

Software Modeling & Analysis

낚시하시계

(Fishing Digital Watch)

Team 8

201714170 조해성

201714168 유호원

201310507 정희찬

201613575 루카이

201712845 류한길

Index

Activity 1001. Define Draft Plan

1. Motivation & Object
- 2 Functional Requirements
- 3 Non-Functional Requirements
4. Resource Estimation

Activity 1002. Create Preliminary Investigation Report

1. Alternative Solutions
2. Project Justification
3. Risk Management
4. Risk Reduction Plans
5. Market Analysis
6. Other Managerial Issues

Activity 1003. Define Requirements

1. Functional Requirements
2. Performance Requirements
3. Operating Requirements
4. Other Requirements

Activity 1006. Define Use Case

1. Define System Boundary
2. Identify Use Case
3. Allocate System Function

4. Categorize Use Cases
5. Identify Relationship Between Use Cases
6. Draw A Use Case Diagram
7. Describe Use Cases
8. Rank Use Cases

Activity 1008. Define Business Concept Models

Activity 1009. Define System Test Plan

Activity 1010. Refine Plan

1. Project Scope
2. Project Objectives
3. Functional Requirements
4. Non-Functional Requirements
5. Resources
6. Scheduling

Activity 1001. Define Draft Plan

1. Motivation & Objects

2017년 낚시가 등산을 누르고 취미생활 1위를 했다는 기사가 나오고, 낚시 카페와 "도시어부" 프로그램이 인기를 끄는 등 점점 낚시가 비전문가도 접하기 쉬워지고 있다. 그러나 그와 반대로 스포츠 중 (아내에게) 가장 사망률이 높다는 통계가 있을 만큼 무턱대고 하면 안되는 위험한 스포츠 중 하나이다. 낚시를 할 때에는 두 가지, 밀물과 썰물 시간을 아는 것과 조차의 차이 정도를 아는 것이 중요하다. 밀물과 썰물에 의해 수심이 달라지고, 이에 따라서 잡을 수 있는 어종이나 낚시 포인트가 달라지기 때문에 낚시를 즐기고 싶다면 밀물과 썰물 시간을 알아야 한다. 조차의 차이는 달의 인력에 따라 달라지는데, 달이 완전히 차거나 그믐일 때에 인력이 커지고 이에 따라 조차의 차이도 커진다. 즉 달의 모양에 따라 물때가 다르다는 뜻으로 물이 빠지고 차는 높이가 달라져 조황과 자리 선택, 공략 어종 선택에 큰 영향을 끼친다는 뜻이다. 따라서 우리는 평소 일반 시계로 쓰다 낚시를 할 때 밀물 썰물 때 및 달의 위상을 통해 조차의 차이 정도를 알려주는 낚시 시계로 쓸 수 있다면 안전하고 즐거운 낚시에 도움이 될 것으로 생각하여 개발하게 되었다.

2. Functional Requirements

- Time Keeping
- Timer
- Stopwatch
- Alarm
- Tide Level (밀물 썰물 표시계)
- Moon Phase (달의 위상)

3. Non-Functional Requirements

- 가시성이 좋아야 한다
- 평소에도 캐주얼하게 사용할 수 있도록 모드가 복잡하면 안된다
- 4 만원대의 너무 비싸지 않은 가격이어야 한다

4. Resource Estimation

- Human Effort : 5M/M
- Human Resource : 5M
- Project Duration : 3 Months
- Cost : 5000 만원

Activity 1002. Create Preliminary Investigation Report

1. Alternative Solution

- 아웃소싱을 한다
- 공모전을 열어 대학생들을 싼값에 이용한다 (상금 100 만원)
- 스마트폰의 낚시 어플리케이션을 사온다
- 인터넷에 정규직 전환여부에 참고한다 말하고 시킨다

2. Project Justification (Business Demands)

- Cost : 비용대비 효율이 뛰어나다
- Duration : 3 개월

- Risk : 개발 능력 부족, 팀원간 의사소통 부족, 낯시 이해도 부족, 팀원의 탈주, 팀원 간의 시간 조율, 맛있는 학식으로 인한 의지 저하
- Effects : 휴대폰이 안되더라도 시계만으로 정보를 확인할 수 있다, 기능이 추가되어 다양한 용도로 쓸 수 있다.

3. Risk Management

Risk	Probability	Significance	Weight
개발능력부족	4	2	8
의사소통부족	3	1	3
낯시 이해도 부족	5	1	5
팀원의 탈주	1	5	5
팀원간 시간 조율	2	4	8
맛있는 학식으로 의지약화	4	5	20

4. Risk Reduction Plan

- 학식 (20) : 밥을 세종대 가서 먹는다
- 개발능력부족 (8) : 관련서적을 읽고 개인적으로 공부한다
- 시간조율 (8) : 화상채팅으로라도 참가한다
- 팀원의 탈주 (5) : 마음의 편지를 써서 섭섭한 점을 공유한다
- 낯시 이해도 부족 (5) : 도시 어부를 본다
- 의사소통 부족 (3) : 매주 회식을 한다

5. Market Analysis

- 낚시에 관심이 많아지면서 저렴한 낚시 시계의 수요가 늘어날 것으로 여겨짐

6. Other Managerial Issues

-이번 학기내에 완성시켜야 한다

Activity 1003. Define Requirements

1. Functional Requirements

No	Function	Description
0	Time Keeping	현재 시간을 알려준다
1	Timer	시간을 정하고 측정한다
2	Alarm	정해진 시간에 부저가 울리게 한다
3	Stopwatch	시간의 흐름을 측정한다
4	Tide Level	현재 물의 높이를 알려준다
5	Moon Phase	현재 달의 상태를 알려준다

R 0.0	keepTime	hidden
R 0.1	adjustTime	evident
R 1.0	setTimer	evident
R 1.1	pauseTimer	evident
R 1.2	resetTimer	evident
R 2.0	setAlarm	evident
R 2.1	buzzAlarm	hidden

R 2.2	stopAlarm	evident
R 3.0	startStopwatch	evident
R 3.1	recordStopwatch	evident
R 3.2	pauseStopwatch	evident
R 3.3	resetStopwatch	evident
R 4.0	showTide	evident
R 4.1	calculateTide	hidden
R 4.2	nextTide	evident
R 5.0	ShowMoonphase	hidden
R 5.1	calculateMoonphase	hidden

2. Performance Requirements

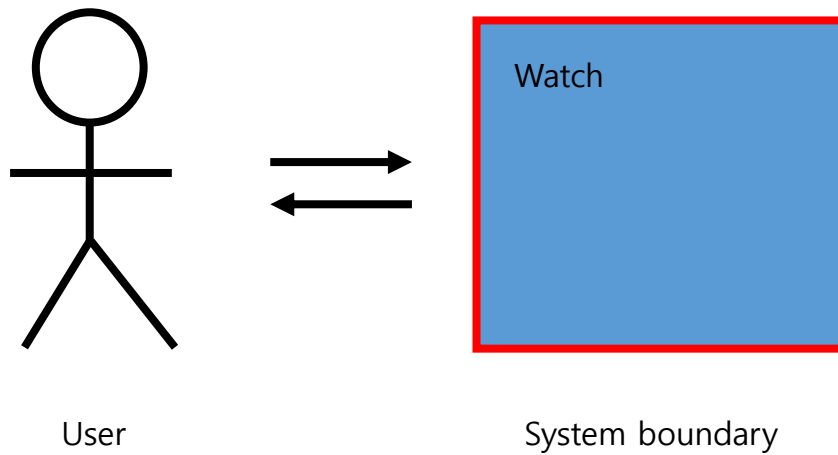
- 간단하고 직관적인 기능 인터페이스를 제공해야 한다
- 비전문가도 사용하기 쉬워야 한다
- 시간이나 밀물 썰물의 오차를 최소화해야 한다

3. Operating Requirements

- OS : Windows 7, 10
- Language : JAVA

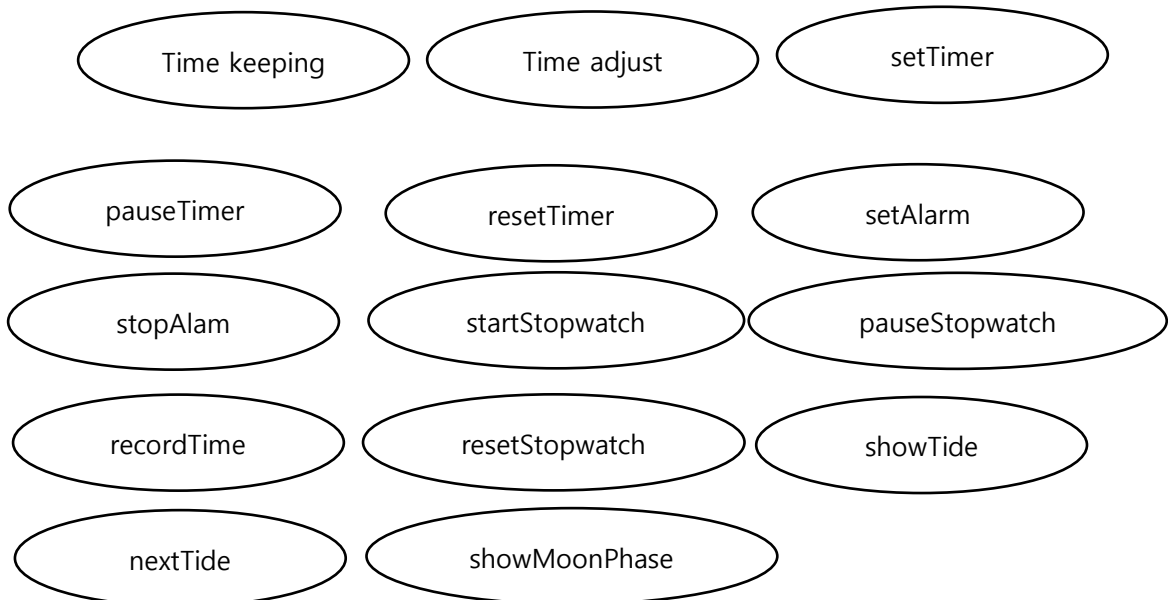
Activity 1006. Define Use Case

1. Define System Boundary

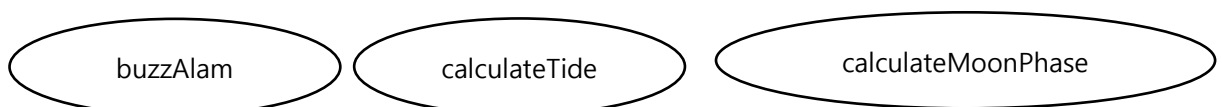


2. Identify Use Case

-use case by actor based



-use case by event



3. Allocate System Function

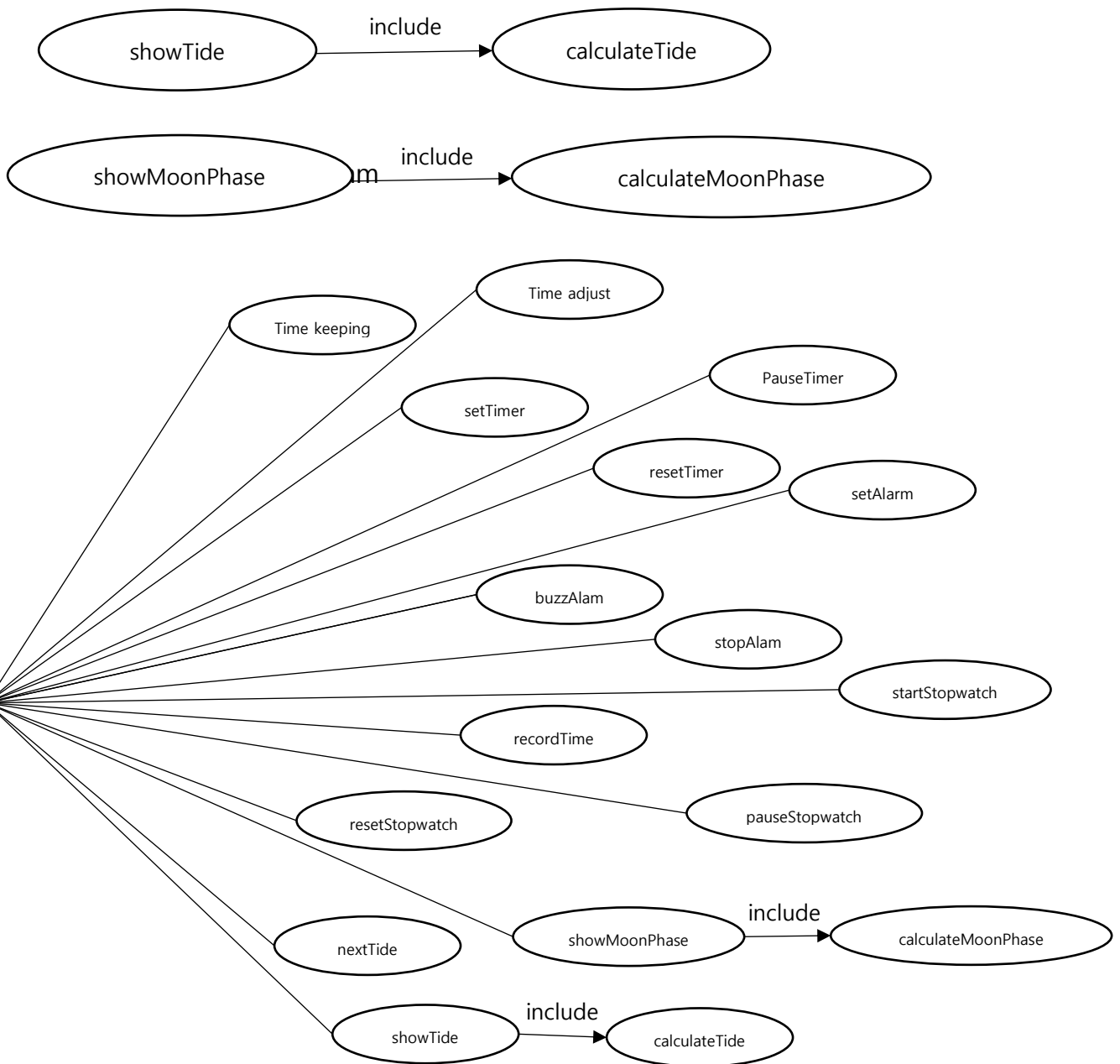
No	Function	Use-Case
0	Time Keeping	Time Keeping
1	Timer	Timer
2	Alarm	Alarm
3	Stopwatch	Stopwatch
4	Tide Level	Tide Level
5	Moon Phase	Moon Phase

4. Categorize Use Case

R 0.0	keepTime	Primary
R 0.1	adjustTime	Primary
R 1.0	setTimer	Primary
R 1.1	pauseTimer	Primary
R 1.2	resetTimer	Primary
R 2.0	setAlarm	Primary
R 2.1	buzzAlarm	Primary
R 2.2	stopAlarm	Primary
R 3.0	startStopwatch	Primary
R 3.1	recordStopwatch	Primary
R 3.2	pauseStopwatch	Primary
R 3.3	resetStopwatch	Primary

R 4.0	showTide	Primary
R 4.1	calculateTide	Primary
R 4.2	nextTide	Primary
R 5.0	ShowMoonphase	Primary
R 5.1	calculateMoonphase	Primary

5. Identify Relationship Between Use Cases



7. Describe Use Case

Use Case	1. keepTime
Actor	User
Description	현재 시간을 보여준다

Use Case	2. adjustTime
Actor	User
Description	현재 시간을 설정한다 버튼을 누르면 차례대로 시, 분이 돌아가며 조정된다

Use Case	3. setTimer
Actor	User
Description	사용자가 입력한 시간의 타이머를 설정한다

Use Case	4. pauseTimer
Actor	User
Description	타이머를 일시정지 시킨다

Use Case	5. resetTimer
Actor	User
Description	타이머를 일시정지 시킨다

Use Case	6. setAlarm
Actor	User
Description	사용자가 입력한 시간에 알람을 설정한다

Use Case	7. buzzAlarm
Actor	User
Description	사용자가 입력한 시간에 알람을 울린다

Use Case	8. stopAlarm
Actor	User
Description	알람을 정지시킨다

Use Case	9. startStopwatch
Actor	User
Description	스탑워치를 시작한다

Use Case	10. recordStopwatch
Actor	User
Description	기록을 측정하여 저장한다 여러 개의 기록도 저장할수 있다

Use Case	11. pauseStopwatch
Actor	User
Description	스탑워치를 일시정지 시킨다

Use Case	12. resetStopwatch
Actor	User
Description	스탑워치를 초기화 시킨다

Use Case	13. showTide
Actor	User
Description	조수간만의 차를 표시해준다 3 개의 바다(동해, 서해, 남해)를 나타낸다

Use Case	14. calculateTide
Actor	None
Description	조수간만의 차를 계산한다

Use Case	15. nextTide
Actor	User
Description	다음 바다의 조수간만의 차로 화면을 바꾼다

Use Case	16. showMoonphase
Actor	User
Description	달의 모양을 표시해준다

Use Case	17. calculateMoonphase
Actor	None
Description	날짜를 계산하여 달의 모양을 맞춰준다

9. Rank Use-Case

R 0.0	keepTime	High
R 0.1	adjustTime	High
R 1.0	setTimer	High
R 1.1	pauseTimer	High
R 1.2	resetTimer	High
R 2.0	setAlarm	High
R 2.1	buzzAlarm	High
R 2.2	stopAlarm	High
R 3.0	startStopwatch	High
R 3.1	recordStopwatch	High
R 3.2	pauseStopwatch	High
R 3.3	resetStopwatch	High
R 4.0	showTide	High

R 4.1	calculateTide	High
R 4.2	nextTide	High
R 5.0	ShowMoonphase	High
R 5.1	calculateMoonphase	High

Activity 1008. Define Business Concept Models

편리한 UI

간편한 휴대성

저렴한 가격

낚시에 필요한 정보

Activity 1009. Define System Test Plan

R 0.0	keepTime	시간이 정상적으로 출력되는지 확인한다 오차가 최대한 적은지 확인한다 가독성이 나쁘지 않은지 확인한다
R 0.1	adjustTime	시간이 정상적으로 변경이 되었는지 확인한다
R 1.0	setTimer	타이머가 잘 설정이 되었는지 확인한다
R 1.1	pauseTimer	버튼을 누르면 타이머가 잘 멈춰야 한다
R 1.2	resetTimer	타이머가 잘 초기화 되는지 확인한다
R 2.0	setAlarm	정상적으로 알람이 설정되었는지 확인한다 알람을 삭제하는 것도 가능한지 확인한다
R 2.1	buzzAlarm	정해진 시간에 정확히 알람이 울리는지 확인한다

R 2.2	stopAlarm	알람이 정상적으로 멈추는지 확인한다
R 3.0	startStopwatch	버튼을 누르면 시간이 잘 가기 시작하는지 확인한다.
R 3.1	recordStopwatch	버튼을 누른 때의 시간이 잘 기록되는지 확인한다.
R 3.2	pauseStopwatch	버튼을 누르면 시간이 잘 멈추는지 확인한다.
R 3.3	resetStopwatch	버튼을 누르면 시간이 잘 초기화되는지 확인한다.
R 4.0	showTide	물높이가 잘 출력되는지 확인한다.
R 4.1	calculateTide	물높이가 잘 계산되는지 확인한다. 오차가 적은지 확인한다
R 4.2	nextTide	다음 바다가 잘 나타나는지 확인한다.
R 5.0	ShowMoonphase	달의 위상을 정확히 표시하는지 확인한다
R 5.1	calculateMoonphase	달의 위상을 정확히 계산하는지 확인한다 오차가 적은지 확인한다

Activity 1010. Refine Plan

1. Project Scope & Objective

비전문가가 평상시에도 캐주얼하게 사용하는 단순하고 저렴한 낚시용 시계를 개발한다

2. Functional Requirements

- Time Keeping
 - Timer
 - Stopwatch
 - Alarm
 - Tide Level (밀물 썰물 표시계)
 - Moon Phase (달의 위상)

3. Non-Functional Requirements

- 가시성이 좋아야 한다
- 평소에도 캐주얼하게 사용할 수 있도록 모드가 복잡하면 안된다
- 4 만원대의 너무 비싸지 않은 가격이어야 한다
- 비전문가도 사용하기 쉬워야 한다

4. Operating Environments

- OS : Windows 7, 10
- Language : JAVA

5. Resources

- Human Resource : 5M
- Project Duration : 3 Months
- Cost : 5000 만원

6. Scheduling

stage	Activity	schedule (week)										
		1	2	3	4	5	6	7	8	9	10	11
1000 Plan & Elaboration	1001. Define Draft Plan	█	█									
	1002. Create Preliminary Investigation Report	█	█									
	1003. Define Requirements	█	█									
	1004. Record Terms in Glossary	█	█									
	1005. Implement Prototype	█	█									
	1006. Define Draft System Architecture	█	█									
	1007. Define Business Use Case	█	█									
	1008. Define Business Concept Model	█	█									
	1009. Define System Test Case	█	█									
	1010. Refine Plan	█	█									
	2010. Revise Plan			█	█	█						
	2020. Synchronize Artifacts			█	█	█						
	2031. Define Essential Use Cases			█	█	█						

2000 Build	2030 Analyze	2032. Refine Use Case Diagrams																	
		2033. Define Domain Model																	
		2034. Refine Glossary																	
		2035. Define System Sequence Diagrams																	
		2036. Define Operation Contracts																	
		2037. Define State Diagrams																	
		2038. Refine System Test Case																	
		2039. Analyze (2030) Traceability Analysis																	
	2040 Design	2041. Design Real Use Cases																	
		2042. Define Reports, UI, and Storyboards																	
		2043. Refine System Architecture																	
		2044. Define Interaction Diagrams																	
		2045. Define Design Class Diagrams																	
		2046. Design Traceability Analysis																	
		2047. Define Database Schema																	
2050 Construct	2051. Implement Class & Methods Definitions																		
	2052. Implement Windows																		
	2053. Implement Reports																		

		2054. Implement DB Schema(SQL, etc.)																
		2055. Write Unit Test Code																
	2060 Test	2061. Unit Testing																
		2062. Integration Testing																
		2063. System Testing																
		2064. Performance Testing																
		2065. Acceptance Testing																
		2066. Documentation Testing																
		2067. Testing Traceability Analysis																