

# 소프트웨어 설계



과목	소프트웨어 설계
학과	컴퓨터공학과
담당교수	장성진 교수님
학번 및 성명	20153128 전준영
	20121981 김동욱
	20153135 차시은

# 목차

1. 서론
  - 1.1. 프로젝트 개요
  - 1.2. 프로젝트 정의 및 시장 분석
  - 1.3. 주요 일정
  - 1.4. 프로젝트 팀 구성
  - 1.5. 개발 계획
2. 개발 규모 산정
  - 2.1. 규모 산정
  - 2.2. WBS
  - 2.3. 일정 계획(간트차트)
3. 투입 공수
4. 마일스톤 및 일정 계획
5. 데이터 관리 계획
6. 프로젝트 계획 TABLE
7. 참고 및 문헌

## 1. 서론

### 1.1 프로젝트 개요

- 목적: 주차 차량 관리 어플리케이션
- 개발 기간: 2017년 3월 ~ 2017년 6월 13일
- 개발 환경: Window 10, MySQL, JDK 1.8, Android Studio
- 개발 언어: JAVA
- 개발 방법론: 객체지향 프로그래밍
- 개발 생명주기: V-모델
- 기능 소개: 관리자 - 등록된 유저와 차량을 관리한다.  
                  사용자 - 주차장을 이용하기 위해 자신의 차량을 등록하고 사용한다.  
                  기능 - 회원 가입, 회원 로그인, 회원 탈퇴, 회원 수정, 주차 입차, 주차 출차,  
                  주차 공간 관리, 요금 계산, 시간 계산, 차량 인식, 차량 파악
- 팀 구성: 전준영, 김동욱, 차시은
- 개발 도구: Eclipse
- 관리 도구: Bitbucket, Git

### 1.2. 프로젝트 정의 및 시장 분석

#### 주차 차량 관리 프로그램 - 주차 관제 시스템

이용자는 스마트폰을 통해 해당 어플리케이션을 이용하여 주차장을 들어갈 때 입장 인식을 한 뒤에 들어간다. 나갈 때에는 다시 재인식을 하고 요금을 계산하고 나가도록 한다. 관리자는 이에 대한 전반적인 주차 시설에 대한 관리를 한다.

주요 특징적인 기능으로는 회원 가입, 로그인 등 회원에 대한 관리 메뉴가 있으며 관제 시스템에 대한 차량 관리, 요금 계산 등이 있다.

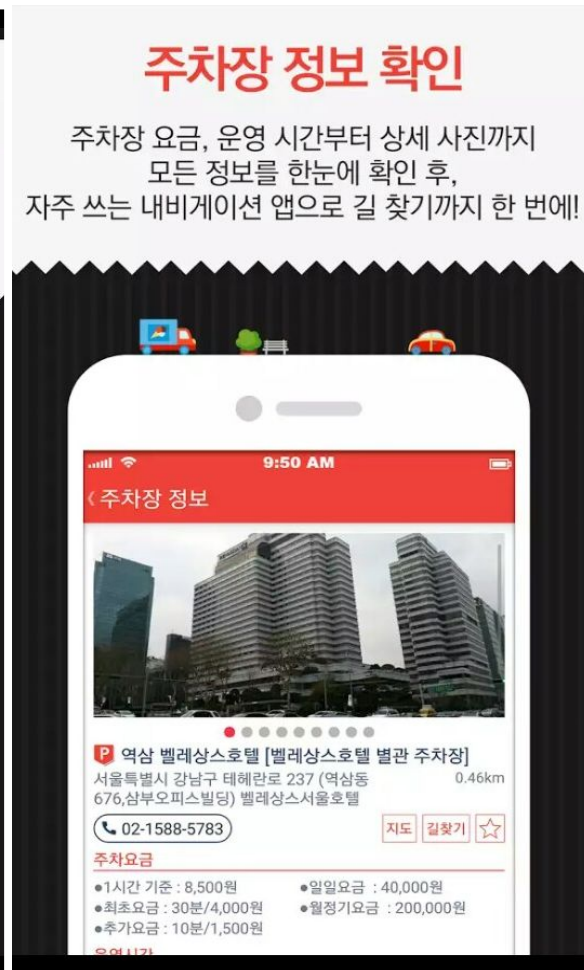
주차 관제 시스템 이전에는 주로 PDA 단말기를 이용하여 직접 요금을 받았다. 그러나 이러한 방법은 많은 문제점이 발생하였다. 주차원 혼자 넓은 면적의 주차장의 출입 차량을 관리하기가 어려웠고 이에 따라 요금 미부과 차량이 증가하고, 정확한 시간 적용이 어려워 요금징수에 차질이 빚는 경우가 많이 발생하였다.

이러한 여러가지 문제점 때문에 최근에는 여러 지역에서 관제 시스템을 도입하기 시작하였다. 시흥시의 경우에는 오는 4월 1일부터 시흥시체육관 부설 주차장에 유인 주차관제 시스템을 도입하기로 결정하였고 3월 현재 시범 운영 중에 있으며 사용 중 발생하는 문제점들을 보완하여 내달 1일부터 정식 운영할 예정이다.

전주시시설관리공단의 경우에는 현재 유인으로 운영중인 공영 및 부설주차장 8개소에 무인주차요금징산시스템을 추가로 도입해 오는 9월부터 운영할 예정이다.

이렇듯 기존에 가지고 있는 주차 관리 시스템의 문제점을 해결하기 위해 이용시간이나 금액에 대한 주차요금의 투명화가 가능해지고 부당요금 징수나 주차요금관련 충돌을 줄일 수 있고 이용도 간편한 관제 시스템으로 교체하고 있는 추세이다.

시중에도 물론 이러한 어플리케이션이 존재한다.



어플리케이션 아이파킹은 타 주차장과 협력하여 지역 내에 주차장의 위치를 확인할 수 있고 해당 주차장의 정보(요금, 운영시간 등)을 확인할 수 있다. 또한 어플리케이션이 설치되어 있으면 와이파이로 연동하여 주차장 진입 시에 원활하게 통과할 수 있다. 또한 음식점이나 주차장과 제휴를 맺어서 주차 요금을 할인해주는 등의 기능이 있다.

이 외의 주차에 도움을 주는 다양한 어플리케이션이 있으며 계속해서 생겨나는 상황이다.

이에 따라 본 프로그램은 기존의 주차 시스템에 대한 문제점을 탈피하고 편의를 제공하고자 스마트폰 어플리케이션과 NFC 기술을 이용하여 관제 시스템을 만드는데 목적을 두고 있다.

### 1.3 주요 일정

구분	제출물명	주요 내용	제출일정	비고
계획 및 분석	프로젝트 제안서	프로젝트에 대한 목표와 범위를 정의하고 일정과 업무 분담에 대한 개요에 대해 작성한다.	17.03.23	
	프로젝트 계획서	프로젝트 개발의 전과정에 걸쳐 개발자가 수행해야 하는 활동 및 산출물에 대한 명세를 작성한다.	17.03.30	
	요구사항 명세서	고객의 요구사항을 바탕으로 개발할 기능에 대한 명세를 작성한다.	17.04.06	
	테스트 케이스	각 개발 단계에서 개발자가 수행해야 할 테스트 케이스를 기록하여 정리한다.	17.04.06	
	테스트 계획서	테스트 일정에 대한 계획에 대한 명세를 작성한다.	17.04.06	
설계	시스템 설계	있거나 변경 사항이 있다면 요사용자 요구사항에 대해 구현 가능성과 필요한 기술들을 파악한다. 요구 사항 중 불가능한 경우가 구사항 문서를 수정한다.	17.04.28	
	아키텍처 설계	모든 구현될 모듈 항목과 그 간략한 기능을 정의하고, 모듈 간의 인터페이스, 관계, 의존성을 기술하며, 필요한 데이터베이스 테이블, 아키텍처 다이어그램, 적용기술 내역을 기술한다	17.04.28	
	모듈 설계	아키텍처 설계 단계에서 정의된 모듈 항목을 더 세분하여 각각의 모듈에 대한 기술을 작성하며, 이를 이용하여 프로그래머들이 코딩을 시작할 수 있도록 만들어 준다.	17.04.28	
구현 및 테스트	소스 코드	설계 사양서를 바탕으로 소스 코드를 작성한다.	17.05.30	
	테스트 결과 보고서	작성한 코드에 대한 테스트 결과에 대한 명세를 작성한다.	17.06.09	
	인수 테스트 결과 보고서	인수 테스트를 실시한다.	17.06.09	
	프로젝트 완료	프로젝트를 종료 후, 다음		

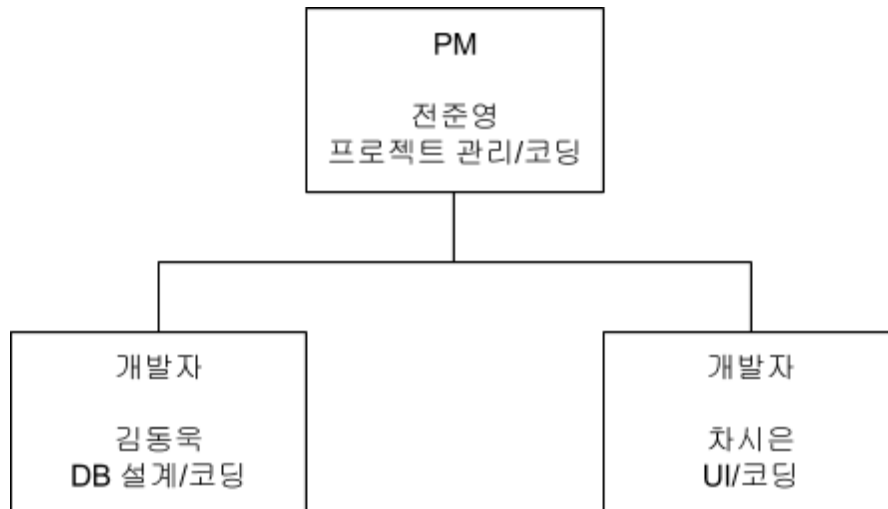
완료	보고서	프로젝트에 참고할 수 있도록 평가하고 기록한다. 테스트 결과 보고서를 포함한다.	17.06.13	
----	-----	--	----------	--

#### 1.4 프로젝트 팀 구성

##### 1.4.1 프로젝트 팀 명단

성명	학년	학번	역할	투입 M/M	비고
전준영	3	20153128	PM 및 코더	1.65	
김동욱	3	20121981	DB 설계 및 코더	1.375	
차시은	3	20153135	UI 디자인 및 코더	1.375	

##### 1.4.2 프로젝트 팀 조직



#### 1.5 개발 계획

##### 1.5.1 개발 생명주기 모델

개발 생명주기: V Model

- V 모델은 폭포수 모델의 확장된 형태 중 하나로 볼 수 있으며, 아래 방향으로 선형적으로 내려가면서 진행되는 폭포수 모델과는 달리, 프로세스는 코딩 단계에서 위쪽으로 꺾여서 알파벳 V자 모양으로 진행된다. V 모델은 개발 생명주기의 각 단계와 그에 상응하는 **소프트웨어 시험** 각 단계의 관계를 보여준다.
- V 모델은 소프트웨어 개발의 각 단계마다 상세한 문서화를 통해 작업을 진행하는 잘 짜인 방법을 사용한다. 또한 테스트 설계와 같은 테스트 활동을 코딩 이후가 아닌 프로젝트 시작 시에 함께 시작하여, 전체적으로 많은 양의 프로젝트 비용과 시간을 감소시킨다.

개발단계	관련요소	내용
계획	입력 산출물	<ul style="list-style-type: none"> <li>소프트웨어 제안 요청서(REP)</li> </ul>
	수행 태스크	<ul style="list-style-type: none"> <li>고객의 요구사항을 문서화한다.</li> <li>고객의 요구사항을 바탕으로 프로젝트 제안서를 작성한다.</li> </ul>
	출력 산출물	<ul style="list-style-type: none"> <li>프로젝트 제안서</li> </ul>
	시작 기준	<ul style="list-style-type: none"> <li>고객으로부터 요구사항이 전달되어야 한다.</li> <li>요구사항 수집 활동에 대한 정책이 수립되어 있어야 한다.</li> <li>프로젝트 제안서가 문서화되고 승인되어야 한다.</li> <li>요구사항 수집 및 프로젝트 제안서 작성 활동에 대한 담당자 및 책임이 할당되어 있어야 한다.</li> <li>요구사항 수집 및 프로젝트 제안서 작성 활동을 지원하기 위한 도구가 제공되고 이용 가능해야 한다.</li> <li>요구사항 수집 및 프로젝트 제안서 작성 활동을 위한 교육 훈련 및 오리엔테이션이 계획되고 작성되어야 한다.</li> </ul>
	완료 기준	<ul style="list-style-type: none"> <li>고객의 요구사항이 수집되고 기록되어 있어야 한다.</li> <li>프로젝트 제안서와 고객 및 이해 관계자에게 전달되어야 한다.</li> </ul>
분석	입력 산출물	<ul style="list-style-type: none"> <li>프로젝트 제안서</li> </ul>
	수행 태스크	<ul style="list-style-type: none"> <li>프로젝트 제안서를 바탕으로 요구사항 명세서를 작성한다. <ul style="list-style-type: none"> <li>요구사항 분석</li> <li>유스케이스 작성</li> <li>기능, 비기능 요구사항 분석</li> <li>UI 프로토타입 작성</li> <li>요구사항 명세서 작성</li> </ul> </li> <li>요구사항 명세서를 바탕으로 프로젝트 계획서를 작성한다. <ul style="list-style-type: none"> <li>상세 WBS 작성 및 자원 할당</li> <li>팀 조직 계획 수립</li> <li>생명주기 및 개발방법론 결정</li> <li>일정 계획 수립</li> <li>산정 및 공수 분석</li> <li>데이터 및 컴퓨터 자원 관리 계획 수립</li> <li>위험 관리 계획 수립</li> <li>프로젝트 계획서 작성</li> </ul> </li> </ul>

## 1.5.2 개발 방법론 및 도구 선정

### 1.5.2.1 개발 방법론

- 객체지향 방법론을 이용한다.

현실에서 표현되는 정보나 데이터들을 프로그램에서 객체로 표현하여 모델링이 적합하며 구현에 용이하기 때문에 해당 방법론을 사용한다.

### 1.5.2.2 객체지향 방법론 절차

개발 단계	상세 내용
객체지향 분석	해당 어플리케이션을 사용하는 유저와 관리자 그리고 어플리케이션에 대한 기능을 UML 언어의 Use Case와 Sequence Diagram을 이용하여 시스템 요구사항을 분석하고, ER 다이어그램과 테이블 목록 및 기술을 하여 데이터베이스 요구사항을 분석한다.
객체지향 설계	시스템 설계 및 객체 설계를 분리하여 설계한다.  - 객체 설계 - UML 언어의 Class Diagram을 이용 class를 정의하고 class들 간의 관계 및 알고리즘 그리고 자료구조를 설계한다.  - DB 설계 - ER 다이어그램을 이용하여 테이블을 상세 기술한다.
객체지향 프로그래밍	객체지향 설계를 바탕으로 Java 프로그래밍 언어를 이용하여 데이터 구조 부분과 관련 메소드를 포함하는 여러 객체들로 구현한다.

## 2. 개발 규모 산정

### 2.1 규모 산정

#### 2.1.1 LOC

- 모듈 목록

모듈 번호	설명
1	회원 관리 모듈
2	차량 관리 모듈
3	요금 계산 모듈
4	시간 계산 모듈

- 각 모듈별 규모 추정

모듈번호	낙관적 LOC	보통의 LOC	비관적 LOC	추정 LOC
1	400	500	650	508
2	520	590	770	581
3	440	540	700	550



4	100	130	300	153
추정 LOC 합계				1,792

- 경험적 데이터  
 생산성 : 600 LOC/ Man-Month  
 LOC당 비용 : 2,500원

## 2.1.2 COCOMO

- 산정

프로젝트	a	b	c	d
기본형	2.4	1.05	2.5	0.38
중간형	3.0	1.12	2.5	0.35
내장형	3.6	1.20	2.5	0.32

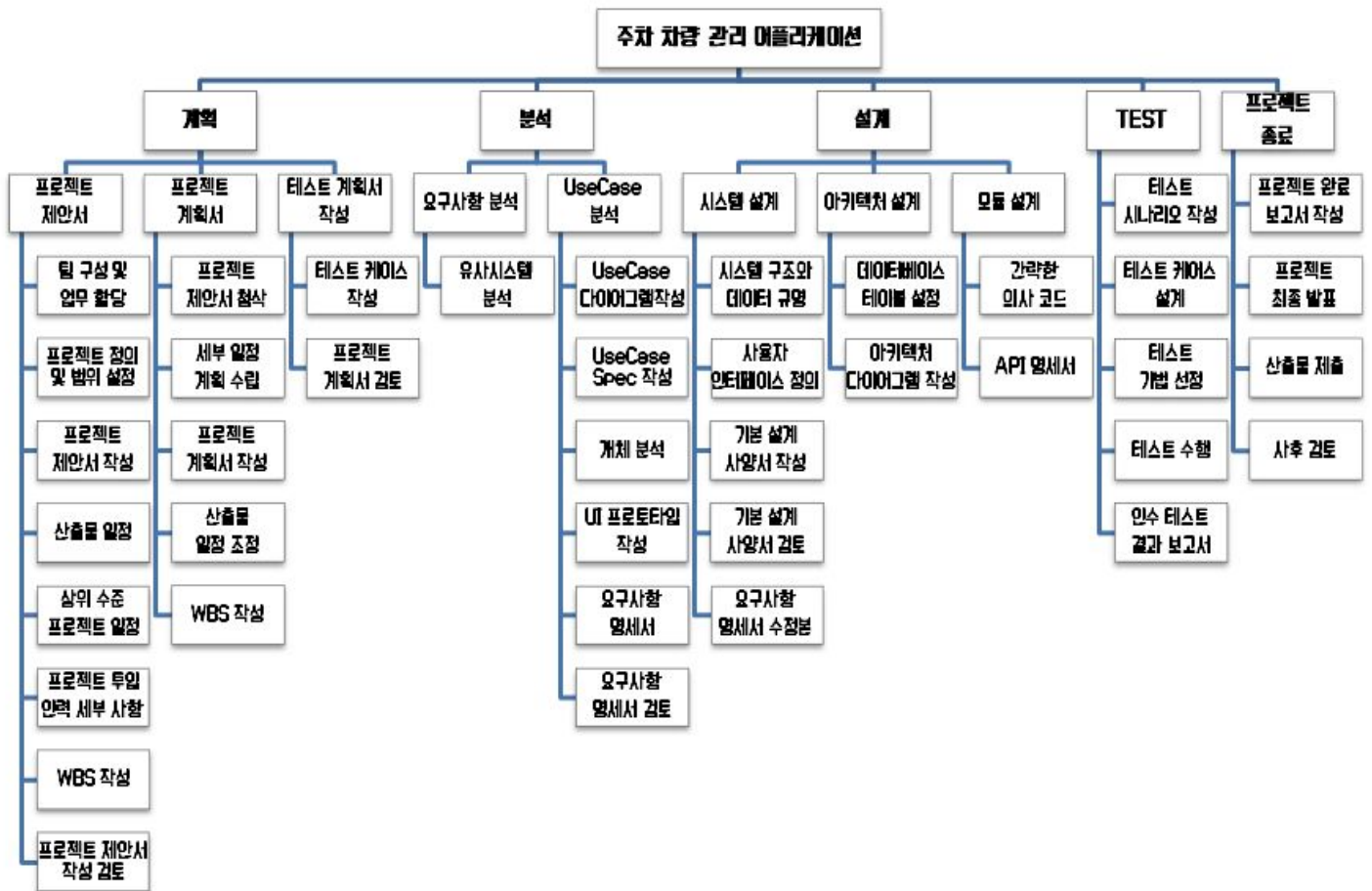
<기본 COCOMO의 프로젝트 유형별 상수값>

- 개발 프로젝트  
 프로젝트 유형 : 기본형  
 예상 규모 : 1,792 LOC ≒ 1.8 KLOC

- 개발 노력  $E = 2.4 \times (KLOC)^{1.05}$   
 $= 2.4 \times (1.8)^{1.05}$   
 $\approx 4.4 \text{ M/M (Man-Month)}$

-개발 기간  $D = 2.5 \times (E)^{0.38}$   
 $= 2.5 \times (4.4)^{0.38}$   
 $\approx 4.38 \text{ M (Month)}$

## 2.2 WBS



## 2.3 일정 계획 (간트 차트)

작업	추진 일정(11주)										
	1	2	3	4	5	6	7	8	9	10	11
프로젝트 제안/계획 프로젝트 제안서 작성 WBS 작성											
프로젝트 계획서 테스트 계획서 작성 요구사항 분석 UseCase 분석											
시스템 설계 아키텍처 설계 모듈 설계											
구현 소스코드 작성 단위 테스트 테스트 결과 보고서											
테스트 테스트 시나리오 작성 테스트 수행 테스트 결과 보고서											
프로젝트 종료 완료 보고서 작성 발표											

### 3. 투입 공수

※M/M(Man-Month)  
단위

개발 단계	계획 (10%)	분석 (15%)	설계 (35%)	구현 (30%)	프로젝트 완료 (10%)	합계 (100%)
투입 공수	0.44	0.66	1.54	1.32	0.44	4.40

#### 4. 마일스톤 및 일정 계획

마일스톤		일정 계획
계획 단계	요구사항 정의	2017. 03.20 ~ 2017. 03. 23
	프로젝트 정의 및 범위 설정	2017. 03.20 ~ 2017. 03. 23
	팀 구성 및 업무 할당	2017. 03.20 ~ 2017. 03. 23
	일정	2017. 03.20 ~ 2017. 03. 23
	WBS 작성	2017. 03.20 ~ 2017. 03. 23
	프로젝트 제안서 작성	2017. 03.20 ~ 2017. 03. 23
분석 단계	요구사항 분석	2017. 03.23 ~ 2017. 03. 30
	유스케이스이스 작성	2017. 03.23 ~ 2017. 03. 30
	기능, 비기능 요구사항 분석	2017. 03.23 ~ 2017. 03. 30
	UI 프로토타입 작성	2017. 03.23 ~ 2017. 03. 30
	요구사항 명세서 작성	2017. 03.23 ~ 2017. 03. 30
	상세 WBS 작성	2017. 03.23 ~ 2017. 03. 30
	팀 조직 계획	2017. 03. 31
	개발 계획 및 마일스톤 및 일정 계획	2017. 03. 31
	산정 및 공수, 비용	2017. 03. 31
	데이터 및 컴퓨터 자원 관리 계획	2017. 03. 31

#### 5. 데이터 관리 계획

5.1 Bitbucket + git 프로그램을 사용한다.

5.2 Bitbucket 저장소 목록

- source: 소스 코드 및 DB 스키마 저장 관리

5.3 관리자: 전준영

#### 5.4 서버 자원

서버	자원명	비고
DB 서버	CPU: i5 RAM: 8GB	유저 정보 및 차량 정보 DB 서버
저장소 서버	-	버전 관리 서버

## 6. 프로젝트 계획 TABLE

	①	이름	기간	시작	종료	
1		주차 차량관리 어플리케이션	62 days	17. 3. 20 오전 8:00	17. 6. 13 오후 5:00	
2		▣계획 및 분석	62 days	17. 3. 20 오전 8:00	17. 6. 13 오후 5:00	
3		▣프로젝트 제안서	4 days	17. 3. 20 오전 8:00	17. 3. 23 오후 5:00	
4		팀 구성 및 업무 할당	4 days	17. 3. 20 오전 8:00	17. 3. 23 오후 5:00	
5		프로젝트 정의 및 범위 설정	4 days	17. 3. 20 오전 8:00	17. 3. 23 오후 5:00	
6		프로젝트 제안서 작성	4 days	17. 3. 20 오전 8:00	17. 3. 23 오후 5:00	
7		산출물 일정	4 days	17. 3. 20 오전 8:00	17. 3. 23 오후 5:00	
8		상위 수준 프로젝트 일정	4 days	17. 3. 20 오전 8:00	17. 3. 23 오후 5:00	
9		프로젝트 투입 인력 세부 사항	4 days	17. 3. 20 오전 8:00	17. 3. 23 오후 5:00	
10		WBS작성	4 days	17. 3. 20 오전 8:00	17. 3. 23 오후 5:00	
11		프로젝트 제안서 작성 검토	4 days	17. 3. 20 오전 8:00	17. 3. 23 오후 5:00	
12		▣프로젝트 계획서	6 days	17. 3. 23 오전 8:00	17. 3. 30 오후 5:00	
13		프로젝트 제안서 첨삭	6 days	17. 3. 23 오전 8:00	17. 3. 30 오후 5:00	
14		세부 일정 계획 수립	6 days	17. 3. 23 오전 8:00	17. 3. 30 오후 5:00	
15		프로젝트 계획서 작성	6 days	17. 3. 23 오전 8:00	17. 3. 30 오후 5:00	
16		산출물 일정 조정	6 days	17. 3. 23 오전 8:00	17. 3. 30 오후 5:00	
17		WBS작성	6 days	17. 3. 23 오전 8:00	17. 3. 30 오후 5:00	
18		▣테스트 계획서 작성	6 days	17. 3. 23 오전 8:00	17. 3. 30 오후 5:00	
19		테스트 케이스 작성	6 days	17. 3. 23 오전 8:00	17. 3. 30 오후 5:00	
20		프로젝트 계획서 검토	6 days	17. 3. 23 오전 8:00	17. 3. 30 오후 5:00	
21		▣요구사항 분석	6 days	17. 3. 30 오전 8:00	17. 4. 6 오후 5:00	
22		유사시스템분석	6 days	17. 3. 30 오전 8:00	17. 4. 6 오후 5:00	
23		▣UseCase 분석	6 days	17. 3. 30 오전 8:00	17. 4. 6 오후 5:00	
24		UseCase 다이어그램 작성	6 days	17. 3. 30 오전 8:00	17. 4. 6 오후 5:00	
25		UseCase Spec 작성	6 days	17. 3. 30 오전 8:00	17. 4. 6 오후 5:00	
26		▣개체 분석	6 days	17. 3. 30 오전 8:00	17. 4. 6 오후 5:00	
27		UI 프로토타입 작성	6 days	17. 3. 30 오전 8:00	17. 4. 6 오후 5:00	
28		요구사항 명세서 작성	6 days	17. 3. 30 오전 8:00	17. 4. 6 오후 5:00	
29		요구사항 명세서 검토	6 days	17. 3. 30 오전 8:00	17. 4. 6 오후 5:00	
30		▣설계	16 days	17. 4. 7 오전 8:00	17. 4. 28 오후 5:00	
31		▣시스템설계	16 days	17. 4. 7 오전 8:00	17. 4. 28 오후 5:00	
32		시스템 구조와 데이터 규명	16 days	17. 4. 7 오전 8:00	17. 4. 28 오후 5:00	
33		사용자 인터페이스 정의	16 days	17. 4. 7 오전 8:00	17. 4. 28 오후 5:00	
34		기본 설계 사양서 작성	16 days	17. 4. 7 오전 8:00	17. 4. 28 오후 5:00	
35		기본 설계 사양서 검토(Tes	16 days	17. 4. 7 오전 8:00	17. 4. 28 오후 5:00	
36		요구사항 명세서 수정본	16 days	17. 4. 7 오전 8:00	17. 4. 28 오후 5:00	
37		▣상세 설계	16 days	17. 4. 7 오전 8:00	17. 4. 28 오후 5:00	
38		▣아키텍처 설계	16 days	17. 4. 7 오전 8:00	17. 4. 28 오후 5:00	
39		데이터베이스 테이블 설계	16 days	17. 4. 7 오전 8:00	17. 4. 28 오후 5:00	
40		아키텍처 다이어그램 작성	16 days	17. 4. 7 오전 8:00	17. 4. 28 오후 5:00	
41		▣모듈 설계	16 days	17. 4. 7 오전 8:00	17. 4. 28 오후 5:00	
42		간략한 의사 코드	16 days	17. 4. 7 오전 8:00	17. 4. 28 오후 5:00	
43		API명세서	16 days	17. 4. 7 오전 8:00	17. 4. 28 오후 5:00	
44		▣구현	22 days	17. 5. 1 오전 8:00	17. 5. 30 오후 5:00	
45		소스 코드 작성	22 days	17. 5. 1 오전 8:00	17. 5. 30 오후 5:00	
46		단위 테스트	22 days	17. 5. 1 오전 8:00	17. 5. 30 오후 5:00	
47		통합 테스트	22 days	17. 5. 1 오전 8:00	17. 5. 30 오후 5:00	
48		시스템 테스트 결과 보고서	22 days	17. 5. 1 오전 8:00	17. 5. 30 오후 5:00	
49		▣TEST	7 days	17. 6. 1 오전 8:00	17. 6. 9 오후 5:00	
50		테스트 시나리오 작성	7 days	17. 6. 1 오전 8:00	17. 6. 9 오후 5:00	
51		테스트 케이스 설계	7 days	17. 6. 1 오전 8:00	17. 6. 9 오후 5:00	
52		테스트 기법 선정	7 days	17. 6. 1 오전 8:00	17. 6. 9 오후 5:00	
53		테스트 수행	7 days	17. 6. 1 오전 8:00	17. 6. 9 오후 5:00	
54		인수 테스트 결과 보고서	7 days	17. 6. 1 오전 8:00	17. 6. 9 오후 5:00	
55		▣프로젝트 종료	2 days?	17. 6. 10 오전 8:00	17. 6. 13 오후 5:00	
56		프로젝트 완료 보고서 작성	2 days	17. 6. 10 오전 8:00	17. 6. 13 오후 5:00	
57		프로젝트 최종 발표	2 days	17. 6. 10 오전 8:00	17. 6. 13 오후 5:00	
58		산출물 제출	2 days	17. 6. 10 오전 8:00	17. 6. 13 오후 5:00	
59		사후 검토	2 days	17. 6. 10 오전 8:00	17. 6. 13 오후 5:00	

## 7. 참고 및 문헌

- 소프트웨어 공학의 소개 - 한혁수 저 (홍릉과학출판사)

- V 모델 - 위키 백과

[https://ko.wikipedia.org/wiki/V\\_%EB%AA%A8%EB%8D%B8](https://ko.wikipedia.org/wiki/V_%EB%AA%A8%EB%8D%B8)