

2020년 2학기 소프트웨어공학 프로젝트

미스터 대박 디너 서비스

팀: 2 MM

(팀장) 컴퓨터과학부 2016920011 김현규

컴퓨터과학부 2016920024 엄현식

최종 작성일: 2020 10월 14일

목차

1.1 개요

- 1.1 프로젝트 개요
- 1.2 프로젝트 산출물

2. 자원 및 비용

- 2.1 자원
- 2.2 비용
- 2.3 기능 점수

3. 일정 계획

- 3.1 WBS
- 3.2 작업 의존 관계
- 3.3 CPM 네트워크
- 3.4 간트 차트

4. 조직 구성 및 인력 배치

- 4.1 조직 구성
- 4.2 직무 기술

5. 기술 관리 방법

- 5.1 변경 관리
- 5.2 위험 관리
- 5.3 비용 및 진도 관리

6. 개발 방법론 및 절차

7. 검토 회의

- 7.1 검토 회의 일정
- 7.2 검토 회의 진행방법
- 7.3 검토 회의 후속조치

8. 개발 환경

9. 성능 시험 방법

- 9.1 UI 테스트
- 9.2 기능 테스트
- 9.3 인수 테스트

10. 문서화

- 10.1 요구분석 및 설계
- 10.2 기술
- 10.3 사용자

11. 유지 보수

- 11.1 정기 보수
- 11.2 긴급 보수
- 11.3 버전 관리

12. 설치, 인수

13. 참고 문헌 및 부록

1. 개요

1.1 프로젝트 개요

본 프로젝트의 명칭은 '미스터 대박 디너 서비스' 라 한다.

프로젝트의 목표는 고객이 원하는 만찬을 집으로 배달하는 서비스를 위한 소프트웨어를 개발하는 것이다.

프로젝트를 통해 고객들은 온라인 주문을 통해 더 편리하게 서비스를 이용할 수 있고, 직원들은 서비스의 운영을 더 효율적으로 할 수 있다.

1.2 프로젝트 산출물

단계	산출물
분석	사용자 요구사항 정의서
	유스케이스 명세서
	요구 모델링 다이어그램
설계	사용자 인터페이스 설계서
	설계 모델링 다이어그램
	패키지 설계도
구현	소스 코드
	데이터베이스 테이블
테스트	테스트 보고서
	최종 실행 파일
	최종 프로젝트 보고서

[표 1.1 프로젝트 산출물]

2. 자원 및 비용

2.1 자원

필요 자원은 개발 인력과 개발 도구 (PC, 노트북)이 해당된다.

해당 프로젝트에 참여 개발 인력은 2명이다.

개발 도구는 개발 인력이 소지하고 있는 프로그래밍용 PC 1대와 노트북 1대이다.

2.2 비용

프로젝트 비용으로는 인건비만을 산정한다.

프로젝트의 기간은 10 월 15 일부터 12 월 4 일까지로 산정하고, 시험 기간인 10 월 25 일 ~ 10 월 30 일을 제외한 총 44 일에 일 평균 2 시간 근무로 2 인 총 176 시간을 계획한다.

시급의 계산은 한국소프트웨어산업협회에서 실시한 2018 년 초급기능사 시간평균임금인 15,119 원으로 계산하여 총 2,660,944원으로 산정하였다.

2.3 기능 점수

	단순	보통	복잡	FP
입력	4	-	-	12
출력	6	2	-	34
질의	1	-	-	3
파일	2	1	-	24
인터페이스	-	-	-	-

[표 2.1 기능 점수 표]

특성	처리 복잡도
신뢰도 높은 백업과 복구	2
데이터 통신	2
분산 처리 기능	0
성능 중요도	2
과부하 운용 환경 실행여부	1
온라인 데이터 입력	3
온라인 입력의 다중 화면 구축을 위한 입력 트랜잭션	0
마스터 파일의 온라인 갱신 필요성	0
입력,출력,파일,질의의 복잡성	1
내부 처리의 복잡성	2
재사용을 위한 설계 필요성	0
변환과 설치 포함 여부	0
다중 사이트 설치 필요성	0
변경과 사용의 용이 여부	1

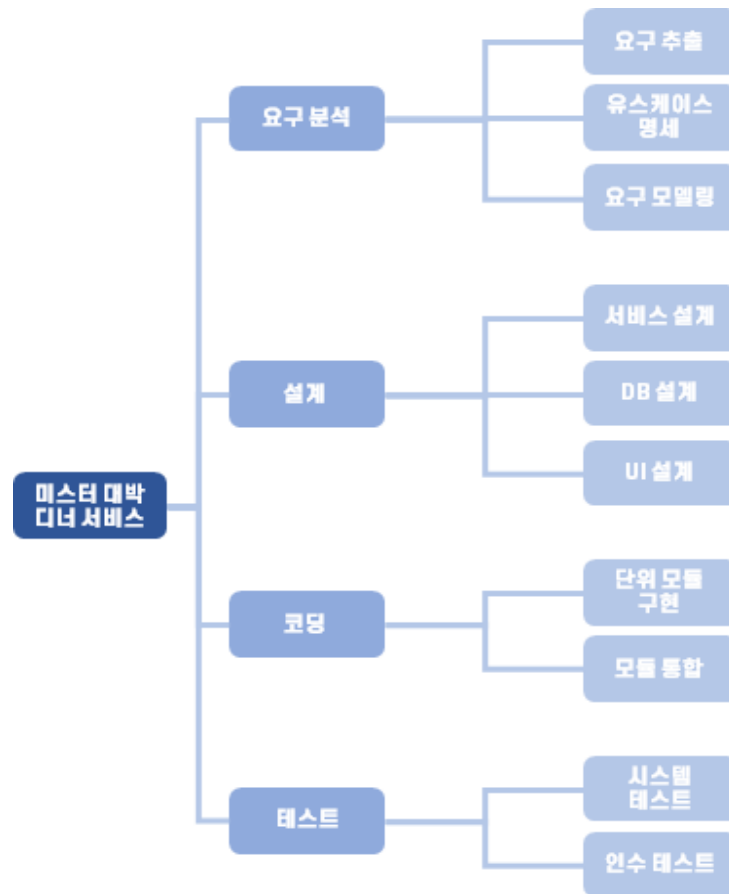
[표 2.2 처리 복잡도 계산표]

GFP (총 기능 점수)	PCA (보정계수)		FP (기능 점수)
63	0.79		49.8
1점당 JAVA 라인 수 (LOC/FP)	총 라인 수	팀의 생산성 (LOC/ 2MM)	예상 총 노력
53	2640	2000	2.6MM

[표 2.3 노력 추정 표]

3. 일정 계획

3.1 WBS



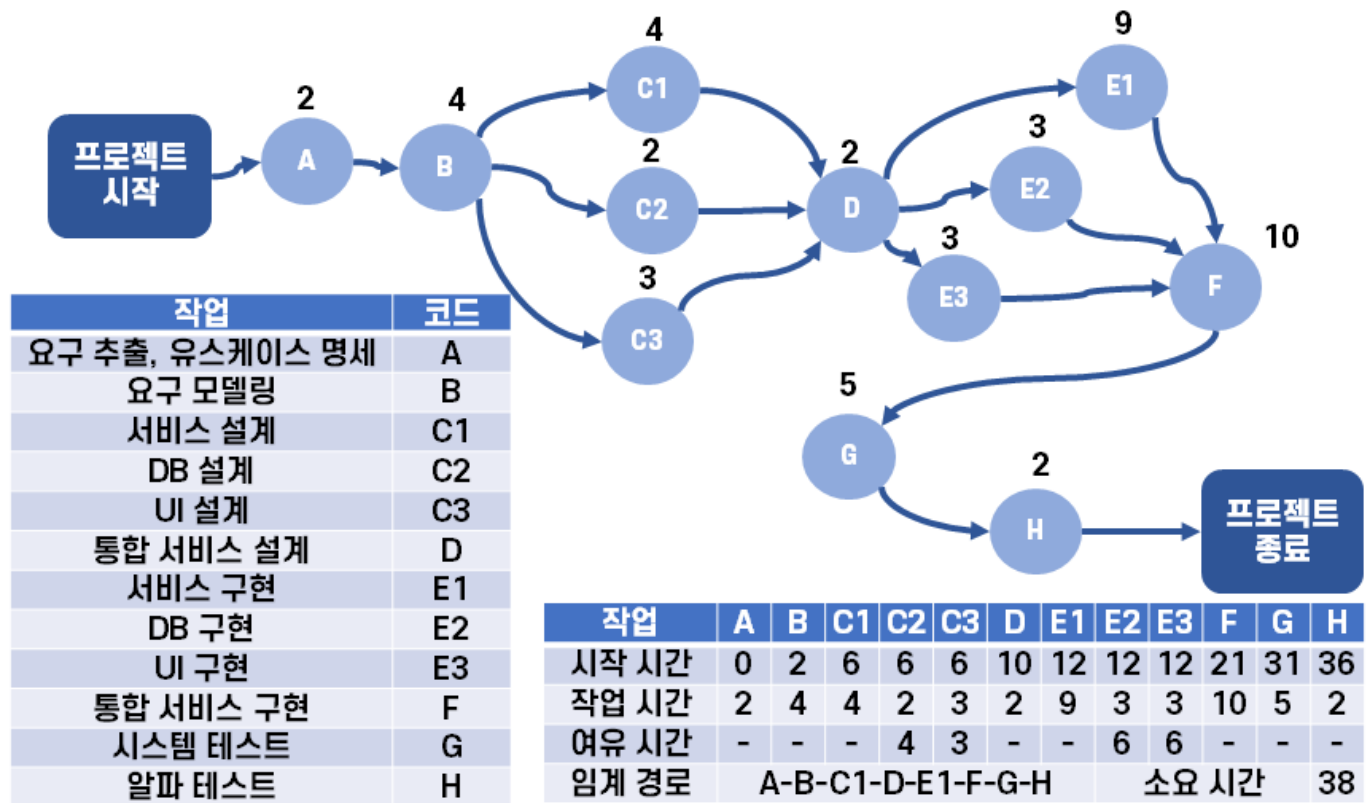
[그림 3.1 WBS]

3.2 작업 의존 관계

작업	선행 작업	소요 기간 (일)
요구 추출, <u>유스케이스 명세</u>	-	2
요구 모델링	요구 추출, <u>유스케이스 명세</u>	4
서비스 설계	요구 모델링	4
DB 설계	요구 모델링	2
UI 설계	요구 모델링	3
통합 서비스 설계	서비스 설계, DB 설계, UI 설계	2
서비스 구현	통합 설계	9
DB 구현	통합 설계	3
UI 구현	통합 설계	3
통합 서비스 구현	서비스 구현, DB 구현, UI 구현	10
시스템 테스트	통합 서비스 구현	5
알파 테스트	시스템 테스트	2

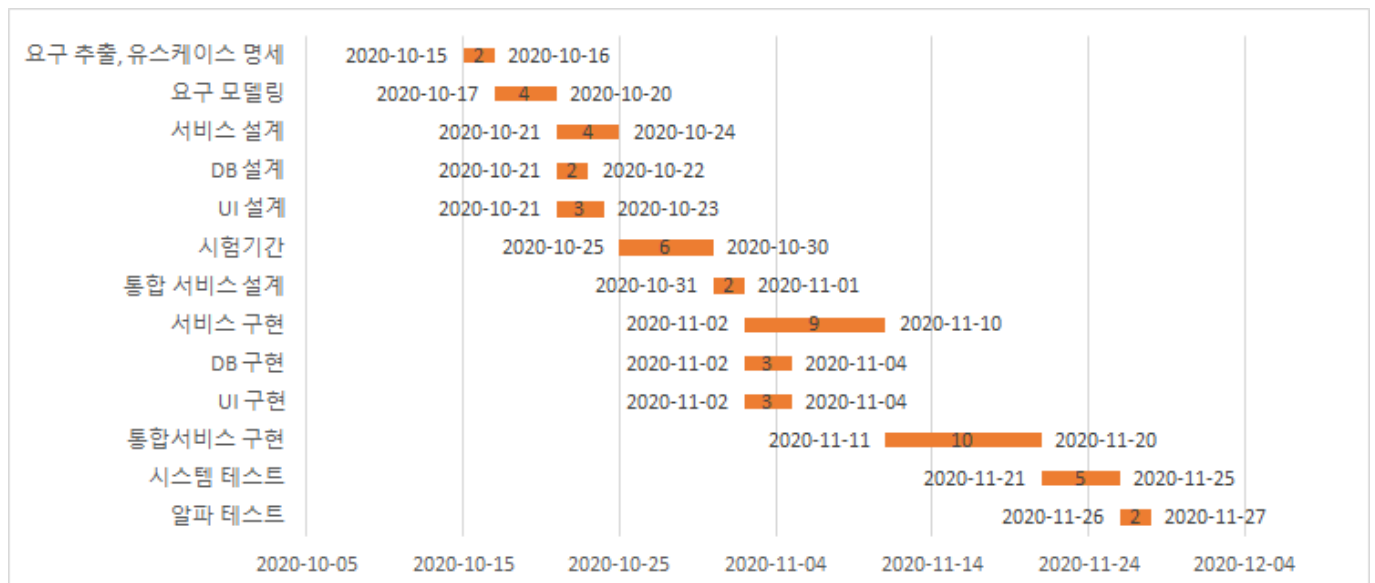
[표 3.1 작업 의존 관계]

3.3 CPM 네트워크



[그림 3.2 CPM 네트워크와 임계 경로]

3.4 간트 차트



[그림 3.3 간트 차트]

개발 프로세스 단계로 선택한 폭포수 모델링에서 요구 분석부터 10/15에 시작하여 최종 테스트까지 11/27로 일정을 계획하였고, 10/25~10/30은 중간고사 시험기간으로 개발 과정에서 제외하였다.

중간 발표는 11/06 이고, 최종 산출물 결과 발표는 12/04 이다.

프로세스 종료는 11/27 일로 위험 상황에 대비한 여유 기간이 총 7 일로 계획되어 있다.

4. 조직 구성 및 인력 배치

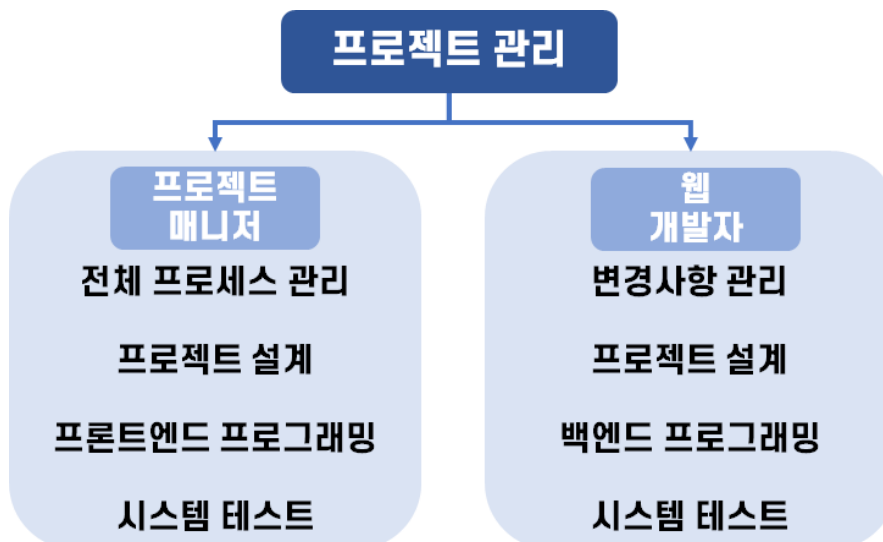
4.1 조직 구성

프로젝트 별 조직 구성 방법으로 팀을 조직한다. 프로젝트별 조직 방법에 따라 프로젝트 매니저와 시스템 개발자로 구성되나 모두 소프트웨어의 전반적인 개발을 담당하고, 프로젝트 진행에 있어 자유롭게 의견을 교환한다. 다만 효율적인 프로젝트 진행을 위해 검토 회의, 일정 관리등의 전체 프로세스를 프로젝트 매니저가 중점적으로 관리하고 변경사항 관리,세부 설계와 구현은 웹 개발자가 중점적으로 담당하여 프로젝트를 진행한다.

직책	프로젝트 매니저	웹 개발자
이름	김현규	엄현식

[표 4.1 조직 구성]

4.2 직무 기술



[그림 4.1 직무 기술]

5. 기술 관리 방법

5.1 변경 관리

개발 단계가 끝날 때마다 지속적으로 변경사항 유무를 확인한다.

변경 내용 발생 시 자유롭게 의견을 제시하고, 회의를 통해 결정한다.

웹 개발자는 개발 진행 도중 발생한 변경 관리 대상 항목들을 체크리스트 형식으로 기록하고 관리한다.

5.2 위험 관리

모든 회의 및 프로젝트 진행 간 팀원 간의 충분한 의사소통으로 오해가 없도록 한다.

매 단계별로 진행될 때 마다 위험 요소에 대해 검토한다.

5.3 비용 및 진도 관리

매주 수요일에 1 시간 이상의 정기 회의 시간과 토요일에 간단한 온라인 회의 시간에 업무의 진행도를 확인하여 일정 지연을 최대한 방지한다.

일정 모니터링을 통해 진행 상황을 공유하고, 지속적으로 관리한다.

추가로 작업이 발생하거나 감소할 시 변경 관리를 통해 일정을 조정한다.

5.4 위험 분석 및 해결 방안

위험	심각성	해결 방안
시험 준비	L	10월 초 중간고사 기간은 일정에서 제외한다.
기술적 취약	H	기술적 난이도를 고려하여 일정을 계획한다. 개발 과정에서 지속적인 피드백 및 지식을 교환한다.
설계 및 모델링 오류	E	모델링과 설계 과정을 최대한 자세하게 계획하고 여러 차례에 걸쳐 검토한다.
잘못된 기능 개발	E	요구 사항에서 분석했던 기능을 변경할 때에는 변경 관리 절차를 통해 목적물과 너무 상이한 결과물을 방지한다.
기술 오류 발생	E	충분한 테스트 단계를 통해 문제점을 찾고 해결한다.
기타 예상치 못한 리스크로 인한 일정 지연	H	초기 일정 계획 시 충분히 위험을 고려한다. 예상치 못한 리스크를 대비하여 여유 시간을 비축한 일정을 수립한다.

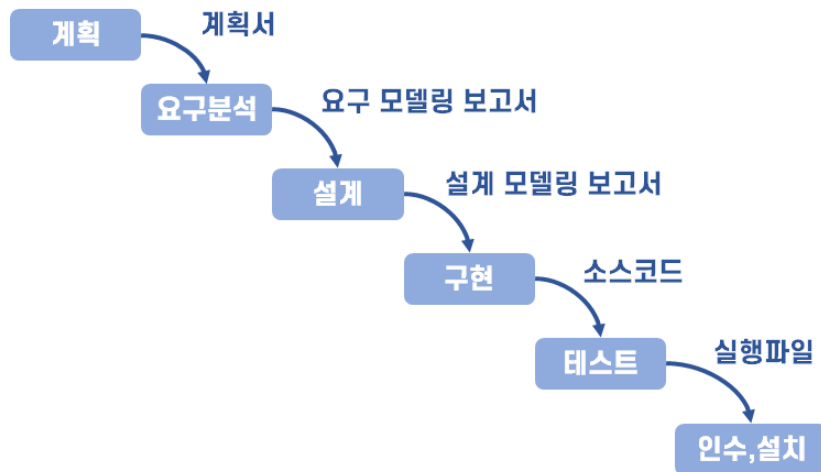
[표 5.1 위험 분석 및 해결 방안]

6. 개발 방법론 및 절차

개발 방법론으로는 객체지향 방법론을 사용한다.

절차는 계획 - 요구분석 - 설계 - 구현 - 테스트 - 인수,설치 과정으로 프로세스를 진행한다

요구분석 단계에서 요구를 분석함과 함께 모델링을 통해 객체와 객체 사이의 상호작용을 개념화하고, 설계 단계에서 상세화시키며 이후 객체지향 언어를 사용하여 객체 단위로 구현 및 테스트를 진행한다. 이때 단계를 진행함에 있어 폭포수 모델처럼 이전 단계를 완벽히 완수하고 그 산출물을 바탕으로 다음 프로세스에 착수하는 방법으로 절차를 진행한다.



[그림 6.1 프로세스 절차]

7. 검토 회의

7.1 검토 회의 일정

매주 수요일 오후 10 시에 1 시간 이상의 정기 회의를 행아웃을 활용한 온라인으로 가진다.

매주 토요일 오후 1 시에는 30 분 정도의 간이 회의를 가진다.

추가적인 검토 회의가 필요한 경우, 팀원간 일정을 조율하여 결정한다.

7.2 검토 회의 진행 방법

수요일 정기 회의에서는 프로젝트 매니저의 주도 하에 변경점과 필요 작업 사항, 산출물에 대한 검토 위주로 진행한다.

정기 회의의 내용은 프로젝트 매니저가 기록하여 관리한다.

간이 회의에서는 진행 사항 위주로 간단하게 변경점과 일정 관리에 대해 의견을 나눈다.

7.3 검토 회의 후속 조치

검토 회의의 내용을 기록하고, 변경되는 부분은 변경 관리 대상 항목 체크리스트에 반영한다.
기타 위험, 일정 등 중요 사항은 추가적으로 공지하여 모든 팀원이 숙지하게 한다.

8. 개발 환경

CPU	Intel i5-8265U
RAM	8GB
OS	WINDOWS 10 64bit
Programming Language	JAVA
IDE	Eclipse 4.16
Server	Apache tomcat 9.0.38
DB	Oracle Database 10g Express Edition

[표 8.1 개발 환경 표]

9. 성능 시험 방법

9.1 UI 테스트

UI 테스트 도구인 Selenium 을 활용하여 테스트를 진행한다.

페이지의 일관성, 데이터 입력과 출력, 오류 처리를 중점적으로 진행하고, 테스트 과정 발생한 변경 관리 대상 항목들은 체크리스트에 반영하고 수정한다.

9.2 기능 테스트

SW 구현 기능별 테스트 케이스를 작성 후, 오류 발생시엔 오류처리를 우선적으로 하며 테스트를 통과하면 다음 테스트 케이스 단계를 진행한다.

9.3 인수 테스트

알파 테스트를 하여 일부 사용자에게 직접 모든 기능을 테스트해보는 것을 보면서 모니터링하며 발견 오류와 문제점을 기록한다.

10. 문서화

10.1 요구분석 및 설계

요구 분석 명세서, 시나리오 설계서, UML 설계도, 데이터 베이스 설계서, 예외사항 설계 문서

10.2 기술

소스코드, API 문서, 테스트 문서, UI 설계도

10.3 사용자

일반 사용자 메뉴얼, 관리자용 메뉴얼

11. 유지 보수

11.1 정기 보수

정기적 유지 보수는 완전형 유지보수를 기본 사항으로 적용하여 사용자의 추가 요구변화 반영 및 코드의 기능 및 효율성 향상을 목적으로 월 2 회를 기준으로 한다.

11.2 긴급 보수

긴급 보수는 수정형 유지보수를 기본 사항으로 적용하여 소프트웨어 이용중 오류가 발견될 시 일주일 이내 시행하는 것을 원칙으로 한다.

11.3 버전 관리

소프트웨어 전체 버전관리는 분산형 버전 제어 시스템 Git 을 통해 기록하며 관리한다.

12. 설치, 인수

전체 소프트웨어 설치, 인수는 프로젝트 패키지를 사용될 서버 컴퓨터에 설치하는 과정을 거쳐 호스팅까지 직접 진행한다. 인수과정이 끝나면 즉시 웹 주소를 배포한다.

13. 참고 문헌 및 부록

최은만(2020). 소프트웨어 공학의 모든 것. 생능출판사

최은만(2017). 객체지향 소프트웨어 공학. 생능출판사

“Function point Languages Table”, QSM, <https://www.qsm.com/resources/function-point-languages-table>