

– OOAS Stage 1000 –

아씨오 자료관리 시스템 – Plan and Elaboration –

– 소프트웨어 모델링 및 분석 –

T4
임국현
김우빈
이자형
차금옥

Project: 아씨오(소환마법 주문) 자료관리 시스템

Activity1001. Define Draft Plan

·Motivation

- 보통 집에 있는 물건을 오랫동안 안 쓰다 보면 사용하고 싶을 때 어디다 뒀는지 제때 못 찾아 약속시간에 늦게 되고, 물건들을 종류대로 정리를 하지 않고 마구마구 놓아 집이 어질러 지는 것을 막기 위해 만들었다.
- 이 시스템을 통하여 집안을 어지럽히지 않고 빠르고 편리하게 원하는 물건을 찾을 수 있어 시간을 절약할 수 있을 것으로 예상된다.

·Project Objectives

- 특정 물건의 위치를 쉽고 빠르게 얻을 수 있다.
- 어떤 장소에 어떤 물건이 있는지 편리하게 파악 할 수 있다.
- 물건의 상세한 정보 조회로 비슷한 물건을 헛갈리지 않고 정확히 찾을 수 있다.
- /* - 물건 관리 시스템을 통하여 물건들이 분류되고 집안의 공간도 효율적으로 사용할 수 있다. */

·Project Scope

- 본 프로젝트는 기반시설(RFID혹은 그와 관련된 하드웨어)과 기반설비가 다 갖추어져 있다는 전제하에 RFID와 관련데이터를 저장하고 컨트롤 하는 소프트웨어 그리고 하드웨어와의 연관 관계 까지가 범위이다.

- Functional Requirements

- Login
- Logout
- Make account
- Delete account
- Find account
- Find password
- Item map
- Map expand
- Map reduction
- Item denoting
- Item highlighting
- Item list
- Item click
- Search
- Category Search
- Detail info
- belonging
- Add account to DB
- Delete account to DB
- Search account to DB

- Non-Functional Requirements

- 시스템의 유지보수가 쉬워야 한다.
- 사용자가 쓰기 편리한 UI 환경을 제공해야 한다.
- 집안의 모든 물건의 위치를 파악 가능토록 한다.
- 디스플레이 화면을 사용자가 쉽게 파악하도록 한다.

- Resource Estimation

- Human Efforts(Man-Month): 4-4
- Human Resource: 4명
- Duration: 16주
- Budget: 1000만원

- Other Information

- Future Version

Activity1002. Create Preliminary Investigation Report

·Alternative Solutions

- RFID로 구현하지 않는 경우엔 현재 물건을 개별적으로 판별할 수단이 존재하지 않음.

·Project Justification (Business Demands)

- Effect: 실시간으로 물건을 정확하게 찾을 수 있음.
- Cost: 소프트웨어 구현 비용 1000만원.
- Duration: 반영구적임.
- Risk:

RFID로 구축했을 때 찾지 못하게 될 가능성.
 프로젝트 수행 경험부족
 OSP를 처음으로 접함
 프로그래밍 지식 부족
 도구 사용 능력 결여
 RFID로 구축했을 때 찾지 못하게 될 가능성.
 빈번한 요구사항 변경
 유사 프로그램 등장
 기반 시스템 부족
 하드웨어의 기술적 결함
 하드웨어의 기술수준의 부족

·Risk Management

Risk	Probability	Significance	Weight
프로젝트 수행 경험부족	20	7	140
OSP를 처음으로 접함	18	9	162
프로그래밍 지식 부족	16	10	160
도구 사용 능력 결여	12	6	72
RFID로 물건을 찾지 못하게 될 가능성.	2	1	2
빈번한 요구사항 변경	14	8	112
유사 프로그램 등장	10	5	50
기반 시스템 부족	4	2	8
하드웨어의 기술적 결함	6	3	18
하드웨어의 기술수준의 부족	8	4	32

·Risk Reduction Plan

- 프로젝트 수행 경험부족
경험이 많은 사람에게 조언을 구한다.
- OSP를 처음으로 접함
OSP관련 자료를 찾고 Demo를 구해서 분석하여 참고자료로 사용한다.
- 프로그래밍 지식 부족
프로젝트를 진행 하면서 개인이 개별적으로 공부한다.

- 도구 사용 능력 부족
참고서적을 구매하거나 교육을 받아 사용능력을 기른다.
- RFID로 구축했을 때 찾지 못하게 될 가능성.
시스템적 결함인지 환경의 영향인지 면밀히 분석하여 해결책은 찾아낸다
시스템적 결함인 경우 프로젝트 관리 시스템으로 해결하고 환경적 영향인 경우 그것을 극복해 낼 수 있는 해결책을 찾아낸다.
- 빈번한 요구사항 변경
프로젝트를 시작하기 전에 최대한 요구사항을 fix 시키고 허용되는 범위를 정하여 그 안에서만 요구사항을 변경할수 있도록 프로젝트 계획을 세운다.
- 기반시스템 부족
SVN 혹은 Bug 관리시스템 등 프로젝트를 수행하는데 있어 필요한 여러 기반시스템을 구축하고 시스템에 관한 교육을 하여 프로젝트 수행 능력을 기른다.
- 유사 프로그램 등장
이 프로그램을 특허 등록하고 관련 기능을 암호화, black box화 하여 시스템 카피를 막고 법적인 보호를 받는다.
- 하드웨어 기술적 결함
하드웨어 구입 업체를 재선정한다.
- 하드웨어의 기술수준의 부족
하드웨어 구입 업체를 재선정 한다.

·Market Analysis

기존에 존재하지 않던 시스템 이므로 경쟁 상대가 없어 시장에 내놓았을 때 독점적으로 제품 공급이 가능하다.

·Other Managerial Issues

프로젝트는 2012/5/31까지 완료가 되어야 한다.

Activity1003. Define Requirements

- Functional Requirements
 - Login
(ID, Password를 입력하여 로그인 한다.)
 - Logout
(사용하지 않을 때 시스템에서 로그아웃을 한다.)
 - Make account
(ID, Password, 이름, ID중복, 비밀번호 재확인, 항목 미 기재 확인)
 - Delete account
(사용자의 계정을 삭제함)
 - Find account
(사용자 계정을 찾음.)
 - Find password
(비밀번호를 잃어버렸을 때를 위한 비밀번호 찾기 기능)
 - Item map
(집 전체 지도, 물건의 위치 표현, 위치 강조 기능,공간 확대 기능)
 - Map expand
(자세한 위치를 보기 위하여 확대 한다.)
 - Map reduction
(확대한 것을 원래대로 돌린다.)
 - Item denoting
(검색한 물건의 위치를 맵에 표시되게 한다.)
 - Item highlighting
(검색 한 항목들 중 선택한 항목이 어디에 있는지 map에 표시)
 - Item list
(물건 검색시 검색 결과 출력)
 - Item click
(아이템을 클릭한다.)
 - Search
(찾고 싶은 물건을 검색한다.)
 - Category Search
(물건을 분류 유형에 따라 검색한다.)
 - Detail info
(검색 항목 중 하나를 두 번 클릭, 새 창에 상세 정보가 나온다.)
 - belonging
(특정 사용자만 물건을 검색 할 수 있게 귀속 시킴)
 - Add account to DB
(DB에 계정을 추가함)
 - Delete account to DB
(DB에서 계정을 삭제함)
 - Search account to DB
(DB에서 계정을 찾음.)

· Functional Requirements (Categorized Table)

Ref.#	Function	Category
R1.1	- Login	Evident
R1.2	- Logout	Evident
R1.3	- Make account	Evident
R1.4	- Delete account	Evident
R1.5	- Find account	Evident

R1.6	- Find password	Evident
R2.1.1	- Item map	Evident
R2.1.2	- Map expand	Evident
R2.1.3	- Map reduction	Evident
R2.2.1	- Item list	Evident
R2.2.2	- Item denoting	Evident
R2.2.3	- Item click	Evident
R2.2.3	- Item highlighting	Evident
R2.2.4.1	- Detail info	Evident
R2.2.4.2	- belonging	Evident
R3.1	- Search	Evident
R3.2	- Category Search	Evident
R4.1	- Add account to DB	Hidden
R4.2	- Delete account to DB	Hidden
R4.3	- Search account to DB	Hidden

· Performance Requirements

- 집에 있는 물건 검색 시 검색 결과를 빠른 속도로 알려줘야 한다.
- 화면에 이미지 처리 속도가 느려서는 안된다.

· Operating Environment

- OS: Microsoft Windows XP
- CPU: Pentium IV
- Memory: 1GB
- H.D.D: 4GB

· 개발환경

- OS: Microsoft Windows XP
- CPU: Intel i7 processor
- DB: Mysql
- Program Language: Java
- Case: Visio
- Editor: Eclipse

· Interface Requirements

- 로그인한 사용자가 기능을 선택하면 각 기능이 수행됨
- 각 사용자가 개별적으로 소유한 물건은 특정 사용자계정으로 로그인해야 검색 결과에 반영이 된다.

· Other Requirements

- RFID System은 항상 Running 상태에 있어야 한다.
- DB와 관련된 데이터는 항상 무결성을 유지해야 한다.
- 집 이외의 장소에서 물건의 RFID tag를 스캔 하는 것을 막아야 하고 tag안에 있는 데이터를 암호화하여 보호해야 함.

Activity1004. Record Terms in Glossary

Term	Description	Remarks
Map	물건들이 존재하는 공간을 2차원 평면으로 보여주는 역할을 함 검색을 했을 때 물건의 위치를 알 수 있게 하는 기능을 가짐.	
List	검색된 물건의 목록을 표시해 주는 역할을 함	
RFID tag	물건에 붙여 물건의 정보, 기능 등 그 물건과 관련된 데이터를 저장하는 역할을 하는 tag	
RFID	통신의 일종으로 단거리 무선 주파수로 특정 개체를 판별할 수 있는 정보를 알 수 있게 하는 차세대 인식 기술.	

Activity1005. Implement Prototype



로그인

계정(U)

암호(P)

비밀번호가 기억안나시나요?
계정을 버려요!

확인(O) 계정추가 취소(C)



계정만들기

계정이름

암호

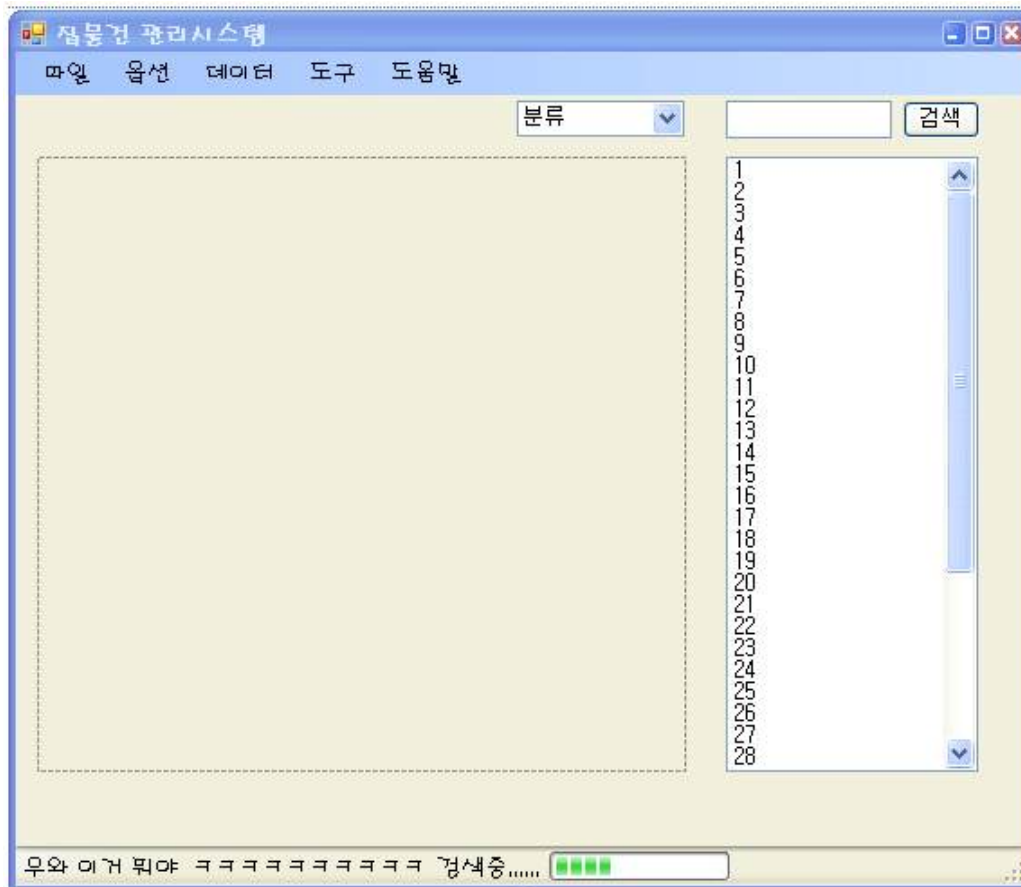
이메일

암호찾기에 사용됩니다.

계정이미지

※노란색부분은 필수입니다

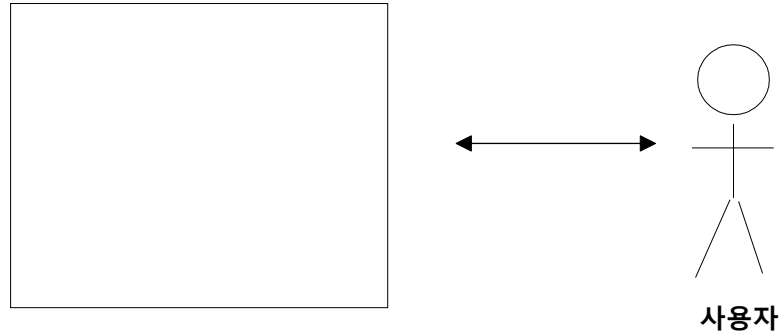
계정 생성 취소



Activity1006. Define Business Use Case

- Step 1. Define system boundary.

아씨오 자료관리 시스템



- Step 2. Identify and describe actors.

사용자 - 하드웨어가 구축된 공간 안에 물건을 가지고 들어오고 나감.

시스템을 사용하여 찾고자 하는 물건이 어떤 것이 있는지 어디에 있는지 찾음.

검색한 물건의 상세정보를 조회하여 정확도를 높임.

집에 어떤 물건들이 있는지 알 수 있음.

시스템에 로그인 할 계정을 만들 수 있음.

- Step 3. Identify use cases

Use-cases by actor-based



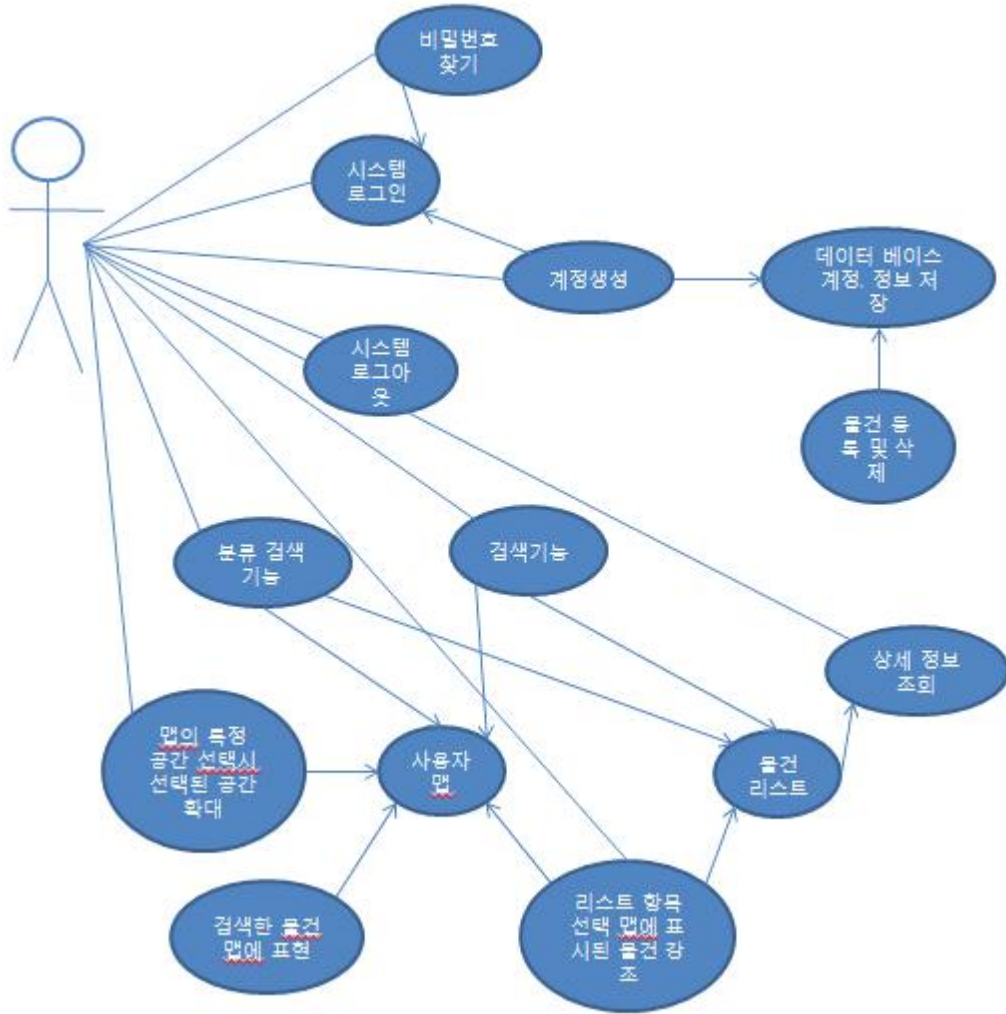
· Step 4. Allocate system functions into related use cases

Ref. #	Function	Use Case Number & Name	category
R1.1	- Login	1. Login	Evident
R1.2	- Logout	2. Logout	Evident
R1.3	- Make account	3. Make account	Evident
R1.4	- Delete account	4. Delete account	Evident
R1.5	- Find account	5. Find account	Evident
R1.6	- Find password	6. Find password	Evident
R2.1.1	- Item map		Evident
R2.1.2	- Map expand	7. Map expand	Evident
R2.1.3	- Map reduction	8. Map reduction	Evident
R2.2.1	- Item list		Evident
R2.2.2	- Item denoting		Evident
R2.2.3	- Item click	9. Item click	Evident
R2.2.3	- Item highlighting		Evident
R2.2.4.1	- Detail info	10. Detail info	Evident
R2.2.4.2	- Belonging	11. Belonging	Evident
R3.1	- Search	12. Search	Evident
R3.2	- Category Search	13. Category Search	Evident
R4.1	- Add account to DB		Hidden
R4.2	- Delete account to DB		Hidden
R4.3	- Search account to DB		Hidden

· Step 5. Categorize use cases

Use-Case	Number & Name	Category
1.	Login	Primary
2.	Logout	Primary
3.	Make account	Primary
4.	Delete account	Secondary
5.	Find account	Secondary
6.	Find password	Secondary
7.	Map expand	Primary
8.	Map reduction	Primary
9.	Item click	Primary
10.	Detail info	Primary
11.	Belonging	Primary
12.	Search	Primary
13.	Category Search	Secondary

· Step 7. Draw a use case diagram



· Step 8. Describe use cases

Name	1. Login
Actors	사용자
Description	- 특정 사용자가 물건 검색 시스템에 접근하기 위해 접속을 해야 한다. - 아이디, 비밀번호 입력 -> 확인 버튼 클릭 -> 아이디와 비밀번호체크 -> 유효하지 않다면 다시 입력 메시지 출력 -> 유효한 경우 시스템 메인 화면 띄움 - 언제든지 취소를 누르면 종료
Name	2. Logout
Actors	사용자
Description	- 시스템에 다른 사람이 접근하는 것을 막기 위해 로그아웃 함. - 로그아웃 버튼 클릭 -> 시스템 로그아웃 -> 시스템 로그인 화면으로 전환
Name	3. Make account
Actors	사용자
Description	- 시스템에 접속 하기 위해 계정을 생성한다. - 계정생성버튼 클릭 -> 계정 생성 항목 입력 -> 필수 입력요소 확인 -> 항목이 유효한 경우 아이디 생성 -> 무효한 다시 입력 메시지 출력 -> - 취소를 누르면 언제든지 로그인 창으로
Name	4. Delete account
Actors	사용자
Description	- 생성했던 계정을 삭제한다. - 계정삭제버튼 클릭 -> 계정 ID 항목 입력 -> password 입력 -> 항목이 유효한 경우 아이디 삭제 -> 무효한 다시 입력 메시지 출력 -> - 취소를 누르면 언제든지 로그인 창으로
Name	5. Find account
Actors	사용자
Description	- 계정을 잊어버린 경우 계정을 찾는다. - 계정 찾기 버튼 클릭 -> Email, password 입력 -> 존재하는 Email인지 확인 -> 존재하는 경우 메일로 아이디 보내줌 -> 없는 Email인 경우 다시 입력 요청 -> - 취소를 누르면 언제든지 로그인 창으로
Name	6. Find password
Actors	사용자
Description	- 비밀번호 찾기 버튼 클릭 -> 사용자아이디 입력 -> 사용자 메일 주소 확인 -> 메일로 비밀번호 전송 - 취소버튼 클릭시 언제든지 로그인 창으로
Name	7. Map expand
Actors	사용자
Description	- Map의 특정 공간을 선택하면 선택된 공간의 화면을 확대하여 더욱 자세하게 볼 수 있도록 함
Name	8. Map reduction
Actors	사용자
Description	- 확대된 Map을 클릭 하면 다시 원래대로 작아짐
Name	9. Detail info
Actors	사용자
Description	- 리스트의 항목을 더블 클릭 하면 상세 조회 창이 뜬.
Name	10. Item click
Actors	사용자

Description	- 리스트의 항목을 클릭하면 화면에 아이템 위치가 강조 됨
Name	11. Belonging
Actors	사용자
Description	- 로그인 한 사용자의 계정에 물건을 귀속 시킴
Name	12. Search
Actors	사용자
Description	- 텍스트 필드에 기입 -> 검색버튼 누름 -> 검색된 항목 리스트에 출력 -> 검색된 항목 맵에 출력

Name	13. Category Search
Actors	사용자
Description	분류할 항목을 선택 하여 검색할 항목을 걸러냄

· Step 9. Rank use cases

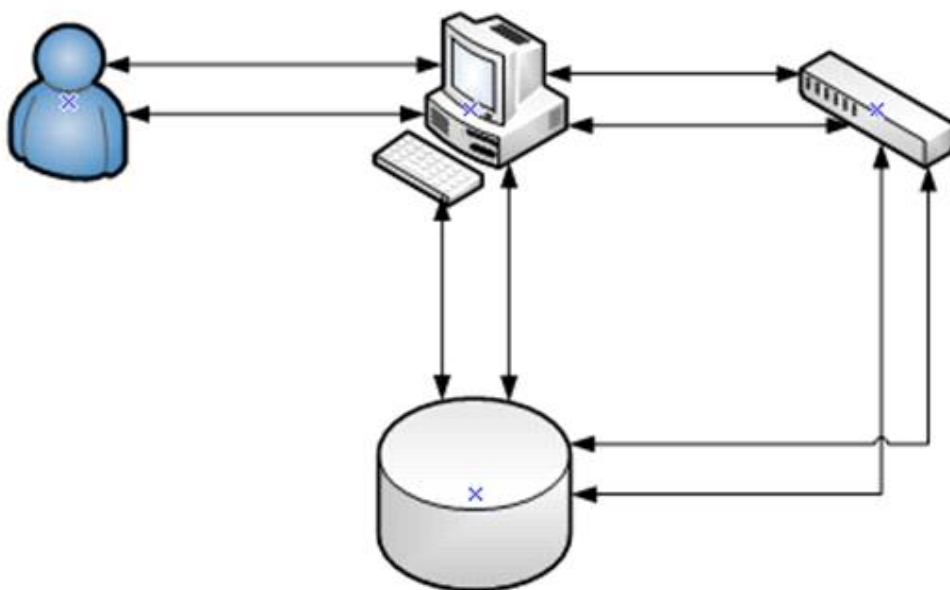
Rank	Use-Case Number & Name
	1. Login
	2. Logout
	3. Make account
	4. Delete account
	5. Find account
	6. Find password
	7. Map expand
	8. Map reduction
	9. Item click
	10. Detail info
	11. Belonging
	12. Search
	13. Category Search

Activity1007. Define Business Concept Model

identify 'Concepts' in the target domain.



Activity1008. Define Draft System Architecture



Activity1009. Refine Plan

· Project Scope

- 본 프로젝트는 기반시설(RFID혹은 그와 관련된 하드웨어)과 기반설비가 다 갖추어져 있다는 전제하에 RFID와 관련데이터를 저장하고 컨트롤 하는 소프트웨어 그리고 하드웨어와의 연관 관계 까지가 범위이다.

· Project Objectives

- 특정 물건의 위치를 쉽고 빠르게 얻을 수 있다.
- 어떤 장소에 어떤 물건이 있는지 편리하게 파악 할 수 있다.
- 물건의 상세한 정보 조회로 비슷한 물건을 헛갈리지 않고 정확히 찾을 수 있다.

/* - 물건 관리 시스템을 통하여 물건들이 분류되고 집안의 공간도 효율적으로 사용할 수 있다. */

• Functional Requirements

- Login
- Logout
- Make account
- Delete account
- Find account
- Find password
- Item map
- Map expand
- Map reduction
- Item denoting
- Item highlighting
- Item list
- Item click
- Search
- Category Search
- Detail info
- belonging
- Add account to DB
- Delete account to DB
- Search account to DB

- Program Language: Java
- Case: Visio
- Editor: Eclipse

· Scheduling

STAGE	PHASE/ACTIVITY	SCHEDULE(WEEK)					
		1	2	3	4	5	6
1000. Plan & Elaborate	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	█					
2000. Build	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 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					█		

· Configuration Management

각 단계는 UML-Project-Repository를 참고한다.

· Quality Assurance Plan

기능들은 요구사항에서 요구하는 기능들을 완벽하게 수행한다.