

Selectable Parking Navigation System

Stage 1000. Plan and Elaboration

v.2.0

T1

200711443 안효빈

200711459 이남섭

200711453 류진렬

200811465 허준행

Activity 1001. Define Draft Plan

Name of system

- Selectable Parking Navigation System (이하 SPN System으로 표기)

Motivation

- 기존 백화점 혹은 마켓 주차장을 이용할 때 빈 주차 공간이 어디에 있는지 이용자가 알지 못해 헤매는 불편함이 있다.
- 대개 주차장을 이용하는 이용자는 자신의 방문 목적에 따라서 주차할 공간을 직접 선택하기를 선호한다.
- 주차장을 처음 이용하는 이용자는 자신이 모르는 주차장 내부 구조에 대한 시각적 안내를 받길 원한다.

Project Objective

- 이용자가 주차장을 이용함에 있어서 시간을 낭비하지 않고 편리하게 이용할 수 있는 시스템을 개발한다.
- 주차장을 관리하는 관리자의 입장에서 주차 상황을 간편하게 열람하고 관리할 수 있는 시스템을 개발한다.

Functional Requirement

- 관리자용 모드 로그인 기능을 제공한다.
- 프로그램 실행 권한 선택이 가능한 기능을 제공한다.
- 관리자용 암호를 수정하는 기능을 제공한다.
- 주차서비스 이외의 유용한 정보를 이용자에게 알려준다.
- 이용자 차량이 접근했을 때, 센서로 번호판의 정보를 시스템에 저장한다.
- 주차장 내부 상황을 이용자에게 시각적으로 출력하는 기능을 제공한다.
- 이용자에게 총별로 주차공간을 interface에 출력하는 기능을 제공한다.
- 관리자에게 총별로 주차공간을 interface에 출력하는 기능을 제공한다.
- 이용자의 주차장 요금을 계산한다.
- 관리자에게 요금 정책을 설정할 수 있도록 하는 interface를 제공한다.
- 이용자에게 주차 공간을 선택하게 하는 기능을 제공한다.
- 이용자에게 선택된 주차 공간으로 가는 경로를 알리는 기능을 제공한다.
- 이용자는 경로 알림 기능의 사용 여부를 선택 할 수 있다.
- 이용자에게 주차증을 발급해주는 기능을 제공한다.
- 시스템은 주차 공간의 상태를 갱신한다.
- 센서가 인식한 차량 번호를 시스템에 저장한다.
- 이용자가 수동으로 주차 공간의 상태를 갱신한다.
- 시스템은 이용 요금을 자동으로 계산한다.

- 이용자가 주차장에서 나갈 때 요금을 부과한다.
- 이용자에게 영수증을 제공한다.
- 이용자는 영수증 발급 여부를 결정 할 수 있다.
- 관리자는 주차 정보를 정렬하여 볼 수 있다.
- 관리자는 주차 정보를 검색해 볼 수 있다.
- 관리자는 주차 정보를 임의로 변경 가능 하다.

Non-Functional Requirement

- 관리자, 사용자, 입구, 출구 단말을 동기화한다.
- 시스템은 Real-Time에 기반한다.
- 사용자가 조작하기 편리한 interface을 제공한다.
- 관리자 모드에는 관리자만 접속이 가능해야한다.

Resource Estimation

- Human Effort (Man-Month): 3~4 M/M
- Human Resource: 컴퓨터 공학 전공 학부생 4명
- Project Duration: 11주
- Cost: 30만원

Activity 1002. Create Preliminary Investigation Report

Alternative Solution

1. 개발 전문 업체에 의뢰하여 제작한다.
2. 기능별 컴포넌트들을 구매하고 조합하여 개발한다.

Project Justification

1. 개발 전문 업체에 의뢰하여 제작한다.
 - Cost: 비용이 더 많이 들어갈 수 있다.
 - Risk: 처음 의도했던 것과 다른 결과물이 나올 수 있다.
 - Effect: 제작 기간 내에 요구사항에 부합하는 결과물을 얻을 수 있다.
2. 기능별 컴포넌트들을 구매하고 조합하여 개발한다.
 - Cost: 비용이 더 많이 들어갈 수 있다.
 - Duration: 컴포넌트 파악과 그것을 조합하는 데에 시간이 더 들어갈 수 있다.
 - Risk: 원하는 컴포넌트가 없을 수 있다.
 - Effect: 개발 후의 유지 보수가 편리하다.

Risk Management

Risk	Probability	Significancy	Weight
타 과목 과제 및 시험	5	4	20
축제, MT와 같은 학교 행사	5	1	10
팀원들 사이의 의견 충돌	2	4	10
Java 소켓 프로그래밍에 관한 이해 부족	5	5	25
GUI 구현에 미숙	3	4	15
OSP 기법에 대한 이해 부족	2	5	10
UML에 대한 이해 부족	3	4	15

Risk Reduction Plan

- Java 소켓 프로그래밍에 관한 이해 부족(25): Java를 활용한 네트워크 기술에 대한 연구.
- 타 과목 과제들(20): 팀원들이 모두 같은 수업을 들으므로, 유동적으로 대처가 가능하다.
또한 일주일에 3회 (목: 3~7시, 금: 2~5시, 토: 12~6시)로 정해 이외의 시간에 과제를 해결하기로 정함.
- UML에 대한 이해 부족(17): 도서관의 해당 자료들과 선택한 StarUML의 매뉴얼을 분석.
- GUI 구현에 미숙(15): Java에서 제공하는 여러 가지 GUI구현을 위한 API중 Swing을 선택하여 연구.

Market Analysis

- 대개 주차시스템의 경우 각 층별로 안내원을 1명 이상 배치하거나, 그렇지 못할 경우에는 이용자가 빈 주차 공간을 찾아 헤매는 경우가 자주 발생한다.
- 주차를 하고 용무를 보고 난 뒤 이용자가 세워뒀던 자신의 자동차의 위치를 기억하지 못해 불편을 겪는다.

Activity 1003. Define Requirements

1. 관리자 권한 로그인

- 비밀번호를 이용하여 프로그램을 관리자 권한으로 사용하는 기능.

2. 관리자 권한 암호 수정

- 관리자 권한을 사용하기 위한 암호를 수정한다.

3. 프로그램 실행 권한 선택

- 프로그램 시작 시 사용자(입구, 출구) 혹은 관리자 권한을 선택할 수 있는 기능.

4. 기타 정보 알림

- 이용자에게 유용한 정보 혹은, 긴급한 정보 등을 문자열로 출력하는 기능

5. 기타 정보 수정

- 관리자는 기타 정보의 내용을 수정 할 수 있다.

6. 주차장 혼잡도 알림

- 혼잡, 보통, 여유의 3가지 상태를 interface에 표시하는 기능.

7. 층별 주차공간을 interface에 출력

- 층별 주차공간을 이용자가 구분하기 편하게 시각적으로 출력하는 기능

8. 층별 주차공간을 interface에 출력 (관리자)

- 층별 주차공간의 상세한 정보를 관리자에게 시각적으로 출력하는 기능

9. 차량 번호판 저장

- 센서가 인식한 사용자의 번호판을 시스템에 저장

10. 요금 정책 설정

- 관리자가 요금계산을 위한 요금정책을 지정 할 수 있는 기능.

11. 주차공간을 선택

- interface를 이용하여 주차 공간을 선택할 수 있는 기능.

12. 선택된 주차공간으로 가는 경로를 알림

- 이용자가 선택한 주차공간으로 가는 경로를 interface에 출력하는 기능, 이용자는 interface를 통해 이 기능을 사용하지 않을 수 있음

13. 주차증 발급

- 주차 공간이 선택되고 차량 번호가 인식되었을 때 주차 주차증을 발급하는 기능.
- 선택된 주차 공간까지의 간략한 안내(도면 혹은 알림)를 출력하는 기능.

14. 층 별 주차공간의 상태를 자동 갱신

- 주차장 상태의 변경사항을 주기적으로 시스템이 자동으로 갱신하는 기능

15. 층 별 주차공간의 상태를 수동 갱신

- 사용자가 수동으로 주차공간 상태 갱신을 요청하는 기능.

16. 이용 요금 계산

- 시스템이 자동적으로 이용 요금을 계산한다.

17. 이용 요금 결제

- 이용자가 주차장에서 나갈 때 계산된 요금을 부과한다.

18. 영수증 출력

- 이용자에게 영수증을 출력하여 제공한다.

19. 영수증 생략

- 이용자는 영수증 출력여부를 선택 할 수 있다.

20. 주차 정보 정렬

- 관리자는 주차 정보를 정렬하여 볼 수 있다.

21. 주차 정보 검색

- 관리자는 여러 가지 정보를 통하여 주차 정보를 검색 할 수 있다.

22. 주차 정보 수정

- 관리자는 임의로 주차상태 및 이용자 주차정보를 수정 할 수 있다.

Ref.#	Function	Category
R.1.1	관리자 권한 로그인	Evident
R.1.2	관리자 권한 비밀번호 수정	Evident
R.1.3	프로그램 실행 권한 선택	Evident
R.2.1	기타 정보 알림	Evident
R.2.2	기타 정보 알림 수정	Evident
R.3.1	주차장 혼잡도 알림	Evident
R.3.2	층별 주차공간을 interface에 출력	Evident
R.3.3	층별 주차공간을 interface에 출력 (관리자)	Evident
R.3.4	층별 주차공간을 갱신	Hidden
R.3.5	층별 주차공간을 수동 갱신	Evident
R.4.1	주차장 요금 계산	Hidden
R.4.2	요금 정책 설정	Evident
R.4.3	이용 요금 결제	Evident
R.5.1	차량 번호판 저장	Hidden
R.6.1	주차 공간을 선택	Evident
R.6.2	선택된 주차공간으로 가는 경로 알림	Evident
R.6.3	경로 알림 생략	Evident
R.7.1	주차증 발급	Evident
R.7.2	영수증 출력	Evident
R.7.3	영수증 생략	Evident
R.8.1	주차정보 검색 기능	Evident
R.8.2	주차정보 정렬 기능	Evident
R.8.3	주차정보 임의 변경기능	Evident

Performance Requirements

- 사용자는 주차공간에 대한 정확한 정보를 실시간으로 제공받아 직접 선택할 수 있다.
- 관리자는 주차장의 다양한 정보를 열람, 수정할 수 있다.

Operating Environments

- OS: Windows XP 상위 버전의 Windows OS/Mac OS
- CPU: CoreDuo 이상의 프로세서
- Memory: 512MB
- HDD: 1GB 이상의 여유공간

Develop Environments

- OS: Window 7
- CPU: i3 이상
- Memory: 2GB 이상
- HDD: 200GB 이상

- Language: JAVA
- Tool: Eclipse, StarUML, MS Word, Hangle

Interface Requirements

- 사용자는 시각적으로 주차공간에 대한 정보를 제공 받는다.
- 관리자는 더욱 자세한 정보를 제공받는다.

Other Requirements

- 입구 프로그램, 출구 프로그램, 서버 프로그램은 서로 동일한 정보를 공유하며 동기화 한다.

Activity 1004. Record terms in Glossary

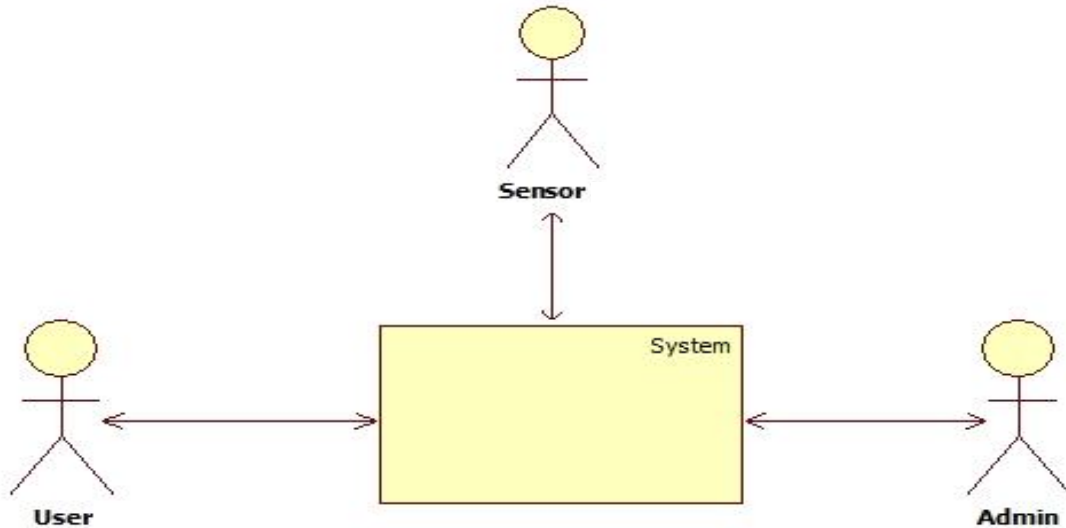
Term	Description	Remark
권한	프로그램 시작시 선택 가능한 항목으로 1)입구, 2)출구 3)관리자의 3개의 권한을 가지고 있다.	
센서	주차장 이용자의 차량이 입구 혹은 출구에 도착 했을 EO, 차량의 번호판을 식별하는 센서.	
차량번호	센서장비를 통해 인식되는 정보, 이용자-차량을 관리하는 고유 ID로 쓰인다. 센서장비가 차량번호를 인식할 때 사용자 프로그램은 활성화 된다.	
주차공간	이용가능, 이용불가, 이용 중의 3가지 상태를 가질 수 있으며, 차량번호, 요금등 기타 사용정보가 주차공간을 한 단위로 하여 관리 된다.	
요금정책	정액제와 후불제 두 가지가 있으며, 요금정책에 따라서 요금계산방식이 바뀐다.	
Interface	사용자에게 시각적으로 주차장 정보를 제공하며, 사용자의 입력을 직접 받는 시스템모듈, 관리자 권한에서는 사용자권한에서 보다 많은 정보를 표시.	
기타정보	주차서비스 이외의 이용자가 유용하게 사용할 수 있는 정보 혹은 긴급한 상황에 대한 정보	

Activity 1005. Implement Prototype

- 생략

Activity 1006. Define Business Use-Case

Step 1. Define System Boundary



Step 2. Identify and Describe Actors

- User: 주차장을 이용하는 모든 이용자.
주차를 하기 위해 주차장에 입장 혹은 퇴장하는 이용자이며, GUI interface를 접해본 이용자.
- Admin: 주차장을 관리하는 관리자.
SPN System의 주차 이용 정보를 검색, 열람, 수정하고 그 정보를 관리하는 주체.
- Sensor: 이용자 차량의 번호판을 인식하여 주차시스템에 전달하는 하드웨어

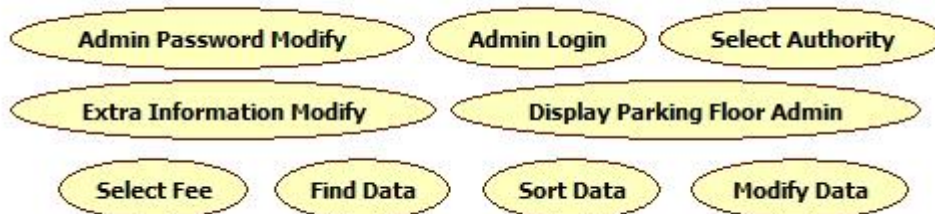
Step 3. Identify Use-cases

- Use-cases by actor-based

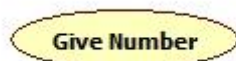
<User>



<Admin>



<Sensor>



- Use-cases by event-based



Step 4. Allocate system functions into Related Use-Cases.

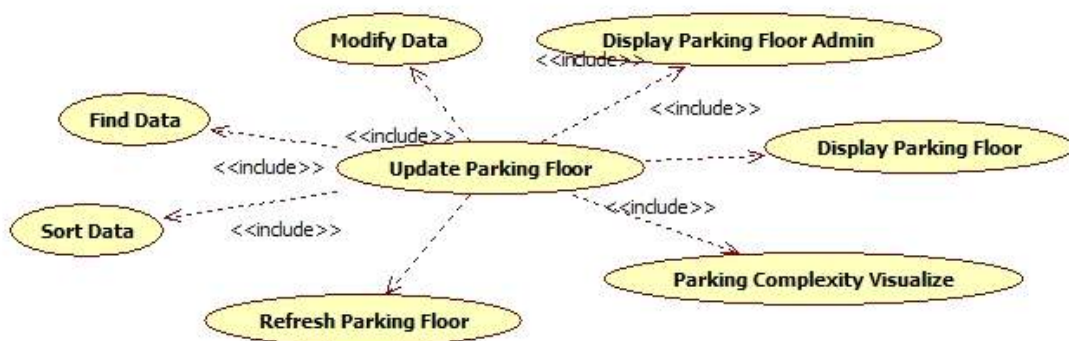
Ref.#	Function	Use Case Number & Name
R.1.1	관리자 권한 로그인	1. Admin Login
R.1.2	관리자 권한 비밀번호 수정	2. Admin Password Modify
R.1.3	프로그램 실행 권한 선택	3. Select Authority
R.2.1	기타 정보 알림	4. Extra Information
R.2.2	기타 정보 알림 수정	5. Extra Information Modify
R.3.1	주차장 혼잡도 알림	7. Parking Complexity Visualize
R.3.2	층별 주차공간을 interface에 출력	8. Display Parking Floor
R.3.3	층별 주차공간을 interface에 출력(관리자)	9. Display Parking Floor Admin
R.3.4	층별 주차공간을 갱신	10. Update Parking Floor
R.3.5	층별 주차공간을 수동 갱신	11. Refresh Parking Floor
R.4.1	주차장 요금 계산	12. Fee Calculation
R.4.2	요금 정책 설정	14. Select Fee
R.4.3	이용 요금 결제	15. Pay Fee
R.5.1	차량 번호판 저장	16. Give Number
R.6.1	주차 공간을 선택	17. Select Parking Lot
R.6.2	선택된 주차공간으로 가는 경로 알림	19. View Navigate
R.7.1	주차증 발급	21. Print Parking Permit
R.7.2	영수증 출력	23. Print Receipt
R.7.3	영수증 생략	24. Skip Receipt
R.8.1	주차정보 검색 기능	26. Find Data
R.8.2	주차정보 정렬 기능	27. Sort Data
R.8.3	주차정보 임의 변경기능	28. Modify Data

Step 5. Categorize Use-case

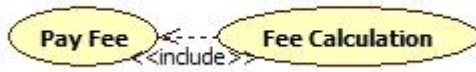
Ref.#	Use Case Number & Name	Category
R.1.1	1. Admin Login	Primary
R.1.2	2. Admin Password Modify	Primary
R.1.3	3. Select Authority	Primary
R.2.1	4. Extra Information	Primary
R.2.2	5. Extra Information Modify	Primary
R.3.1	7. Parking Complexity Visualize	Primary
R.3.2	8. Display Parking Floor	Primary
R.3.3	9. Display Parking Floor Admin	Primary
R.3.4	10. Update Parking Floor	Primary
R.3.5	11. Refresh Parking Floor	Secondary
R.4.1	12. Fee Calculation	Primary
R.4.2	14. Select Fee	Primary
R.4.3	15. Pay Fee	Primary
R.5.1	16. Give Number	Primary
R.6.1	17. Select Parking Lot	Primary
R.6.2	19. View Navigate	Primary
R.7.1	21. Print Parking Permit	Primary
R.7.2	23. Print Receipt	Primary
R.7.3	24. Skip Receipt	Secondary
R.8.1	26. Find Data	Primary
R.8.2	27. Sort Data	Primary
R.8.3	28. Modify Data	Primary

Step 6. Identify the relationships between Use-Cases

<주차정보 관리>



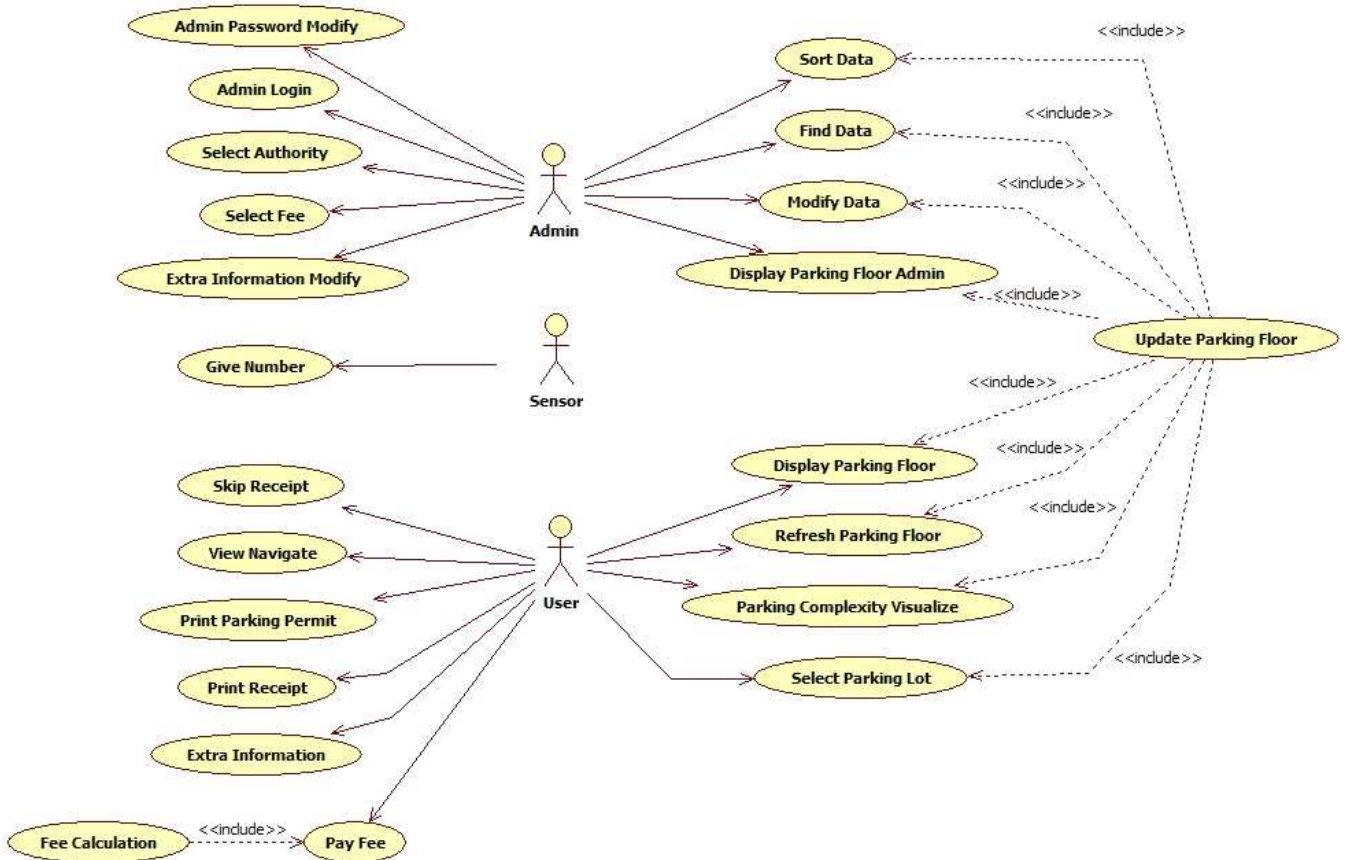
<요금 지불>



<관리자 로그인>



Step 7. Draw a Use-Case Diagram



Step 8. Describe Use-Case

Use Case Name	1. Admin Login
Actors	Admin
Description	관리자 비밀번호를 입력하여 관리자 권한으로 프로그램을 실행시킨다.
	비밀번호는 최대 3회까지 입력 할 수 있다.
	3회 이상 잘못 입력 할 경우 프로그램이 종료된다.
Use Case Name	2. Admin Password Modify
Actors	Admin
Description	기존 관리자 비밀번호와 기타 보안 정보를 입력하여 새로운 관리자 비밀번호를 설정한다.
	새로운 비밀번호 설정 후 다시 한 번 바뀐 비밀번호를 확인한다.
Use Case Name	3. Authority
Actors	Admin
Description	프로그램 실행 시 어떤 수준의 권한으로 프로그램이 실행될 지를 묻는다.
	이용자가 주차장에 입장할 때 서비스를 제공하는 입구권한, 나올 때 서비스를 제공하는 출구권한, 관리자가 주차장을 관리하기 위한 관리자 권한이 있다.
Use Case Name	4. Extra Information
Actors	User
Description	이용자에게 유용한 정보 혹은 긴급한 정보 등을 문자열로 표시한다.
	긴급한 정보는 더욱 눈에 잘 띄게 표시한다.
Use Case Name	5. Extra Information
Actors	Admin
Description	이용자에게 알릴 정보의 내용을 관리자가 수정 할 수 있다.
Use Case Name	6. Parking Complexity Visualize
Actors	User
Description	센서가 번호판을 식별할 때까지 주차장의 상태를 interface에 출력한다.
	'혼잡','보통','여유'의 3가지 문자로 주차공간에 대해서 알린다. - 70%이상 사용불가이면 '혼잡' - 70%미만 30%이상이면 '보통' - 30%미만이면 '여유'
Use Case Name	7. Display Parking Floor
Actors	User
Description	각각의 층마다 주차공간의 상황을 interface에 출력한다.
	주차공간은 '사용불가','사용가능','사용 중'의 상태를 가지며, 색상으로 구분한다.

Use Case Name	8. Display Parking Floor Admin
Actors	Admin
Description	Display Parking Floor보다 더 많은 정보를 출력한다.
	각 주차공간에 지정된 차량정보를 확인 할 수 있다.
	종합적인 주차정보를 한눈에 볼 수 있게 시각화 한다.

Use Case Name	9. Update Parking Floor
Actors	None
Description	이용자가 사용함에 불편함이 없게 시스템이 자동적으로 주차 공간의 상태를 갱신한다.

Use Case Name	10. Refresh Parking Floor
Actors	User
Description	사용자가 interface에 표시된 refresh 버튼을 이용하여 주차 공간의 상태를 수동으로 갱신 한다.

Use Case Name	11. Fee Calculation
Actors	None
Description	사용자가 좌석을 선택한 시점에서부터 주차장 이용요금을 계산한다.
	기본적으로 분단위로 요금을 계산한다.

Use Case Name	12. Select Fee
Actors	Admin
Description	관리자는 적용될 요금정책을 선택 할 수 있다.
	'정량제'는 이용시간에 관계없이 동일한 요금을 부과한다.
	'시간제'는 이용시간에 비례하여 요금을 부과한다.

Use Case Name	13. Pay Fee
Actors	User
Description	사용자가 주차장에서 나올 때 요금을 결제한다.
	사용시간 및 요금정책을 interface에 표시한다.
	최종 합계 금액을 interface에 표시한다.

Use Case Name	14. Activate Sensor
Actors	None
Description	사용자 차량의 번호를 센서가 식별할 때, 프로그램이 활성화 된다.
	비활성화 상태에서는 Parking Lot Visualize만 작동하며, 프로그램이 활성화될 때, Display Parking Floor도 함께 작동된다.

Use Case Name	15. Select Parking Lot
Actors	User
Description	사용자는 interface에 표시된 주차공간을 선택하여 주차장을 이용할 수 있다.
	'사용 가능'상태인 공간을 선택할 경우 안내 메시지를 표시하며 다음 단계로 넘어간다.
	'사용 불가','사용 중'상태를 선택할 때, 경고를 띄우며 올바른 선택을 기다린다.'

Use Case Name	16. View Navigate
Actors	User
Description	사용자가 주차공간을 선택한 후, 선택된 주차 공간까지의 경로를 interface에 출력한다.

Use Case Name	18. Print Parking Permit
Actors	User
Description	주차장이용 정보 및 주차공간 경로가 입력된 주차증을 발급한다.

Use Case Name	19.Print Receipt
Actors	User
Description	요금 결제 후, 사용자에게 영수증을 출력한다.

Use Case Name	21. Find Data
Actors	Admin
Description	관리자가 선택한 조건으로 원하는 데이터를 찾을 수 있다.
	차량번호, 주차공간 번호, 건물 층 등으로 검색 할 수 있다.

Use Case Name	22. Sort Data
Actors	Admin
Description	관리자가 선택한 조건으로 원하는 데이터를 정렬 할 수 있다.
	차량번호, 주차공간 번호, 건물 층 등으로 정렬 할 수 있다.

Use Case Name	23. Modify Data
Actors	Admin
Description	시스템의 데이터를 관리자가 수정 할 수 있다.
	주차공간과 주차공간 사이의 정보를 교체를 할 수 있다.
	이용시간, 차량정보 등을 수정 할 수 있다.
	수동으로 주차공간에 차량을 등록 할 수 있다.

Step 9. Rank Use-Case

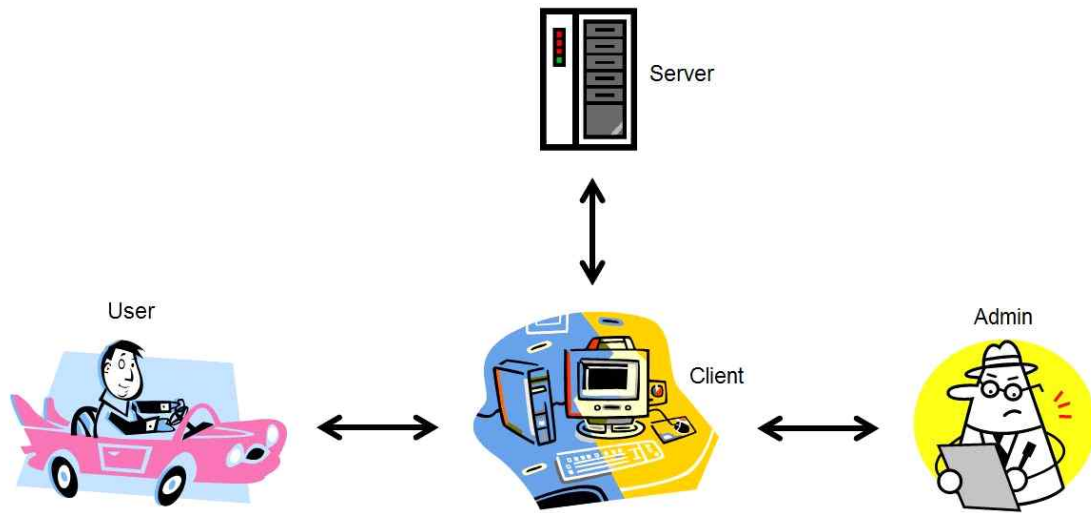
Ref.#	Use Case Number & Name	Rank
R.1.1	1. Admin Login	High
R.1.2	2. Admin Password Modify	High
R.1.3	3. Select Authority	High
R.2.1	4. Extra Information	Low
R.2.2	5. Extra Information Modify	Low
R.3.1	7. Parking Complexity Visualize	High
R.3.2	8. Display Parking Floor	High
R.3.3	9. Display Parking Floor Admin	High
R.3.4	10. Update Parking Floor	High
R.3.5	11. Refresh Parking Floor	Medium
R.4.1	12. Fee Calculation	High
R.4.2	14. Select Fee	High
R.4.3	15. Pay Fee	High
R.5.1	16. Give Number	High
R.6.1	17. Select Parking Lot	High
R.6.2	19. View Navigate	High
R.7.1	21. Print Parking Permit	High
R.7.2	23. Print Receipt	High
R.7.3	24. Skip Receipt	Medium
R.8.1	26. Find Data	High
R.8.2	27. Sort Data	High
R.8.3	28. Modify Data	High

Activity 1007. Define Business Concept Model

Identify 'Concepts' in the target domain

입구	출구	관리자	차량번호	정량제
계산영수증	층	주차 공간	경로	공지
이용시간	요금	결제	이용기록	주차장 상태
주차장데이터	자료 정렬	자료 검색	자료 수정	프로그램 권한
인터페이스	요금 계산기	시간제	비상	번호판 센서
주차증				

Activity 1008. Define Draft System Architecture



Activity 1009. Refine Plan

Project Scope

- 하드웨어(영수증, 주차증 출력기, 요금 결제기, 자동차 번호판 인식기)는 직접 만들지 않는다.
- 위의 하드웨어를 기반으로 하였을 때, 결제 정보, 주차장 이용 정보 및 관리 정보를 나타내주는 프로그램을 개발한다.

Project Objectives

- 시스템을 처음 이용하는 이용자로도 어려움 없이 사용할 수 있는 편리한 interface를 제공한다.
- 주차장 이용자에게 빈 주차 공간을 안내하고 주차장 내부 상태를 표시하여 이용자가 헤매는 시간을 일 없이 주차장을 이용할 수 있게 한다.

Functional Requirements

Ref.#	Function	Category
R.1.1	관리자 권한 로그인	Evident
R.1.2	관리자 권한 비밀번호 수정	Evident
R.1.3	프로그램 실행 권한 선택	Evident
R.2.1	기타 정보 알림	Evident
R.2.2	기타 정보 알림 수정	Evident
R.3.1	주차장 혼잡도 알림	Evident
R.3.2	층별 주차공간을 interface에 출력	Evident
R.3.3	층별 주차공간을 interface에 출력 (관리자)	Evident
R.3.4	층별 주차공간을 갱신	Hidden
R.3.5	층별 주차공간을 수동 갱신	Evident
R.4.1	주차장 요금 계산	Hidden
R.4.2	요금 정책 설정	Evident
R.4.3	이용 요금 결제	Evident
R.5.1	차량 번호판 저장	Hidden
R.6.1	주차 공간을 선택	Evident
R.6.2	선택된 주차공간으로 가는 경로 알림	Evident
R.7.1	주차증 발급	Evident
R.7.2	영수증 출력	Evident
R.7.3	영수증 생략	Evident
R.8.1	주차정보 검색 기능	Evident
R.8.2	주차정보 정렬 기능	Evident
R.8.3	주차정보 임의 변경기능	Evident

Performance Requirements

- 이용자가 주차 공간을 선택하고 시스템이 주차 안내를 하기까지의 시간은 최대 3초 이내이다.
- 화면이 전환될 때 최대 2초 이내에 완료되어야 한다.
- 퇴장하는 차량의 번호를 센서가 인식한 후 요금을 계산하기까지 최대 2초 이내여야 한다.

Operating Environment

- OS: Windows XP 상위 버전의 Windows OS/Mac OS
- CPU: CoreDuo 이상의 프로세서
- Memory: 512MB
- HDD: 1GB 이상의 여유공간

User Interface Requirements

- 시스템이 제공하는 주차 공간 안내는 이용자가 이해하기 쉽고 간단해야한다.
- 사용하기 쉽고 간단한 interface.

Other Requirements

- 시스템과 연결된 다른 기기들과 통신하는 부분을 위해서 시스템의 모듈화를 확실히 해야 한다.
- 다음 버전을 위해서 확장성을 마련해야한다.
- 데이터베이스에 내용들이 확실하게 보존되어야한다.
- 시스템 접근 권한이 확실히 구분되어야 한다.

Resources

- Human Effort (Man-Month) : 3~4 M/M
- Human Resource : 컴퓨터 공학 전공 학부생 4명
- Project Duration : 11주
- Cost : 30만원
- Software : JAVA, JDK, Eclipse, StarUML
- Hardware : i3 이상 CPU, 2GB 이상 Memory, 200GB 이상 HDD

Configuration Management

- 각 단계는 UML Project Repository를 참고한다.

Quality Assurance Plan

- 모든 기능이 요구사항을 완벽하게 수행한다.

Scheduling

Phase(00x0)/Activity(000x)	Schedule(week)											
	1	2	3	4	5	6	7	8	9	10	11	12
1001. Define Draft Plan	—											
1002. Create Preliminary Investigation Report	—											
1003. Define Requirements	—											
1004. Record Terms in Glossary	—											
1005. Implement Prototype		—										
1006. Define Use Cases		—										
1007. Define Draft Conceptual Model		—										
1008. Define Draft System Architecture		—										
1009. Refine Plan		—										
2010. Revise Plan			—									
2020. Synchronize Artifacts			—									
2030. Analyze			—									
2031. Define Essential Use Case			—									
2032. Refine Use Case Diagrams			—									
2033. Refine conceptual Model			—									
2034. Refine Glossary			—									
2035. Define System Sequence Diagrams			—									
2036. Define Operation Contracts			—									
2037. Define State Diagrams			—									
2040. Design			—									
2041. Define Real Use Cases			—									
2042. Define Reports, UI and Storyboards			—									
2043. Refine System Architecture			—									
2044. Define Interaction Class Diagrams			—									
2045. Define Design Class Diagrams			—									
2046. Define Database Schema			—									
2050. Construct			—									
2051. Implement Class & Interface Definition			—									
2052. Implement Methods			—									
2053. Implement Windows			—									
2054. Implement Reports			—									
2055. Implement DB Schema			—									
2056. Write Test Code			—									
2060. Test			—									
2061. Unit Testing			—									
2062. Integration Testing			—									
2063. System Testing			—									
2064. Performance Testing			—									
2065. Acceptance Testing			—									
2066. Documentation Testing			—									
2110. Revise Plan				—								
2120. Synchronize Artifacts				—								
2130. Analyze				—								
2131. Define Essential Use Case				—								
2132. Refine Use Case Diagrams				—								
2133. Refine conceptual Model				—								
2134. Refine Glossary				—								
2135. Define System Sequence Diagrams				—								
2136. Define Operation Contracts				—								
2137. Define State Diagrams				—								
2140. Design				—								
2141. Define Real Use Cases				—								
2142. Define Reports, UI and Storyboards				—								
2143. Refine System Architecture				—								
2144. Define Interaction Class Diagrams				—								
2145. Define Design Class Diagrams				—								
2146. Define Database Schema				—								
2150. Construct				—								
2151. Implement Class & Interface Definition				—								
2152. Implement Methods				—								
2153. Implement Windows				—								
2154. Implement Reports				—								
2155. Implement DB Schema				—								
2156. Write Test Code				—								
2160. Test				—								
2161. Unit Testing				—								
2162. Integration Testing				—								
2163. System Testing				—								
2164. Performance Testing				—								
2165. Acceptance Testing				—								
2166. Documentation Testing				—								
2210. Revise Plan					—							
2220. Synchronize Artifacts					—							
2230. Analyze					—							
2231. Define Essential Use Case					—							
2232. Refine Use Case Diagrams					—							
2233. Refine conceptual Model					—							
2234. Refine Glossary					—							
2235. Define System Sequence Diagrams					—							
2236. Define Operation Contracts					—							
2237. Define State Diagrams					—							
2240. Design					—							
2241. Define Real Use Cases					—							
2242. Define Reports, UI and Storyboards					—							
2243. Refine System Architecture					—							
2244. Define Interaction Class Diagrams					—							
2245. Define Design Class Diagrams					—							
2246. Define Database Schema					—							
2250. Construct					—							
2251. Implement Class & Interface Definition					—							
2252. Implement Methods					—							
2253. Implement Windows					—							
2254. Implement Reports					—							
2255. Implement DB Schema					—							
2256. Write Test Code					—							
2260. Test					—							
2261. Unit Testing					—							
2262. Integration Testing					—							
2263. System Testing					—							
2264. Performance Testing					—							
2265. Acceptance Testing					—							
2266. Documentation Testing					—							