

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)
- Mode (기능 선택)

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
- 인턴에게 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
6	Mode	원하는 기능을 선택해 해당 기능만 상시 사용하도록 한다.

R 0.0	showTime	hidden
R 0.1	adjustTime	evident
R 1.0	showTimer	hidden
R 1.1	setTimer	Evident

R 1.2	startTimer	Evident
R 1.3	pauseTimer	Evident
R 1.4	resetTimer	Evident
R 1.5	buzzTimer	Hidden
R 2.0	showAlarm	Hidden
R 2.1	nextAlarm	Evident
R 2.2	addAlarm	Evident
R 2.3	deleteAlarm	Evident
R 2.4	buzzAlarm	Evident
R 2.5	stopAlarm	Evident
R 3.0	showStopwatch	Evident
R 3.1	startStopwatch	Evident
R 3.2	recordStopwatch	Evident
R 3.3	pauseStopwatch	Evident
R 3.4	resetStopwatch	Evident
R 4.0	showTide	Hidden
R 4.1	nextTide	Evident
R 4.2	calculateTide	Hidden
R 5.0	showMoonphase	Hidden
R 5.1	calculateMoonphase	Hidden
R 6.0	modeSelect	Evident
R 6.1	nextMode	evident

2. Performance Requirements

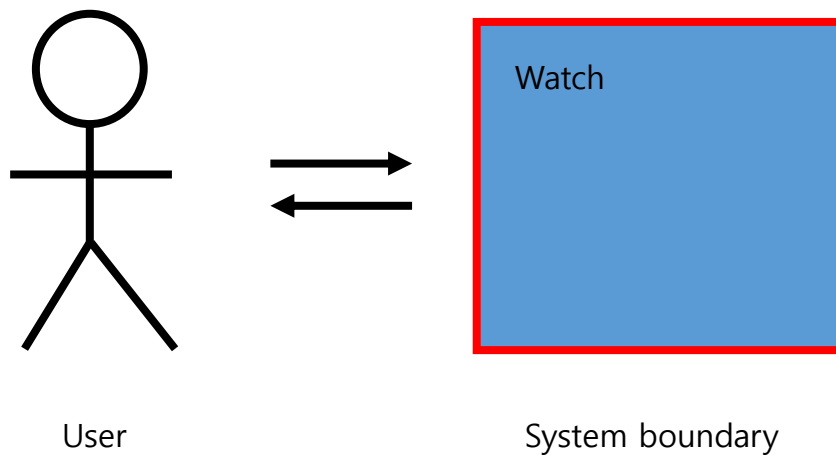
- 간단하고 직관적인 기능 인터페이스를 제공해야 한다
- 비전문가도 사용하기 쉬워야 한다
- 시간이나 밀물 썰물의 오차를 최소화해야 한다

3. Operating Requirements

- OS : Windows 7, 10
- Language : JAVA

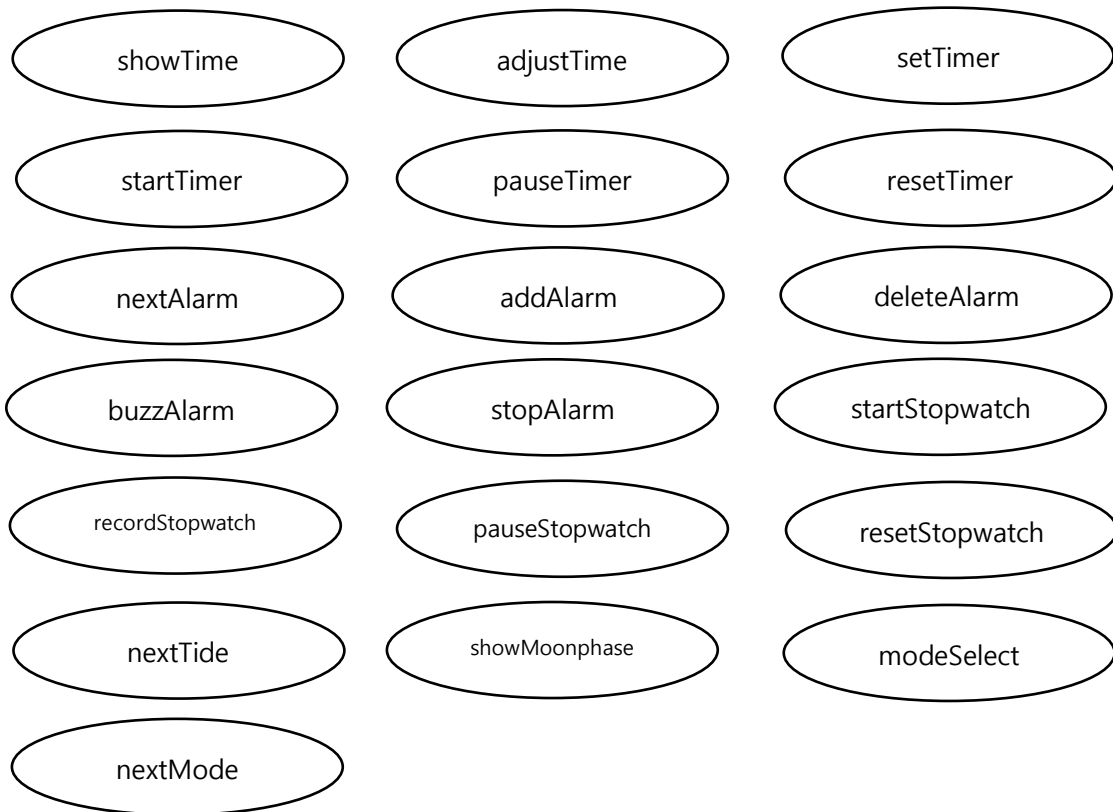
Activity 1006. Define Use Case

1. Define System Boundary

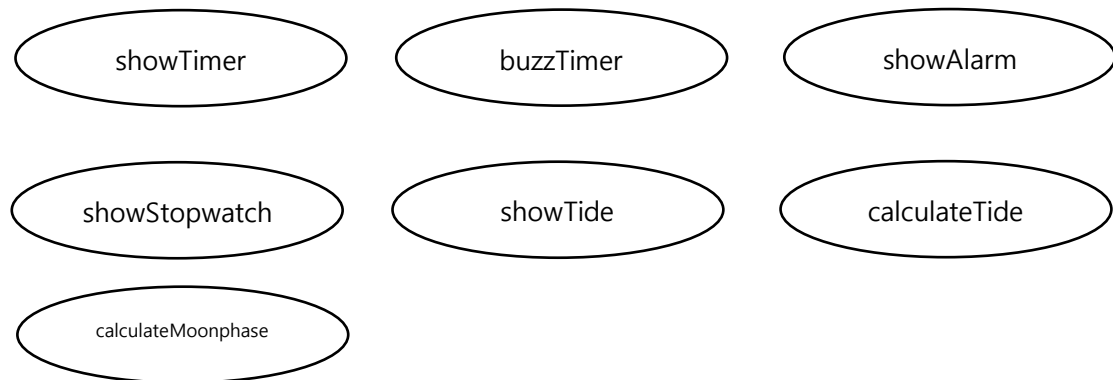


2. Identify Use Case

-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

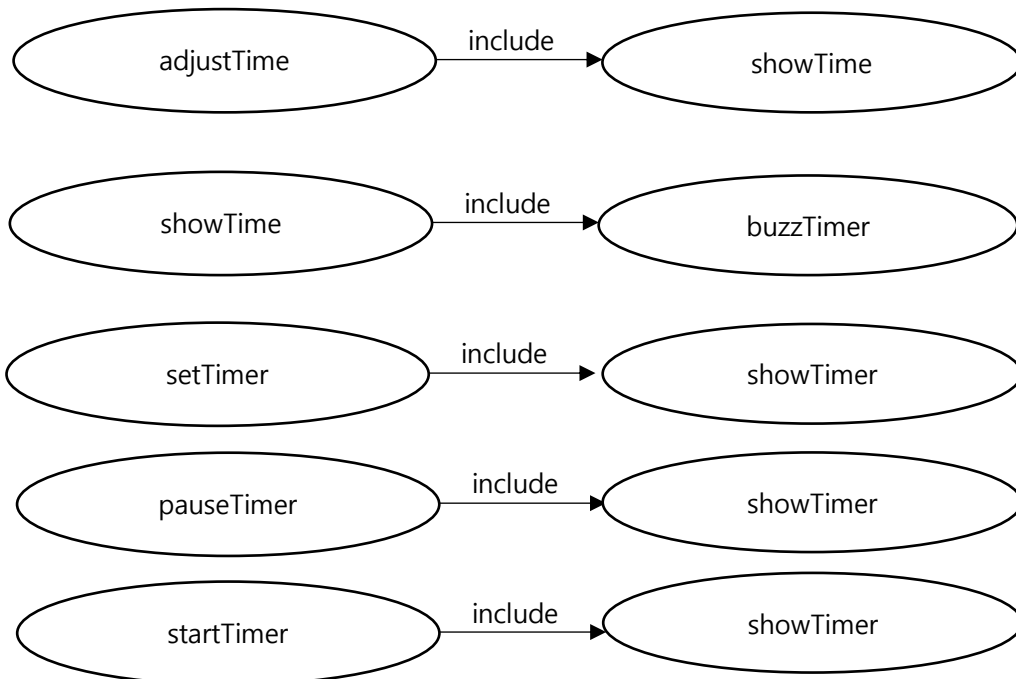
2	Alarm	Alarm
3	Stopwatch	Stopwatch
4	Tide Level	Tide Level
5	Moon Phase	Moon Phase
6	Mode	Mode

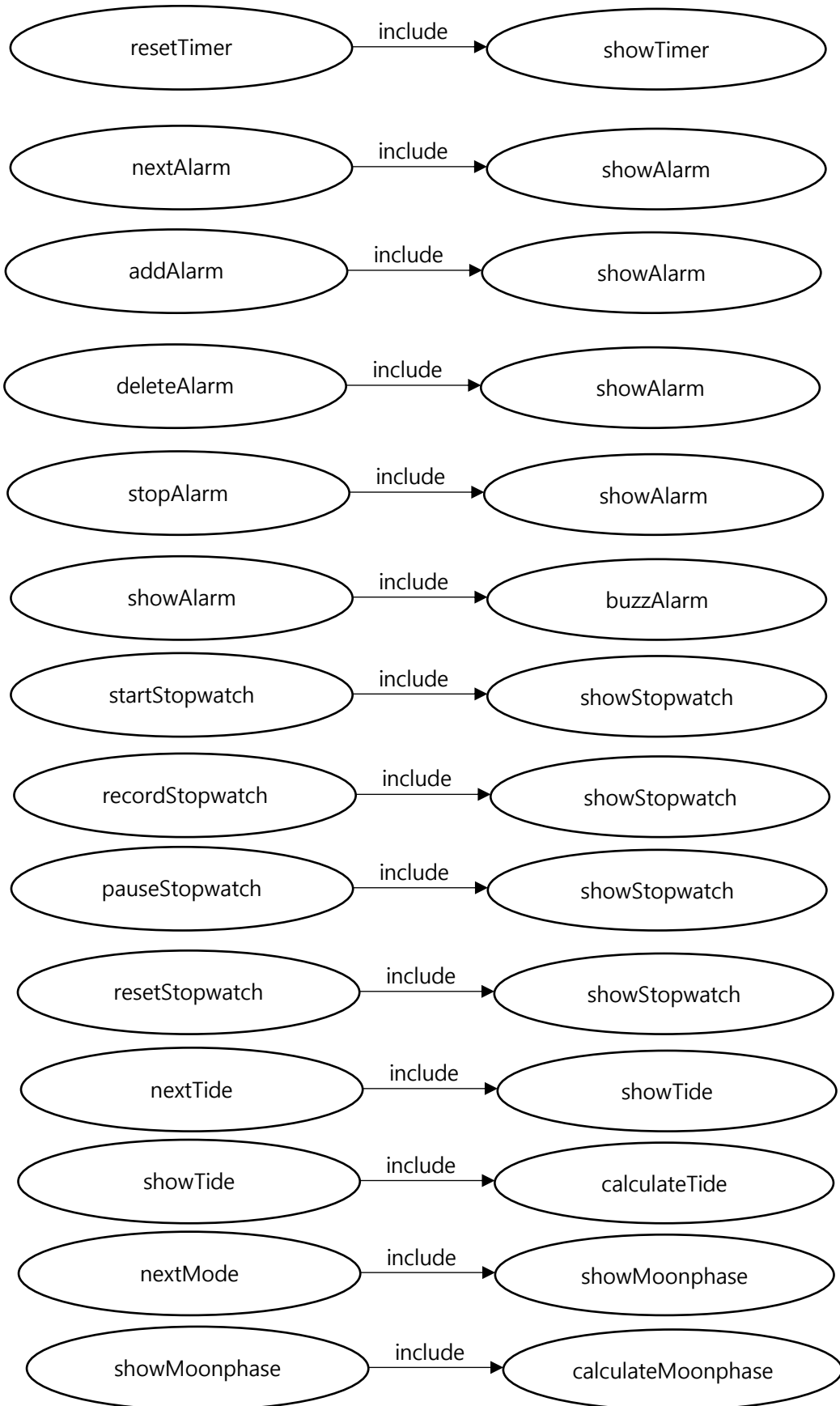
4. Categorize Use Case

R 0.0	showTime	primary
R 0.1	adjustTime	Primary
R 1.0	showTimer	Primary
R 1.1	setTimer	Primary
R 1.2	startTimer	Primary
R 1.3	pauseTimer	Primary
R 1.4	resetTimer	Primary
R 1.5	buzzTimer	Primary
R 2.1	showAlarm	Primary
R 2.2	nextAlarm	Primary
R 2.3	addAlarm	Primary
R 2.4	deleteAlarm	Primary
R 2.5	buzzAlarm	Primary
R 2.6	stopAlarm	Primary
R 3.1	showStopwatch	Primary

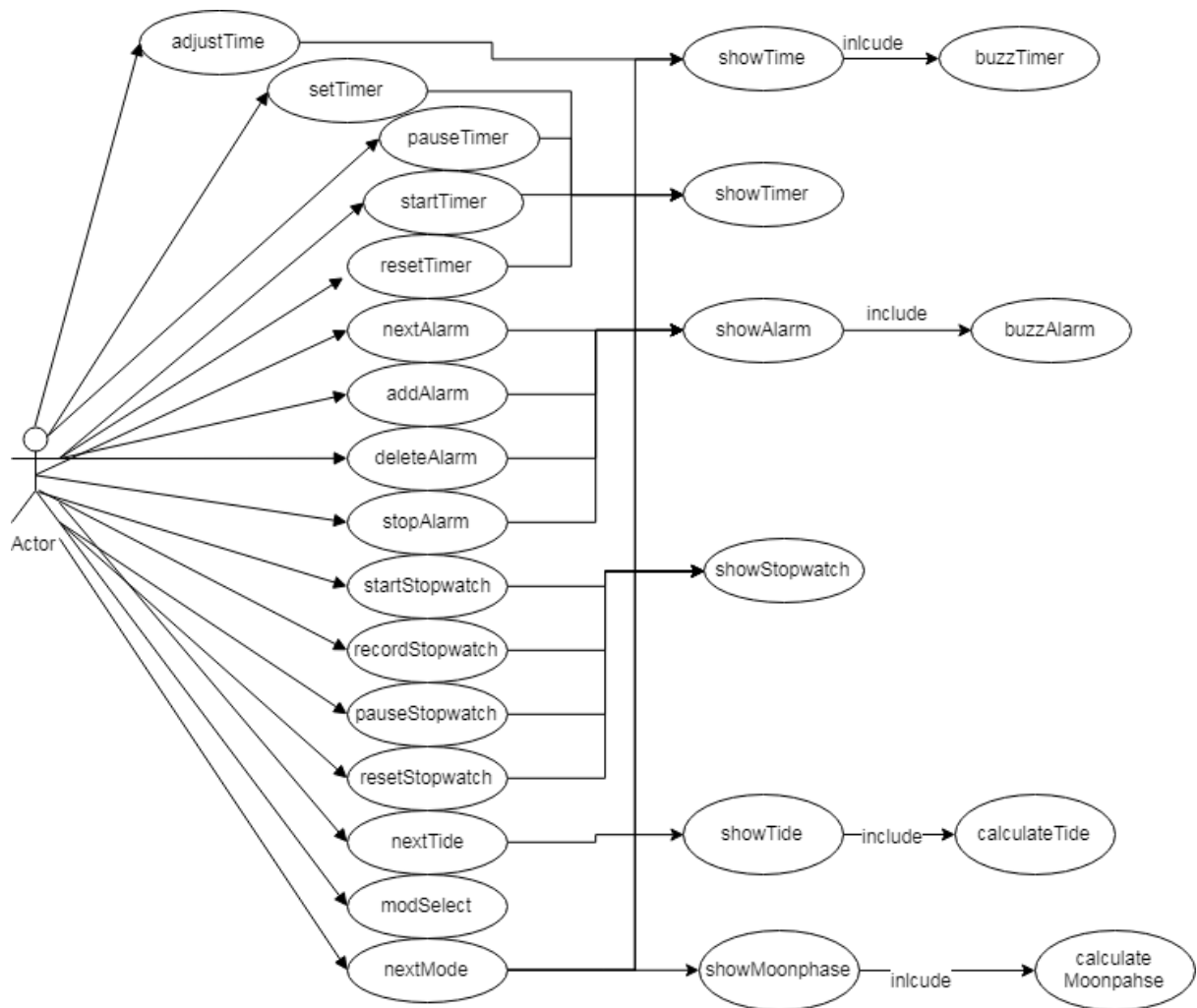
R 3.2	startStopwatch	Primary
R 3.3	recordStopwatch	Primary
R 3.4	pauseStopwatch	Primary
R 3.5	resetStopwatch	Primary
R 4.1	showTide	Primary
R 4.2	nextTide	primary
R 4.3	calculateTide	Primary
R 5.1	showMoonphase	Primary
R 5.2	calculateMoonphase	Primary
R 6.1	modeSelect	Primary
R 6.2	nextMode	Primary

5. Identify Relationship Between Use Cases





6. Draw A Use Case Diagram



7. Describe Use Case

Use Case	1. showTime
Actor	System
Description	TimeDB 를 통해 현재 시간을 보여준다

Use Case	2. adjustTime
Actor	User

Description	TimeDB 의 현재 시간을 설정한다 버튼을 누르면 차례대로 시, 분이 돌아가며 조정된다
-------------	--

Use Case	3. showTimer
Actor	System
Description	현재 타이머를 표시한다

Use Case	4. setTimer
Actor	User
Description	사용자가 입력한 시간의 타이머를 설정한다

Use Case	5. startTimer
Actor	User
Description	타이머를 시작한다

Use Case	6. pauseTimer
Actor	User
Description	타이머를 일시정지 시킨다

Use Case	7. resetTimer
Actor	User
Description	타이머를 일시정지 시킨다

Use Case	8. buzzTimer
Actor	System
Description	타이머가 0 이 되면 버저를 울린다

Use Case	9. showAlarm
Actor	System
Description	현재 설정된 알람을 표시한다

Use Case	10. nextAlarm
----------	---------------

Actor	User
Description	다음 알람을 불러온다

Use Case	11. addAlarm
Actor	User
Description	알람을 추가한다

Use Case	12. deleteAlarm
Actor	User
Description	알람을 지운다

Use Case	13. buzzAlarm
Actor	System
Description	TimeDB 를 참조해 부저를 울린다

Use Case	14. stopAlarm
Actor	User
Description	부저가 울린 알람을 정지시킨다

Use Case	15. showStopwatch
Actor	System
Description	현재 스탑워치를 출력한다

Use Case	16. startStopwatch
Actor	User
Description	스탑워치를 시작한다

Use Case	17. recordStopwatch
Actor	User
Description	record 버튼을 누르면 현재 스탑워치를 출력한다

Use Case	18. pauseStopwatch
Actor	User

Description	스탑워치를 정지시킨다
-------------	-------------

Use Case	19. resetStopwatch
Actor	User
Description	스탑워치를 0 초로 되돌린다

Use Case	20. showTide
Actor	System
Description	현재 Tide 정보를 표시한다

Use Case	21. nextTide
Actor	User
Description	다음 바다를 선택한다

Use Case	22. calculateTide
Actor	System
Description	조수간만의 차를 계산해 그래픽 정보로 바꾼다

Use Case	23. showMoonphase
Actor	System
Description	현재 Moonphase 정보를 출력한다

Use Case	24. calculateMoonphase
Actor	System
Description	달모양을 계산해 그래픽 정보로 바꾼다

Use Case	25. modeSelect
Actor	User
Description	User 가 6 개의 모드중 사용할 4 개를 선택한다

Use Case	26. nextMode
----------	--------------

Actor	User
Description	선택된 4 개 모드중 다음 모드로 넘어간다

9. Rank Use-Case

R 0.0	showTime	High
R 0.1	adjustTime	High
R 1.0	showTimer	High
R 1.1	setTimer	High
R 1.2	startTimer	High
R 1.3	pauseTimer	High
R 1.4	resetTimer	High
R 1.5	buzzTimer	High
R 2.1	showAlarm	High
R 2.2	nextAlarm	High
R 2.3	addAlarm	High
R 2.4	deleteAlarm	High
R 2.5	buzzAlarm	High
R 2.6	stopAlarm	High
R 3.1	showStopwatch	High
R 3.2	startStopwatch	High
R 3.3	recordStopwatch	High
R 3.4	pauseStopwatch	High
R 3.5	resetStopwatch	High

R 4.1	showTide	High
R 4.2	nextTide	High
R 4.3	calculateTide	High
R 5.1	showMoonphase	High
R 5.2	calculateMoonphase	High
R 6.1	modeSelect	High
R 6.2	nextMode	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	showTime	시간이 정상적으로 출력되는지 확인한다 오차가 최대한 적은지 확인한다 가독성이 나쁘지 않은지 확인한다
R 0.1	adjustTime	시간이 정상적으로 변경이 되었는지 확인한다
R 1.0	showTimer	현재 타이머를 잘 표시해야 한다
R 1.1	setTimer	타이머가 잘 설정이 되었는지 확인한다
R 1.2	startTimer	버튼을 누르면 타이머가 잘 시작해야 한다
R 1.3	pauseTimer	버튼을 누르면 타이머가 잘 멈춰야 한다

R 1.4	resetTimer	타이머가 잘 초기화 되는지 확인한다
R 1.5	buzzTimer	정확히 0 초에 버저가 울려야 한다
R 2.1	showAlarm	사용자가 설정한 알람이 맞아야 한다
R 2.2	nextAlarm	사용자가 설정한 다음번 알람을 불러와야 한다
R 2.3	addAlarm	알람이 정확히 추가가 되어야 한다
R 2.4	deleteAlarm	알람이 확실하게 삭제되어야 한다
R 2.5	buzzAlarm	사용자가 설정한 알람이 정확한 시간에 울려야한다
R 2.6	stopAlarm	알람이 제대로 정지해야 한다
R 3.1	showStopwatch	현재 스탑워치를 정확히 표시한다
R 3.2	startStopwatch	버튼을 누르면 시간이 잘 가기 시작하는지 확인한다.
R 3.3	recordStopwatch	버튼을 누른 때의 시간이 잘 기록되는지 확인한다.
R 3.4	pauseStopwatch	버튼을 누르면 시간이 잘 멈추는지 확인한다.
R 3.5	resetStopwatch	버튼을 누르면 시간이 잘 초기화되는지 확인한다.
R 4.1	showTide	물높이가 잘 출력되는지 확인한다
R 4.2	nextTide	다음 바다의 물높이가 잘 출력되는지 확인한다
R 4.3	calculateTide	물높이가 잘 계산되는지 확인한다. 오차가 적은지 확인한다
R 5.1	showMoonphase	달의 위상을 정확히 표시하는지 확인한다
R 5.2	calculateMoonphase	달의 위상을 정확히 계산하는지 확인한다 오차가 적은지 확인한다
R 6.1	modeSelect	선택한 기능만이 잘 사용가능상태로 있게 되는지 확인한다
R 6.2	nextMode	다음 모드로 바로 넘어가야 한다

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)
 - Mode(기능 선택)

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			■	■	■						

2000 Build	2030 Analyze	2031. Define Essential Use Cases																	
		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																			
	2051. Implement Class & Methods Definitions																		
	2052. Implement Windows																		

	2050 Construct	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															