실시한다.

7.2 정보 공유 방법

- 공유할 가치가 있는 정보를 찾았을 경우 카카오톡 채팅 서비스를 통해 공유하거나, 인쇄물을 회의에 서 배부한다.
- 개발 단계에서 오류가 발생되면 이를 문서화하여 팀원에게 공유한다.
- 산출물과 소스코드는 버전 관리 기능을 제공하는 Git과 Git Hub를 이용한다.