

**2019 Software Modeling & Analysis**  
**OOPT Stage 2030.**  
**【 Analyze 】**

**골라보 시계**

**Team #1**

**201411273 박재범**

**201411275 박진호**

**201411311 장원영**

**201311313 정인원**

# Contents

**2031. Define Essential Use Cases**

**2032. Define Domain Model**

**2033. Refine Glossary**

**2034. Define System Sequence Diagrams**

**2035. Define Operation Contracts**

**2036. Define State Diagrams**

**2037. Refine System Test Case**

**2038. Analyze(2030) Traceability Analysis**

## 2031. Define Essential Use Cases

<b>Use Case</b>	Set Time
<b>Actor</b>	User
<b>Purpose</b>	시간을 설정한다.
<b>Overview</b>	User가 funct, start, stop 키를 이용해 시간을 맞춘다.
<b>Type</b>	Evident
<b>Cross Reference</b>	Functional Requirements ; R.1.1, R.1.2
<b>Pre-Requisites</b>	Mode가 Time-Keeping인 상태여야 한다.
<b>Typical Courses Of Events</b>	<p>(A) : Actor, (S) : System</p> <ol style="list-style-type: none"> <li>1. (A) : funct버튼을 2초간 누른다.</li> <li>2. (S) : am, pm 표시부분을 깜빡인다.</li> <li>3. (A) : start버튼을 눌러 am, pm, 24hour 포맷을 설정한다. 원하는 포맷을 설정 후 funct버튼을 누른다.</li> <li>4. (S) : 시간 설정모드로 넘어간다. 설정할 구간을 깜빡인다.</li> <li>5. (A) : start(+)버튼과 reset(-)버튼을 이용해서 시간을 설정한다. 구간 설정 후 funct버튼을 눌러 다음 구간으로 넘어간다. -&gt;분(1), 분(10), 시(1), 시(10), 일(1), 일(10), 월, 요일 순으로 설정을 할 수 있다.</li> <li>6. (S) : 다음으로 설정할 구간의 숫자를 깜빡인다.</li> <li>7. (A) : 마지막 설정을 마치고 funct버튼을 누른다.</li> <li>8. (S) : 설정 값을 저장하고 저장된 시간을 표시한다.</li> </ol>
<b>Alternative Courses Of Events</b>	N/A
<b>Exceptional Courses Of Events</b>	<p>E6. 12시간 포맷일 경우 12시를 넘길 수 없음 E8. 언제든지 funct버튼을 2초간 누르면 그때까지 설정한 값을 저장한다.</p>

<b>Use Case</b>	Display Time
<b>Actor</b>	System
<b>Purpose</b>	시간을 화면에 표시한다.
<b>Overview</b>	화면에 시간, 남쪽, 날짜 등을 표시한다.
<b>Type</b>	Hidden
<b>Cross Reference</b>	Functional Requirements ; R.1.2 R.1.1
<b>Pre-Requisites</b>	mode가 Time-Keeping 상태여야 한다.
<b>Typical Courses Of Events</b>	(A) : Actor, (S) : System 1. (S) : Set Time으로 설정된 시간을 화면에 표시한다.
<b>Alternative Courses Of Events</b>	N/A
<b>Exceptional Courses Of Events</b>	N/A

<b>Use Case</b>	Set Timer
<b>Actor</b>	User
<b>Purpose</b>	Timer 시간을 설정한다.
<b>Overview</b>	User가 일정시간을 정하고 입력하여 Timer를 설정한다.
<b>Type</b>	Evident
<b>Cross Reference</b>	Functional Requirements ; R.2.1, R.2.2
<b>Pre-Requisites</b>	Mode가 Timer 상태여야 한다. Select Function에 Timer가 ON 상태여야 한다.
<b>Typical Courses Of Events</b>	(A) : Actor, (S) : System 1. (A) : funct버튼을 눌러 설정을 시작한다. 2. (S) : 설정할 구간의 숫자를 깜빡인다. 3. (A) : start(+)버튼과 reset(-)버튼을 이용해서 시간을 설정한다. 4. (A) : funct버튼을 이용해서 다음 구간으로 넘어간다. ->초(1) 초(10) 분(1) 분(10) 시(1) 시(10) 순서대로 설정할 수 있다. 5. (S) : 설정할 구간의 숫자를 깜빡인다. 6. (A) : 이 과정을 시(10)까지 마친 후 funct키를 누르면 끝이 난다. 7. (A) : 위 과정 중 언제든 start버튼을 누르면 Timer가 시작된다. 8. (S) : start버튼이 눌린 순간까지 설정된 시간을 토대로 Timer의 시간이 줄어든다.
<b>Alternative Courses Of Events</b>	N/A
<b>Exceptional Courses Of Events</b>	E3, E6. 설정이 99시간 59분 59초를 넘길 수 없다.

<b>Use Case</b>	Beep Timer Buzzer
<b>Actor</b>	System
<b>Purpose</b>	Buzzer를 울린다.
<b>Overview</b>	User가 설정한 시간이 지나면 System에서 5초간 Buzzer를 울린다.
<b>Type</b>	Hidden
<b>Cross Reference</b>	Functional Requirements ; R.2.2, R.2.1, R.2.3
<b>Pre-Requisites</b>	설정된 시간이 모두 지난 상태여야 한다. Select Function에 Timer가 ON 상태여야 한다.
<b>Typical Courses Of Events</b>	(A) : Actor, (S) : System 1. (S) : 설정된 Timer가 끝나면 Buzzer를 울린다.
<b>Alternative Courses Of Events</b>	N/A
<b>Exceptional Courses Of Events</b>	N/A

<b>Use Case</b>	Beep Timer Buzzer
<b>Actor</b>	User
<b>Purpose</b>	Buzzer를 멈춘다.
<b>Overview</b>	Buzzer를 User가 버튼을 눌러 즉시 멈춘다.
<b>Type</b>	Evident
<b>Cross Reference</b>	Functional Requirements ; R.2.3, R.2.2
<b>Pre-Requisites</b>	Buzzer가 울리는 상태여야 한다. Select Function에 Timer가 ON 상태여야 한다.
<b>Typical Courses Of Events</b>	(A) : Actor, (S) : System 1. (A) : mode버튼을 제외한 버튼을 아무것이나 누른다. 2. (S) : 울리고 있는 Buzzer를 즉시 멈춘다.
<b>Alternative Courses Of Events</b>	N/A
<b>Exceptional Courses Of Events</b>	N/A

<b>Use Case</b>	Beep Timer Buzzer
<b>Actor</b>	User
<b>Purpose</b>	Buzzer를 멈춘다.
<b>Overview</b>	Buzzer를 User가 버튼을 눌러 즉시 멈춘다.
<b>Type</b>	Evident
<b>Cross Reference</b>	Functional Requirements ; R.2.3, R.2.2
<b>Pre-Requisites</b>	Buzzer가 울리는 상태여야 한다. Select Function에 Timer가 ON 상태여야 한다.
<b>Typical Courses Of Events</b>	(A) : Actor, (S) : System 1. (A) : mode버튼을 제외한 버튼을 아무것이나 누른다. 2. (S) : 울리고 있는 Buzzer를 즉시 멈춘다.
<b>Alternative Courses Of Events</b>	N/A
<b>Exceptional Courses Of Events</b>	N/A

<b>Use Case</b>	Store Lap & Split
<b>Actor</b>	User
<b>Purpose</b>	Lap과 Split을 저장한다.
<b>Overview</b>	Stopwatch가 시작된 상태에서 funct키를 눌러 Lap과 Split을 저장한다.
<b>Type</b>	Evident
<b>Cross Reference</b>	Functional Requirements ; R.3.2
<b>Pre-Requisites</b>	Stopwatch가 작동중인 상태여야 한다. Select Function에 Stopwatch가 ON 상태여야 한다.
<b>Typical Courses Of Events</b>	(A) : Actor, (S) : System 1. (A) : funct키를 누른다. 2. (S) : 지금까지 흐른 시간을 저장한다. 3. (A) : funct키를 누른다. (반복) 4. (S) : 이전에 저장한 시간과 현재까지 흐른 시간의 간격을 저장한다.
<b>Alternative Courses Of Events</b>	N/A
<b>Exceptional Courses Of Events</b>	E 4. 30회 이상 수행할 시 데이터가 저장되지 않음.

<b>Use Case</b>	Pause Stopwatch
<b>Actor</b>	User
<b>Purpose</b>	Stopwatch를 일시 정지 및 다시 시작한다.
<b>Overview</b>	Stopwatch가 작동중인 동안 start버튼을 눌러 정지시킨다. 정지 상태에서 다시 start버튼을 누르면 다시 시작한다.
<b>Type</b>	Evident
<b>Cross Reference</b>	Functional Requirements ; R.3.3
<b>Pre-Requisites</b>	Stopwatch가 작동중인 상태여야 한다. Select Function에 Stopwatch가 ON 상태여야 한다.
<b>Typical Courses Of Events</b>	(A) : Actor, (S) : System 1. (A) : start버튼을 누른다. 2. (S) : Stopwatch를 정지시킨다. 3. (A) : start버튼을 다시 누른다. 4. (S) : Stopwatch가 정지된 시간부터 다시 시작된다.
<b>Alternative Courses Of Events</b>	N/A
<b>Exceptional Courses Of Events</b>	N/A

<b>Use Case</b>	Check Lap & Split
<b>Actor</b>	User
<b>Purpose</b>	현재까지 저장된 Lap과 Split 시간을 확인한다.
<b>Overview</b>	funcnt버튼을 2초간 누르고 있을 경우 저장된 Lap, Split 데이터를 확인할 수 있다.
<b>Type</b>	Evident
<b>Cross Reference</b>	Functional Requirements ; R.3.4
<b>Pre-Requisites</b>	Lap & Split 데이터가 존재해야한다. Stopwatch가 일시정지 상태여야 한다. Select Function에 Stopwatch가 ON 상태여야 한다.
<b>Typical Courses Of Events</b>	(A) : Actor, (S) : System 1. (A) : funcnt버튼을 2초간 누른다. 2. (S) : 날짜를 표시하는 부분에 현재 확인중인 데이터의 번호를 표시하고 시간 부분에 데이터의 값을 표시한다. 3. (A) : start(다음)버튼과 reset(이전)버튼을 이용해 원하는 데이터를 확인한다. ->표시 순서는 1번 lap->1번 split->2번 lap->2번 split->.....이다. 4. (S) : 날짜를 표시하는 부분에 현재 확인중인 데이터의 번호를 표시하고 시간 부분에 데이터의 값을 표시한다. 5. (A) : 다시 stopwatch를 사용하고 싶은 경우 funcnt버튼을 누른다. 6. (S) : 일시정지상태였던 stopwatch를 화면을 표시한다.
<b>Alternative Courses Of Events</b>	N/A
<b>Exceptional Courses Of Events</b>	E2. 저장된 데이터가 없을 경우 실행을 제한한다.



<b>Use Case</b>	Reset Stopwatch
<b>Actor</b>	User
<b>Purpose</b>	Stopwatch의 데이터를 초기화한다.
<b>Overview</b>	reset버튼을 눌러 시간과 저장된 Lap, Split 데이터를 초기화한다.
<b>Type</b>	Evident
<b>Cross Reference</b>	Functional Requirements ; R.3.5
<b>Pre-Requisites</b>	Stopwatch가 일시정지 상태여야 한다. Select Function에 Stopwatch가 ON 상태여야 한다.
<b>Typical Courses Of Events</b>	(A) : Actor, (S) : System 1. (A) : reset버튼을 누른다. 2. (S) : 현재까지 측정한 시간을 0으로 초기화하고 lap, split 데이터 도 모두 초기화한다.
<b>Alternative Courses Of Events</b>	N/A
<b>Exceptional Courses Of Events</b>	E2. Stopwatch가 작동 중에는 reset버튼을 눌러도 초기화하지 않는다.

<b>Use Case</b>	Set Alarm
<b>Actor</b>	User
<b>Purpose</b>	Alarm을 설정한다.
<b>Overview</b>	User가 원하는 시간에 알람이 울리게 시간을 설정한다.
<b>Type</b>	Evident
<b>Cross Reference</b>	Functional Requirements ; R.4.1 R.4.2
<b>Pre-Requisites</b>	Mode가 Alarm 상태여야 한다. Select Function에 Alarm가 ON 상태여야 한다.
<b>Typical Courses Of Events</b>	(A) : Actor, (S) : System (A) : Actor, (S) : System 1. (A) : Mode버튼으로 Alarm모드에 진입한다. 2. (S) : 저장된 알람 리스트의 첫 번째 데이터를 표시한다. 3. (A) : start(다음)버튼과 reset(이전)버튼을 이용해서 리스트를 확인한다. 4. (S) : 다음이나 이전의 데이터를 화면에 표시한다. 5. (A) : funct버튼을 2초간 누른다. 6. (S) : 현재 표시된 데이터를 설정하는 화면을 표시한다.
<b>Alternative Courses Of Events</b>	N/A
<b>Exceptional Courses Of Events</b>	E2. 저장된 데이터가 없는 경우 바로 설정모드를 실행한다.

<b>Use Case</b>	Beep Alarm Buzzer
<b>Actor</b>	System
<b>Purpose</b>	Alarm버저를 울린다.
<b>Overview</b>	User가 설정된 시간이 되면 System이 버저를 울린다.
<b>Type</b>	Hidden
<b>Cross Reference</b>	Functional Requirements ; R.4.2 R.4.1 R.4.3 R.4.4
<b>Pre-Requisites</b>	Alarm Set으로 설정된 시간에 도달해야 한다. Select Function에 Stopwatch가 ON 상태여야 한다.
<b>Typical Courses Of Events</b>	(A) : Actor, (S) : System 1. (S) : 설정된 시간에 도달할 경우 1분간 버저를 울린다.
<b>Alternative Courses Of Events</b>	N/A
<b>Exceptional Courses Of Events</b>	E1. 시간을 설정하는 동안에는 버저가 울리지 않는다.

<b>Use Case</b>	Stop Alarm Buzzer
<b>Actor</b>	User
<b>Purpose</b>	Buzzer를 멈춘다.
<b>Overview</b>	Buzzer가 울리는 동안 start나 reset버튼을 눌러 멈춘다.
<b>Type</b>	Evident
<b>Cross Reference</b>	Functional Requirements ; R.4.3 R.4.2
<b>Pre-Requisites</b>	Buzzer가 울리는 상태여야 한다. Select Function에 Stopwatch가 ON 상태여야 한다.
<b>Typical Courses Of Events</b>	(A) : Actor, (S) : System 1. (A) : start나 reset버튼을 누른다. 2. (S) : 울리고 있는 버저를 즉시 멈춘다.
<b>Alternative Courses Of Events</b>	N/A
<b>Exceptional Courses Of Events</b>	E1. Timer와 다르게 mode, funct키를 누를 경우 Snooze가 된다.

<b>Use Case</b>	Set Snooze
<b>Actor</b>	User
<b>Purpose</b>	Snooze기능을 사용한다.
<b>Overview</b>	User가 mode나 funct버튼을 눌러 버저를 즉시 멈추고 5분 후에 다시 울리게 한다.
<b>Type</b>	Evident
<b>Cross Reference</b>	Functional Requirements ; R.4.4 R.4.2
<b>Pre-Requisites</b>	Buzzer가 울리는 상태여야 한다. Select Function에 Stopwatch가 ON 상태여야 한다.
<b>Typical Courses Of Events</b>	(A) : Actor, (S) : System 1. (A) : mode나 funct버튼을 누른다. 2. (S) : 버저를 즉시 멈추고 5분후에 다시 울리도록 설정한다.
<b>Alternative Courses Of Events</b>	N/A
<b>Exceptional Courses Of Events</b>	N/A

<b>Use Case</b>	Set Scope
<b>Actor</b>	User
<b>Purpose</b>	난수 생성 범위를 설정한다.
<b>Overview</b>	User가 난수 생성에 쓰일 범위를 설정한다.
<b>Type</b>	Evident
<b>Cross Reference</b>	Functional Requirements ; R.5.1 R.5.2
<b>Pre-Requisites</b>	Select Function에 Random Number Generator가 ON 상태여야 한다.
<b>Typical Courses Of Events</b>	(A) : Actor, (S) : System 1. (A) : Random Number Generator 모드에 진입한다. 2. (S) : 범위를 설정할 수 있도록 화면에 표시한다. 3. (A) : start(+)버튼과 reset(-)버튼을 이용해서 범위를 설정한다. 구간의 설정이 완료되면 funct로 다음 자릿수 설정으로 넘어간다. ->6자리로 1부터 999999까지 범위를 정할 수 있다. 4. (S) : 다음 구간의 숫자를 깜빡인다. 5. (A) : 마지막 구간까지 설정을 완료하고 funct키를 누른다. 6. (S) : 설정된 숫자를 화면에 표시한다.
<b>Alternative Courses Of Events</b>	N/A
<b>Exceptional Courses Of Events</b>	N/A

<b>Use Case</b>	Generate Random Number
<b>Actor</b>	User
<b>Purpose</b>	난수를 생성한다.
<b>Overview</b>	Set Scope에 의해 정해진 범위 내에서 난수를 생성한다.
<b>Type</b>	Evident
<b>Cross Reference</b>	Functional Requirements ; R.5.2 R5.1
<b>Pre-Requisites</b>	Select Function에 Random Number Generator가 ON 상태여야 한다. Set Scope의 설정 값이 1이상 이어야 한다.
<b>Typical Courses Of Events</b>	(A) : Actor, (S) : System 1. (A) : start버튼을 누른다. 2. (S) : 설정된 범위 내에서 난수를 생성한다. 3. (A) : reset버튼을 누른다. 4. (S) : 다시 범위를 설정할 수 있도록 화면을 표시한다.
<b>Alternative Courses Of Events</b>	E2. 1~60의 범위의 경우 임의방향 생성 기능을 이용할 수 있다.
<b>Exceptional Courses Of Events</b>	Set Scope의 설정 값이 0일 경우 작동하지 않는다.

<b>Use Case</b>	Generate Random Direction
<b>Actor</b>	User
<b>Purpose</b>	임의의 방향을 표시한다.
<b>Overview</b>	User가 버튼을 눌러 임의의 방향을 생성하는 기능을 실행한다.
<b>Type</b>	Evident
<b>Cross Reference</b>	Functional Requirements ; R.6.1
<b>Pre-Requisites</b>	Select Function에 Random Direction Generator가 ON 상태여야 한다.
<b>Typical Courses Of Events</b>	(A) : Actor, (S) : System 1. (A) : start 버튼을 누른다. 2. (S) : 1~60 범위의 임의의 수를 생성하고 해당 위치에 맞게 테두리 화면에 결과를 출력한다.
<b>Alternative Courses Of Events</b>	E2. 1~60의 범위의 경우 난수 생성 기능을 이용할 수 있다.
<b>Exceptional Courses Of Events</b>	N/A

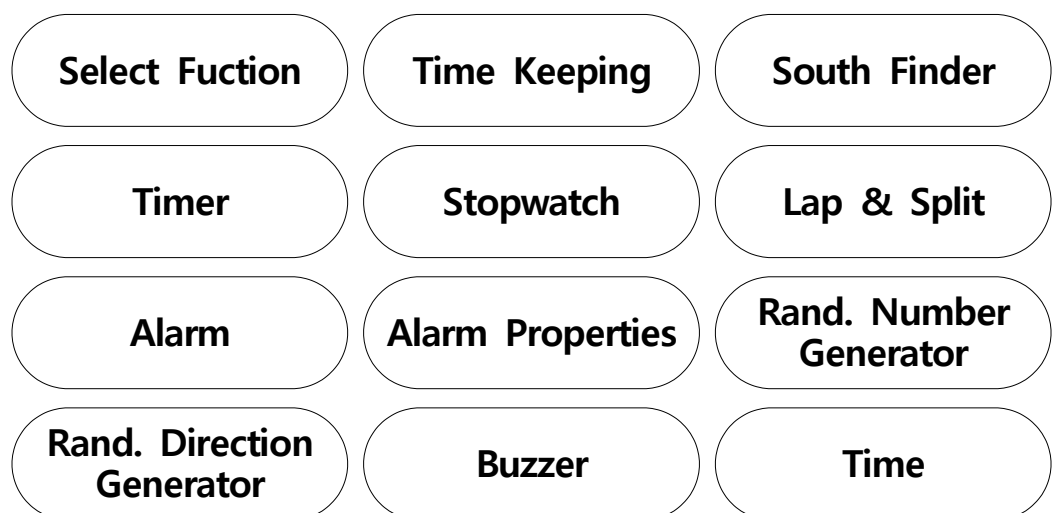
<b>Use Case</b>	Select Function
<b>Actor</b>	User
<b>Purpose</b>	기능을 선택한다.
<b>Overview</b>	mode버튼을 활용하여 기능을 선택하거나 on off할 기능을 정한다.
<b>Type</b>	Evident
<b>Cross Reference</b>	Functional Requirements ; R.6.1
<b>Pre-Requisites</b>	N/A
<b>Typical Courses Of Events</b>	<p>(A) : Actor, (S) : System</p> <ol style="list-style-type: none"> <li>1. (A) : mode버튼을 2초간 누른다.</li> <li>2. (S) : Time Keeping을 제외한 5개의 기능을 on off 하는 설정 화면을 표시한다.</li> <li>3. (A) : start(on)버튼 또는 reset(off)버튼을 누른다.</li> <li>4. (S) : 기능이 on 되거나 off 된다.</li> <li>5. (A) : funct버튼을 누른다.</li> <li>6. (S) : 다음기능을 표시한다.</li> </ol> <p>-&gt;Timer, Stopwatch, Alarm, Random Number Generator, Random Direction Generator 순으로 표시된다.</p> <ol style="list-style-type: none"> <li>7. (A) : start(on)버튼 또는 reset(off)버튼을 누른다.</li> <li>8. (S) : 기능이 on 되거나 off 된다.</li> </ol>
<b>Alternative Courses Of Events</b>	N/A
<b>Exceptional Courses Of Events</b>	<p>E4. off버튼은 기능이 on되어있는 상태에서만 작동한다.</p> <p>E6. 사이클을 도는 동안 3개를 선택하지 않으면 다시 Timer부터 선택을 시작한다.</p> <p>E8. 반복동안 3개의 기능이 on되면 자동으로 설정이 저장되고 Time Keeping 화면을 표시한다.</p>

## 2032. Define Domain Model

### 1. List Concepts from Use Cases or Business Concept Model



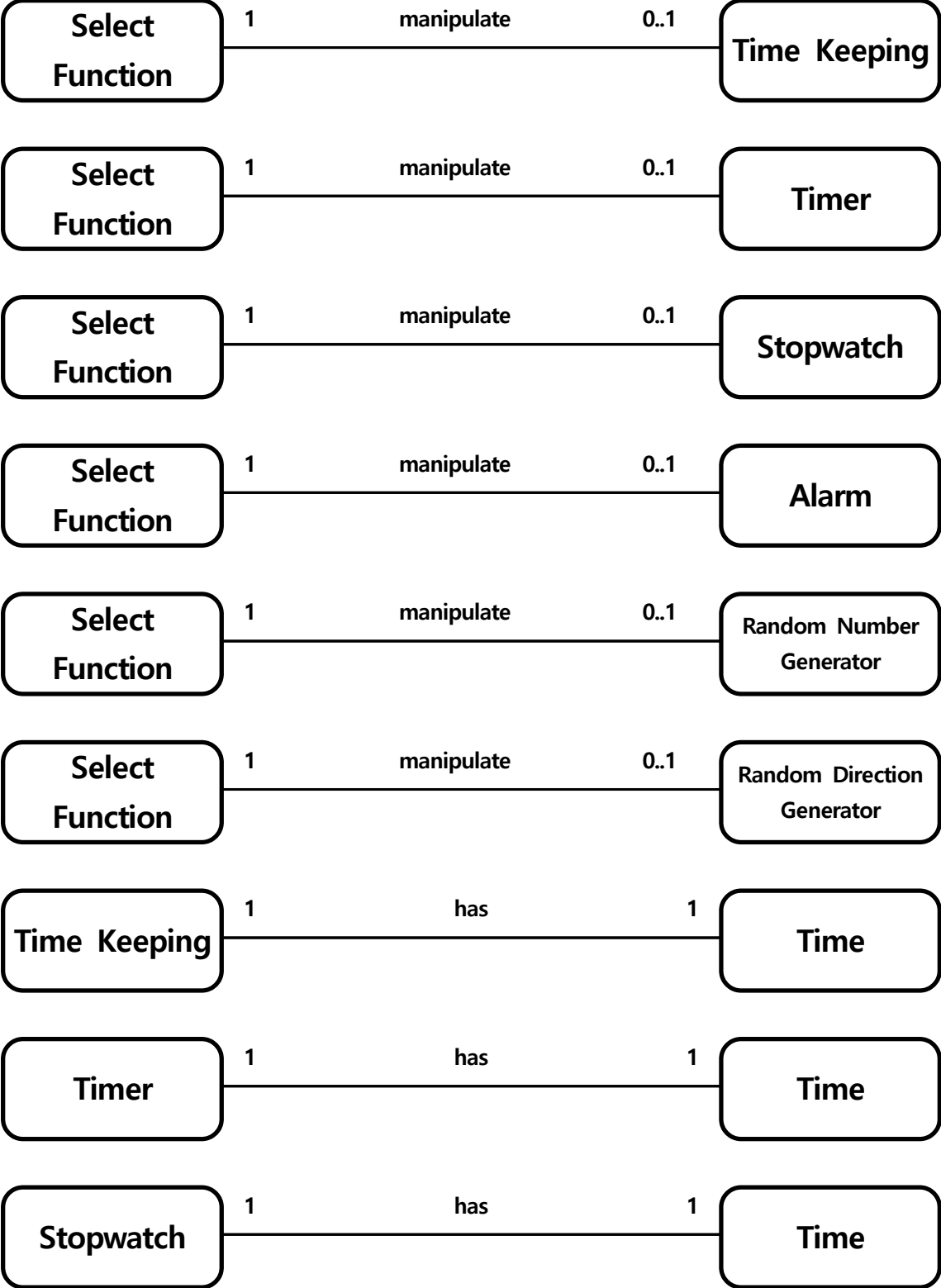
### 2. Assign Class Name into Concepts & Draw a Conceptual Class Diagram



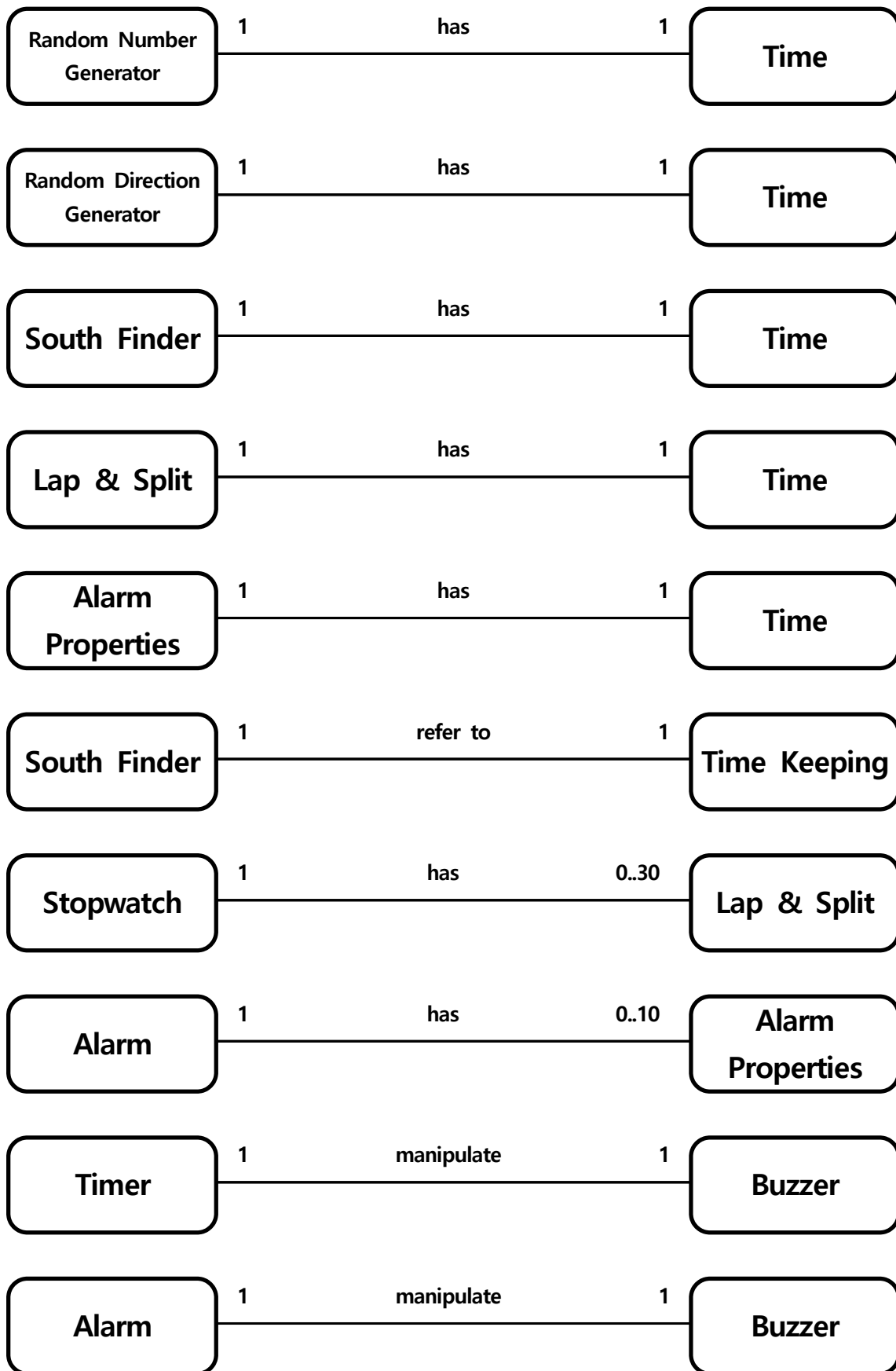
### 3. Identify Associations According to Association Categories

Association Category	Associations
A manipulate B	Select Function – Time Keeping Select Function – Timer Select Function – Stopwatch Select Function – Alarm Select Function – Random Number Generator Select Function – Random Direction Generator Timer – Buzzer Alarm – Buzzer
A refer to B	Time Keeping – South Finder
A has B	Time Keeping – Time South Finder – Time Stopwatch – Lap & Split Lap & Split – Time Timer – Time Alarm – Alarm Properties Alarm Properties – Time Random Number Generator – Time Random Direction Generator – Time

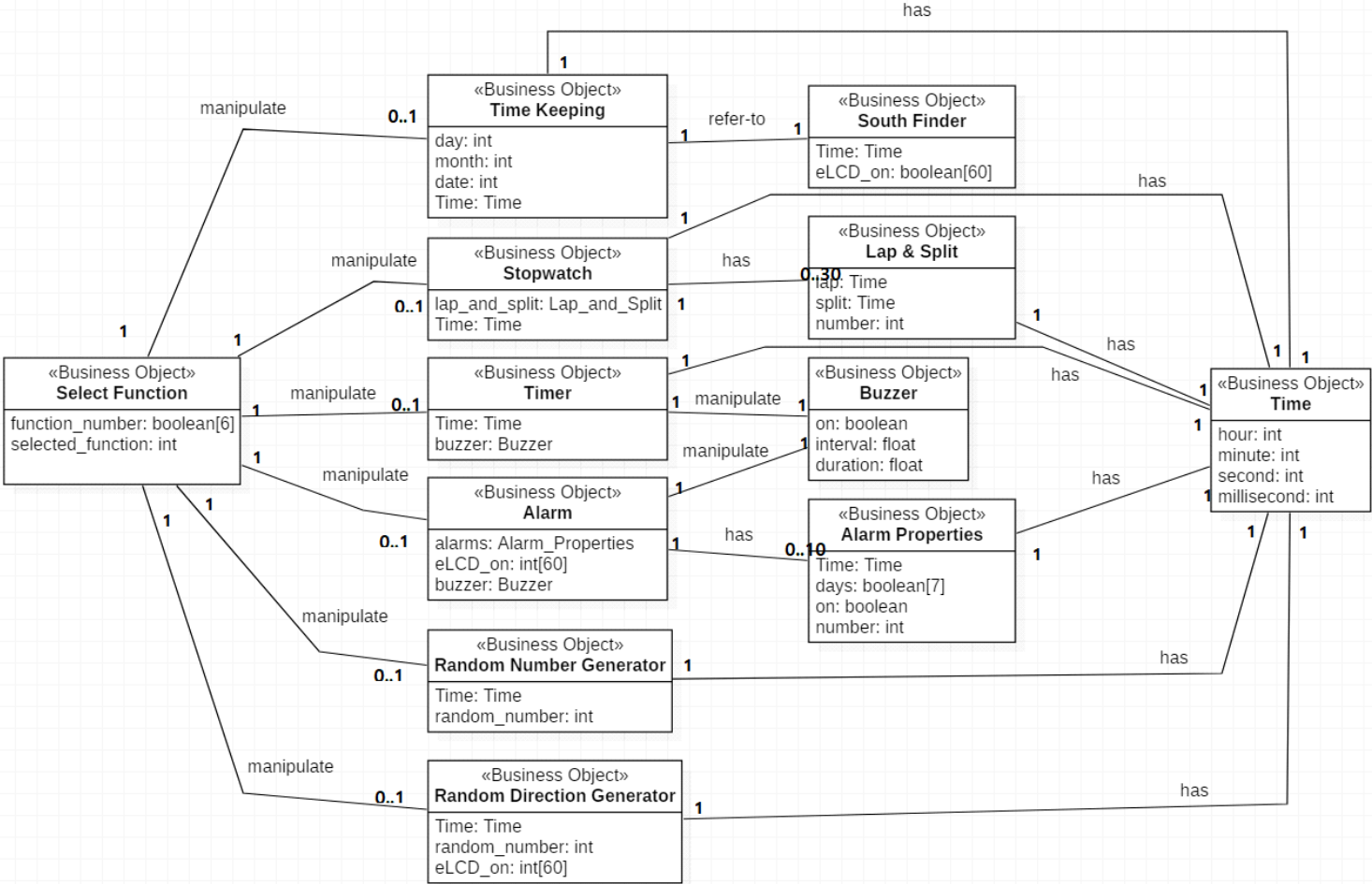
4. Add Role and Multiplicity







### 5. Draw a Conceptual Class Diagram and Add Attributes



## 2033. Refine Glossary

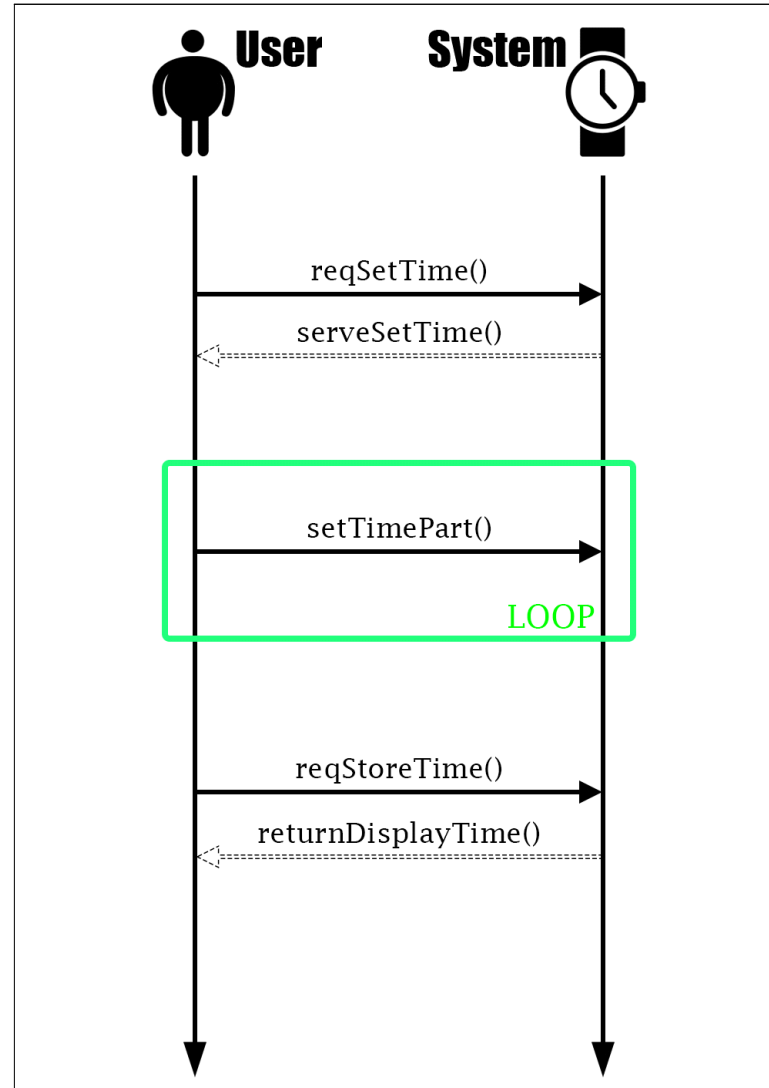
Glossary	Description
function_number: boolean[6]	function_number[0]~[5] 순서로 Time Keeping, Timer, Stopwatch, Alarm, Random Number Generator, Random Direction Generator 의 On/Off 여부를 의미하는 배열(1 is On).
selected_function: int	현재 사용중인 Function의 번호(0~5)를 의미하는 변수.
day: int	1(일요일)~7(월요일)의 값을 가지는 요일을 의미하는 변수.
month: int	1~12의 값을 가지는 월을 의미하는 변수.
date: int	1~31의 값을 가지는 일을 의미하는 변수.
Time: Time	시, 분, 초, 밀리초의 값을 저장할 수 있는 구조체(Class). Class에 따라 적절하게 사용된다.
eLCD_on: boolean[60]	테두리에 있는 60개 LCD의 On/Off 여부를 의미하는 배열. Class에 따라 적절하게 사용된다.
buzzer: Buzzer	버저의 On/Off 여부, 간격, 지속시간을 의미하는 구조체(Class). Class에 따라 적절하게 사용된다.
lap_and_split: Lap_And_Split	스톱워치의 Lap & Split을 저장하기 위한 구조체(Class).
lap: Time	Lap 시간을 저장하기 위한 구조체(Class).
split: Time	split 시간을 저장하기 위한 구조체(Class).
number: int //L&S	Lap & Split의 번호를 의미하는 변수.
alarms: Alarm_Properties	알람의 세부 속성들을 저장하기 위한 구조체(Class).
days: boolean[7]	특정 알람의 요일별 적용 여부(On/Off)를 의미하는 배열.
on: boolean	특정 알람 자체의 On/Off 여부를 의미하는 변수.
number: int //Alarm Properties	알람의 번호를 의미하는 변수.
random_number: int	생성된 난수를 저장하는 변수. Class에 따라 적절하게 사용된다.
on: boolean	버저의 On/Off 여부를 의미하는 변수.
interval: int	버저가 울리는 간격(밀리초)을 의미하는 변수.
duration: int	버저가 지속되는(밀리초) 시간을 의미하는 변수.
hour: int	시간을 의미하는 변수.
minute: int	분을 의미하는 변수.
second: int	초를 의미하는 변수.
milisecond: int	밀리초를 의미하는 변수.

## 2034. Define System Sequence Diagrams

Use Case	Name of Actor-Activated Event
1. Set Time	reqSetTime
	setTimePart
	reqStoreTime
2. Display Time	Hidden
3. Set Timer	reqSetTimer
	setTimerPart
	reqStartTimer
	reqResetTimer
4. Beep Timer Buzzer	Hidden
5. Stop Timer Buzzer	reqStopTimerBuzzer
6. Start Stopwatch	reqStartStopwatch
7. Store Lap & Split	reqStoreLnS
8. Pause Stopwatch	reqPauseStopwatch
	reqResumeStopwatch
9. Check Lap & Split	reqCheckLnS
	reqNextLnS
	reqBackToStopwatch
10. Reset Stopwatch	reqResetStopwatch
11. Set Alarm	reqSetAlarm
	reqDetailSet
	setAlarmPart
	reqStoreAlarm
	reqDeleteAlarm
12. Beep Alarm Buzzer	Hidden
13. Stop Alarm Buzzer	reqStopAlarmBuzzer
14. Set Snooze	reqSetSnooze
15. Set Scope	reqSetScope
16. Generate Random Number	reqRandomNumber
	reqBackToSetScope
17. Generate Random Direction	reqRandomDirection
18. Select Function	reqNextFuction
	reqSelectFuction
	setFunction

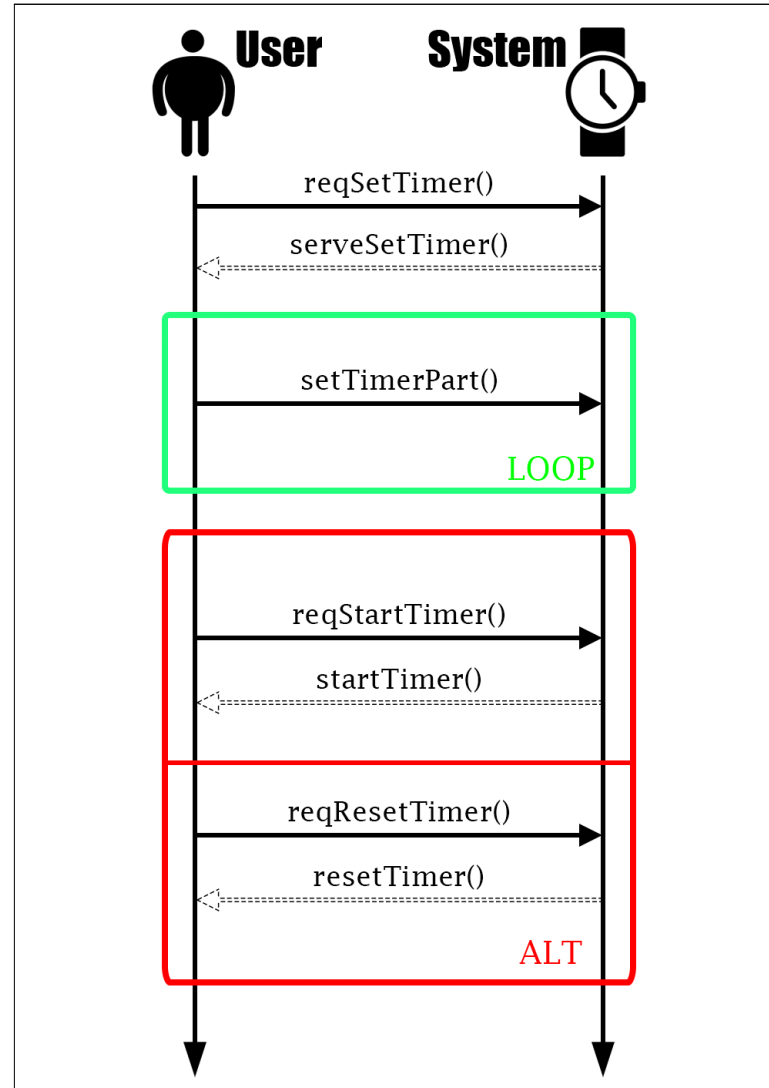
### Use Case: Set Time

1. 사용자가 Time Keeping 기능에서 Funct 버튼을 2초간 눌러 시간 설정 기능을 요청한다.
2. 시스템이 시간 설정 기능을 제공한다.
3. 12/24h 포맷, 분, 시, 일, 월, 요일에 대한 부분별 설정을 반복한다.
4. 사용자는 설정 도중 언제든지 Funct 버튼을 2초간 눌러 설정의 저장을 요청한다.
5. 시스템은 새로운 설정을 저장하고 시간을 띄우는 화면으로 전환시킨다.



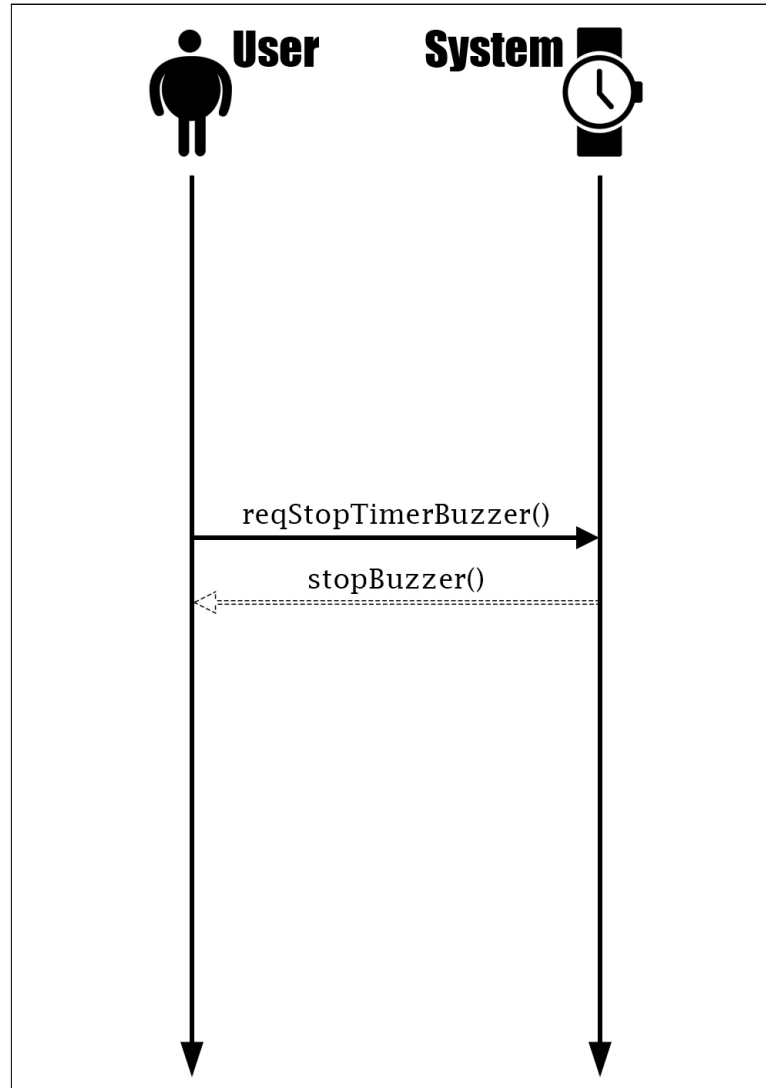
### Use Case: Set Timer

1. 사용자가 Timer 기능을 선택하여 타이머 설정 기능을 요청한다.
2. 시스템은 타이머 설정 기능을 제공한다.
3. 타이머의 초, 분, 시에 대한 부분별 설정을 반복한다.
- 4-a. 사용자가 설정 중 Start 버튼을 눌러 타이머 시작을 요청한다.
- 5-a. 시스템이 설정된 시간부터 타이머를 작동시킨다.
- 4-b. 사용자가 설정 중 Reset 버튼을 눌러 타이머 설정 초기화를 요청한다.
- 5-b. 시스템이 타이머 설정을 초기화한다.



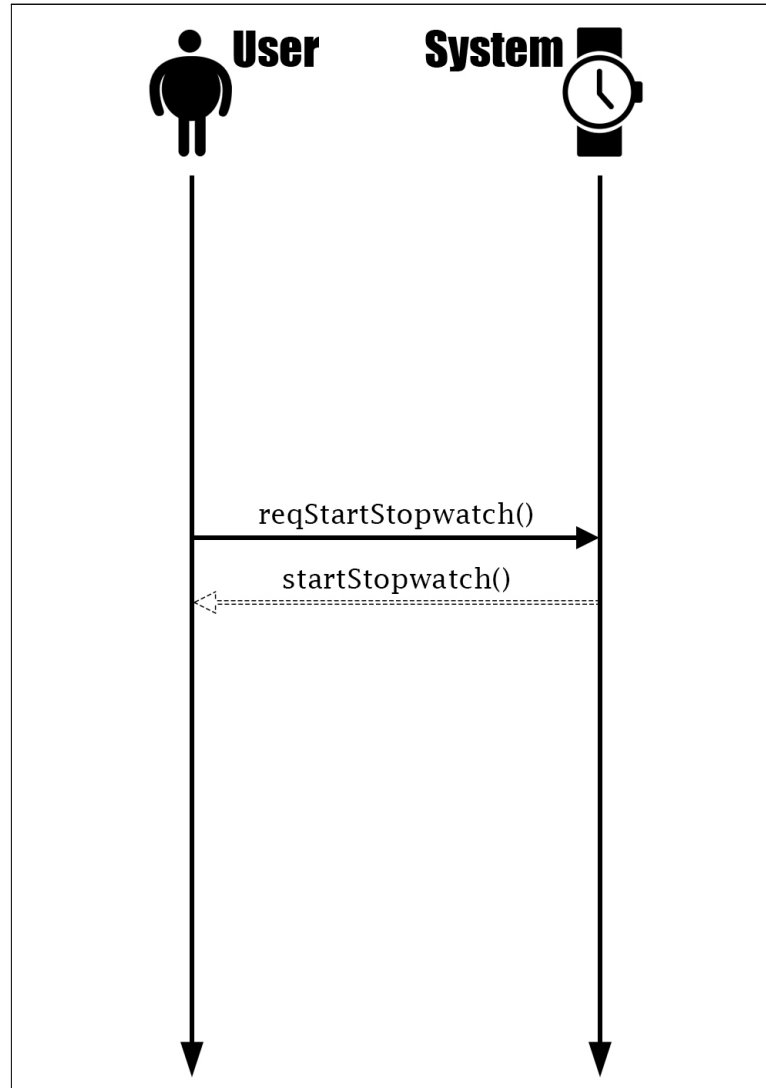
### Use Case: Stop Timer Buzzer

1. 타이머의 버저가 울릴 때 사용자가 아무 버튼이나 눌러 버저 멈춤을 요청한다.
2. 시스템이 버저를 즉시 멈춘다.



### Use Case: Start Stopwatch

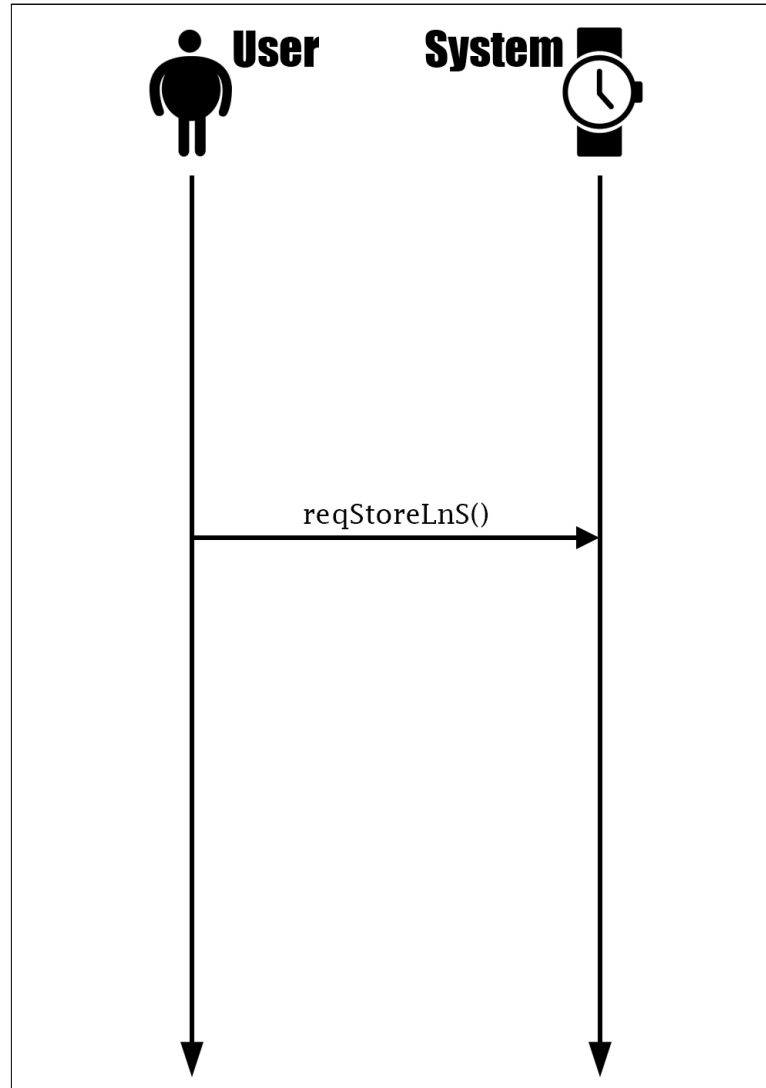
1. 사용자가 Stopwatch 기능을 선택한 상태로 Start 버튼을 눌러 스톱워치의 시작을 요청한다.
2. 시스템이 스톱워치를 작동시킨다.





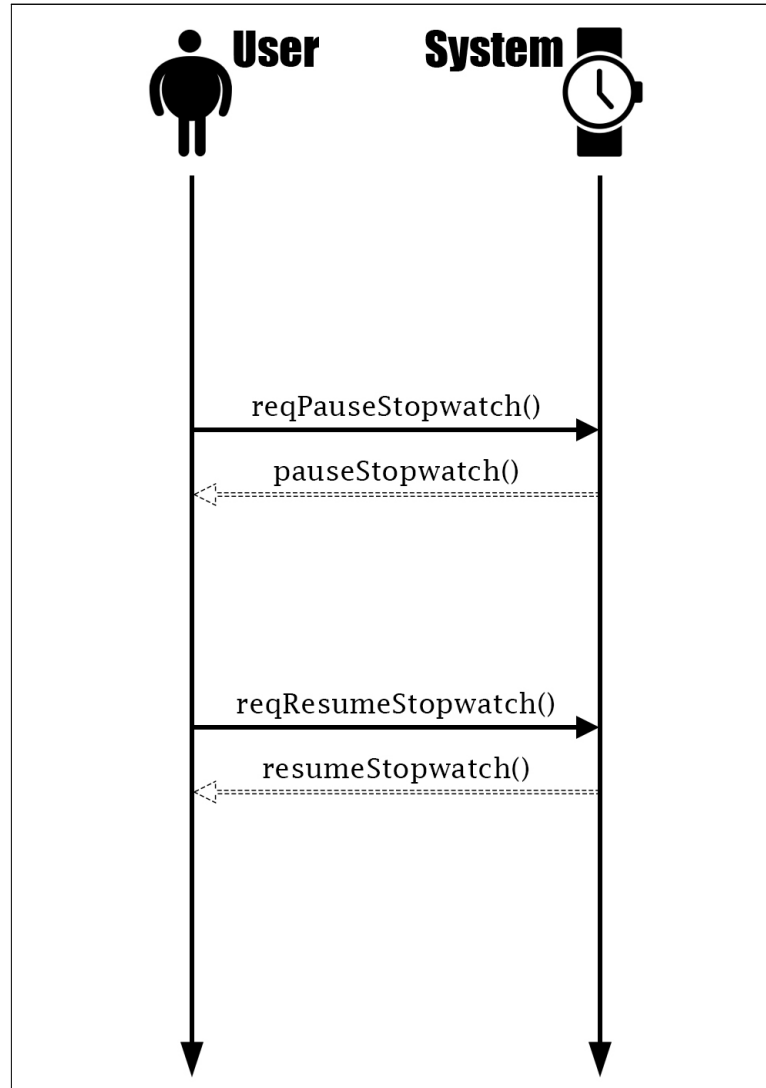
### Use Case: Store Lap & Split

1. 사용자가 스톱워치가 진행 중일 때 Funct 버튼을 눌러 Lap과 Split 시간을 저장할 것을 요청한다.
2. 시스템이 내부적으로 Lap과 Split을 저장한다.



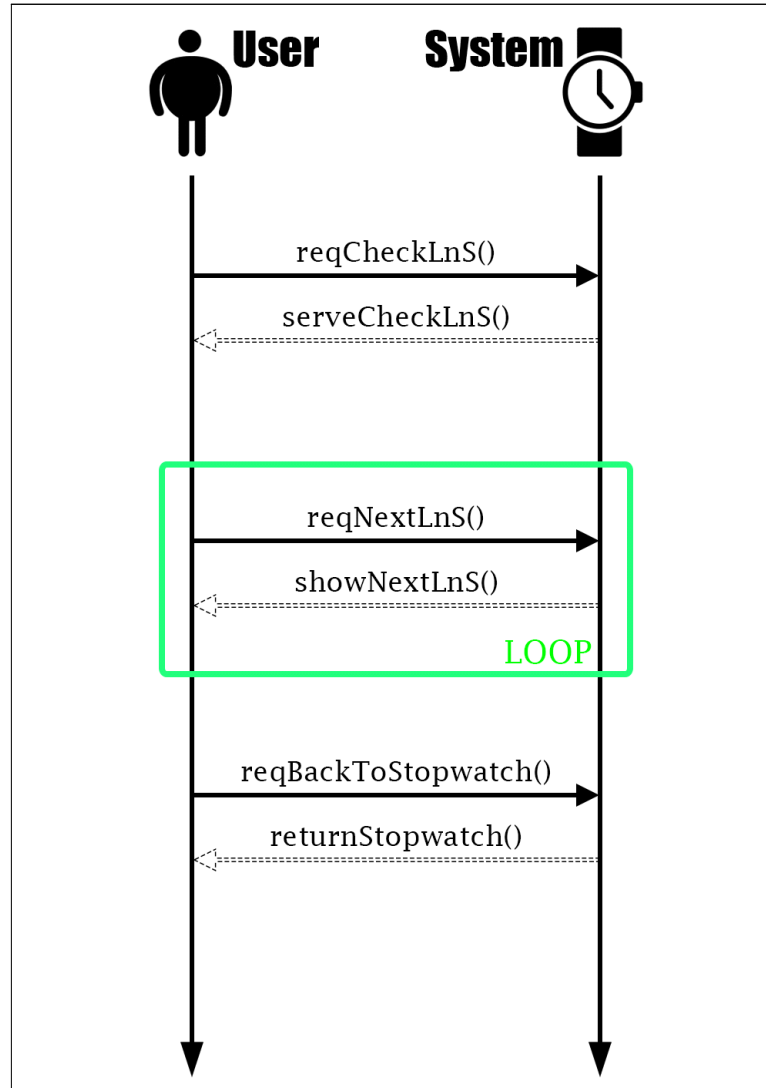
### Use Case: Pause Stopwatch

1. 사용자가 스톱워치가 진행 중일 때 Start 버튼을 눌러 일시 정지를 요청한다.
2. 시스템이 스톱워치를 일시 정지한다.
3. 사용자가 다시 Start 버튼을 눌러 스톱워치 재개를 요청한다.
4. 시스템이 스톱워치를 다시 시작한다.



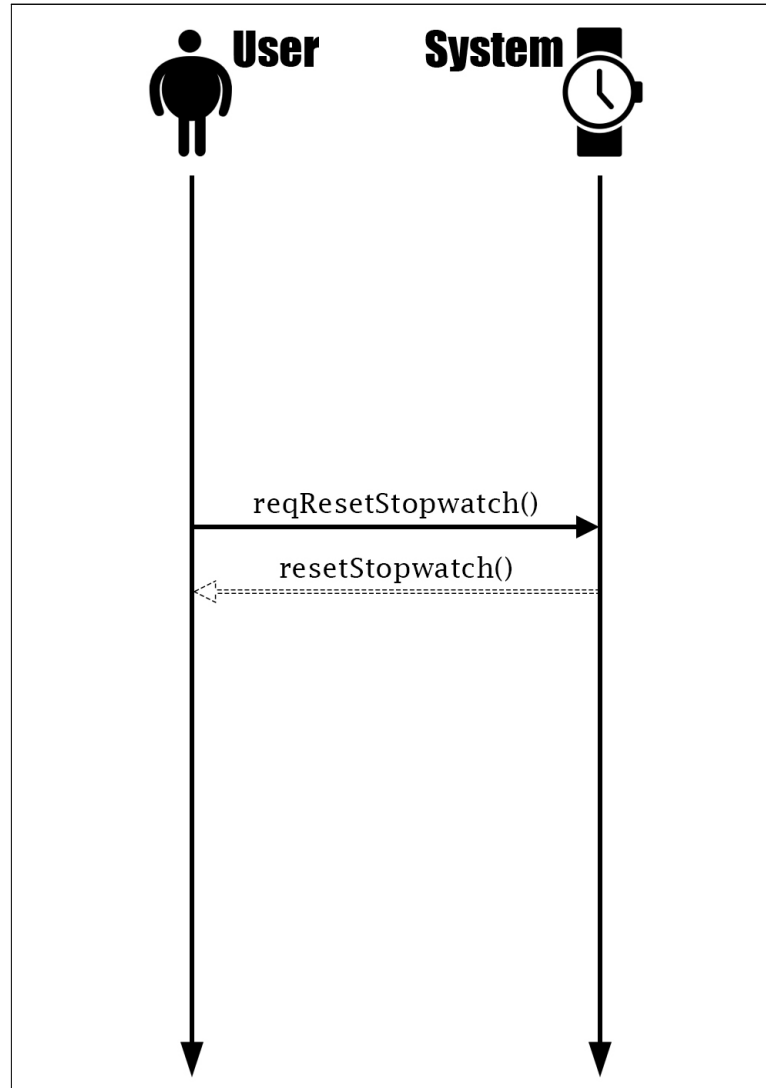
### Use Case: Check Lap & Split

1. 스톱워치가 일시 정지된 상태에서 사용자가 Funct 버튼을 2초간 눌러 Lap과 Split 데이터 조회를 요청한다.
2. 시스템이 Lap & Split 리스트 조회를 제공한다.
3. 사용자가 Start(다음), Reset(이전) 버튼을 조작하며 다음 데이터의 조회를 요청한다.
4. 시스템이 요청에 맞는 데이터를 보여 준다(3~4의 과정을 반복한다).
5. 사용자가 Funct 버튼을 눌러 다시 스톱워치로 돌아가는 것을 요청한다.
6. 시스템이 스톱워치 화면으로 전환 시킨다.



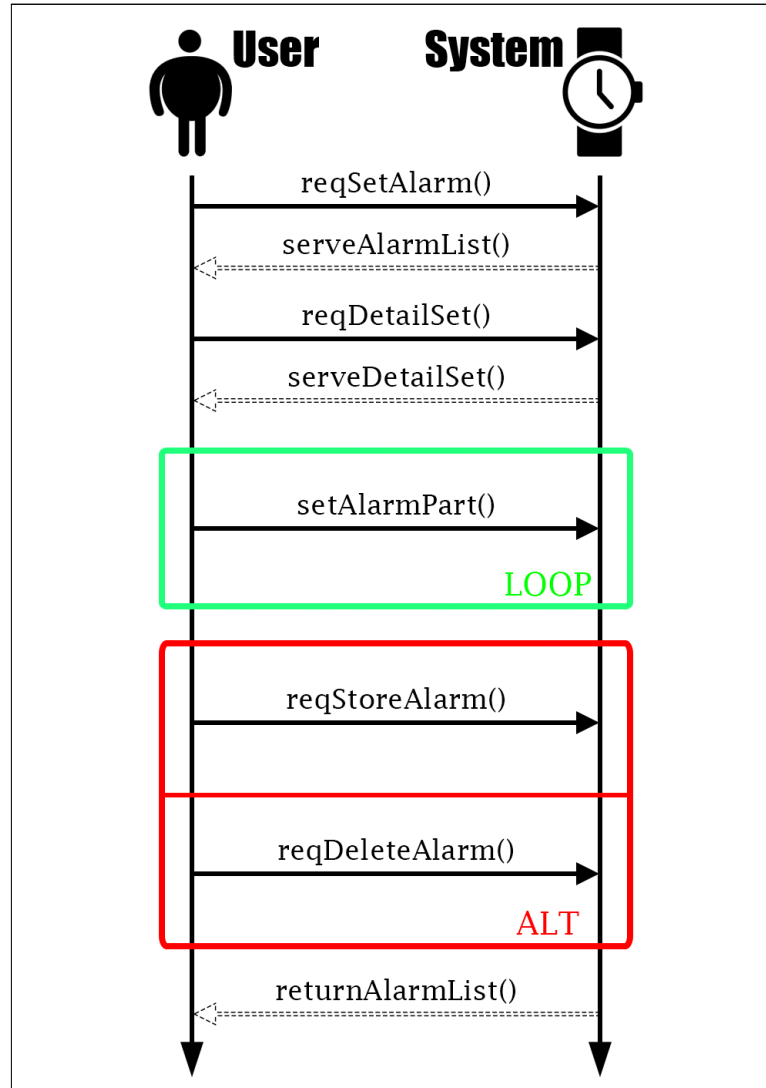
### Use Case: Reset Stopwatch

1. 스톱워치가 일시 정지된 상태에서 사용자가 Reset 버튼을 눌러 초기화를 요청한다.
2. 시스템은 저장된 Lap & Split 데이터를 삭제하고 스톱워치를 초기화한다.



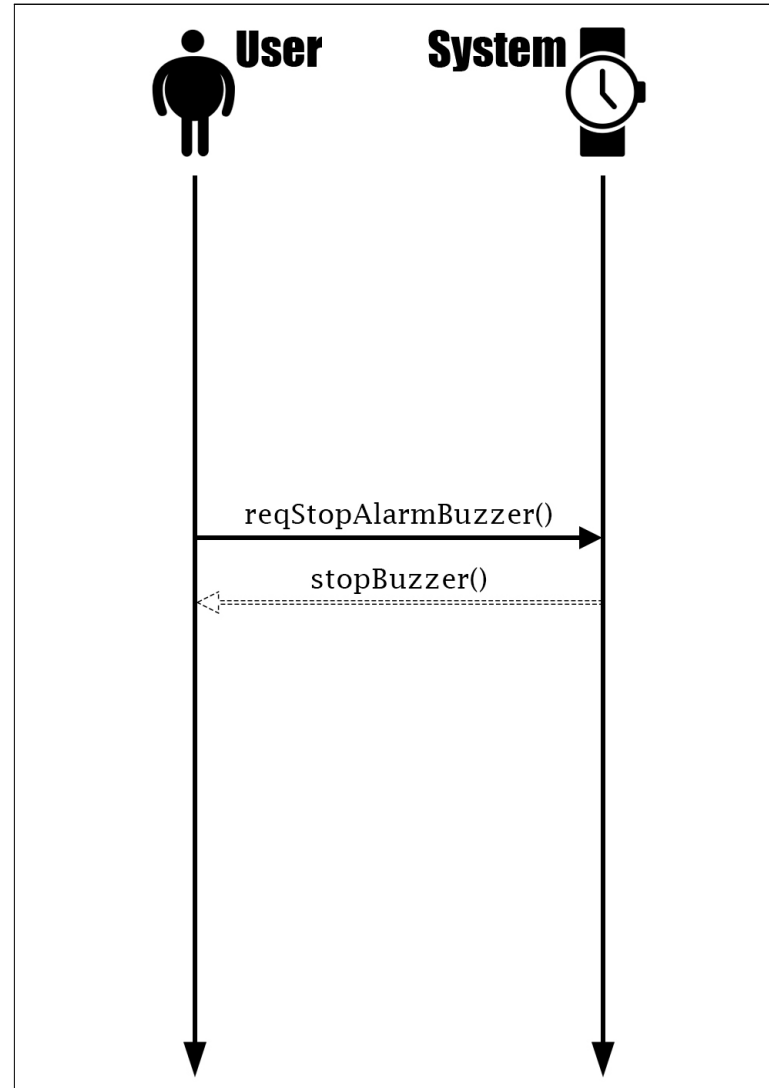
### Use Case: Set Alarm

1. 사용자가 Alarm 기능을 선택해 알람 설정을 요청한다.
2. 시스템이 알람 리스트를 제공한다.
3. 사용자가 원하는 알람을 선택한 뒤 Funct 버튼을 2초간 눌러 세부 설정을 요청한다.
4. 시스템이 알람 세부 설정을 제공한다.
5. 사용자가 알람 On.Off, 분, 시, 요일을 부분적으로 설정하는 것을 반복한다.
- 6-a. 설정 중 Funct 버튼을 2초간 눌러 설정 내용 저장을 요청한다.
- 6-b. 설정 중 Reset 버튼을 2초간 눌러 해당 알람을 리스트에서 삭제하도록 요청한다.
7. 시스템이 6에서 요청한 작업을 마친 뒤 다시 알람 리스트 화면으로 전환시킨다.



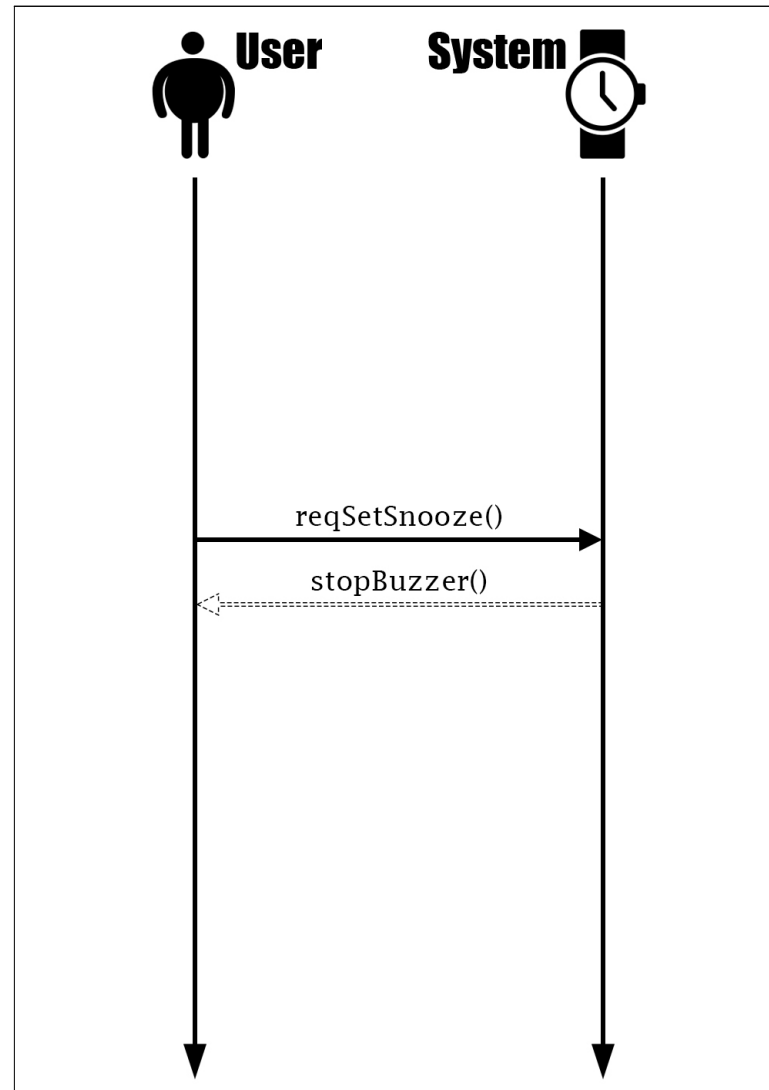
### Use Case: Stop Alarm Buzzer

1. 알람의 버저가 울릴 때 사용자가 Start 또는 Reset 버튼을 눌러 버저 멈춤을 요청한다.
2. 시스템이 버저를 즉시 멈춘다.



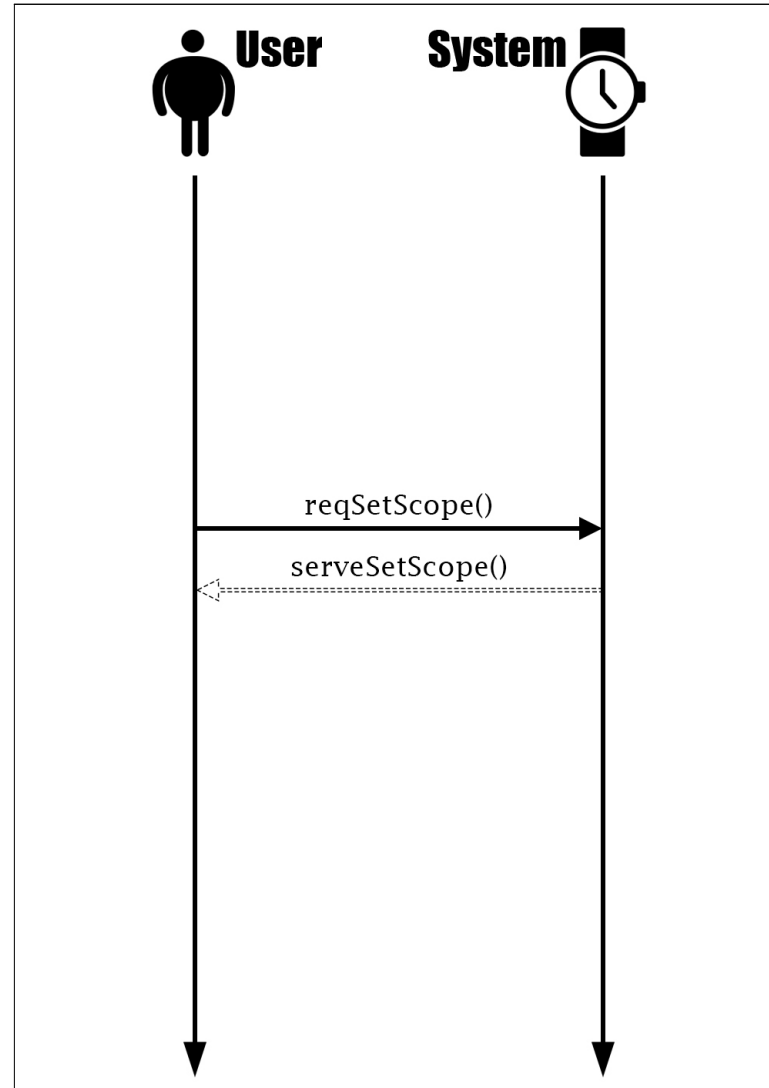
### Use Case: Set Snooze

1. 알람의 버저가 울릴 때 사용자가 Mode 또는 Funct 버튼을 눌러 버저 멈춤 및 Snooze를 요청한다.
2. 시스템이 버저를 즉시 멈추고 5분 뒤 다시 알람이 울리도록 설정한다.



### Use Case: Set Scope

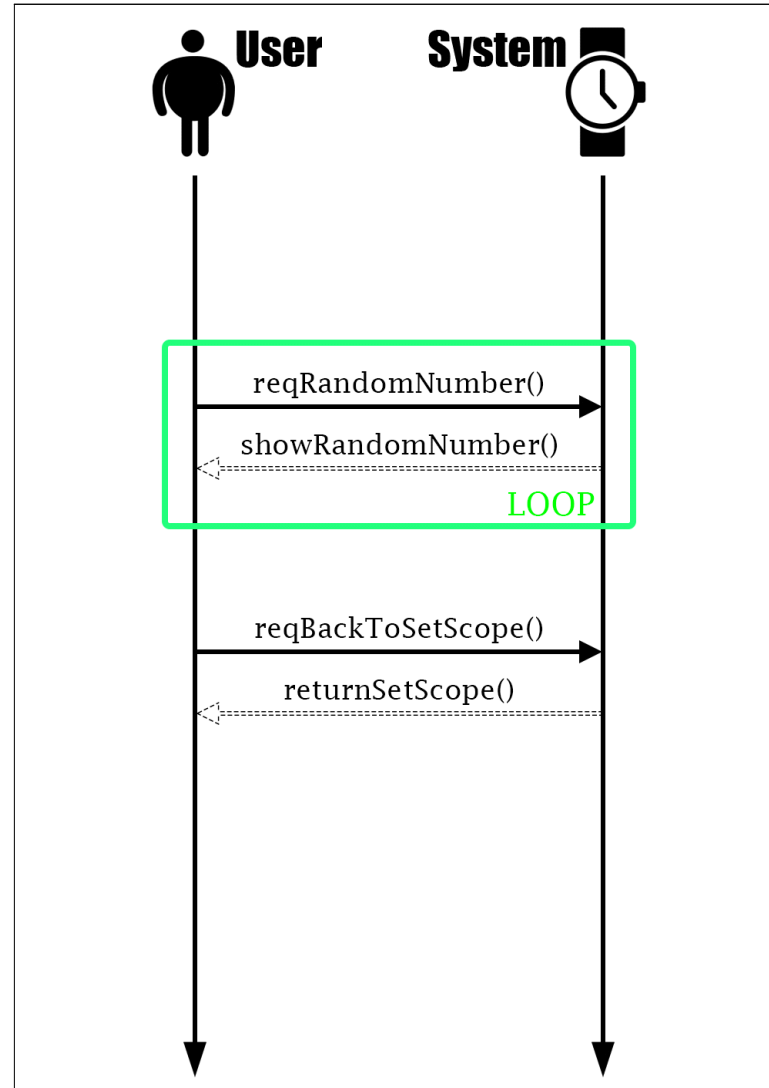
1. 사용자가 Random Number Generator 기능을 선택하여 난수 범위 설정 기능을 요청한다.
2. 시스템이 난수 범위 설정 기능을 제공한다.





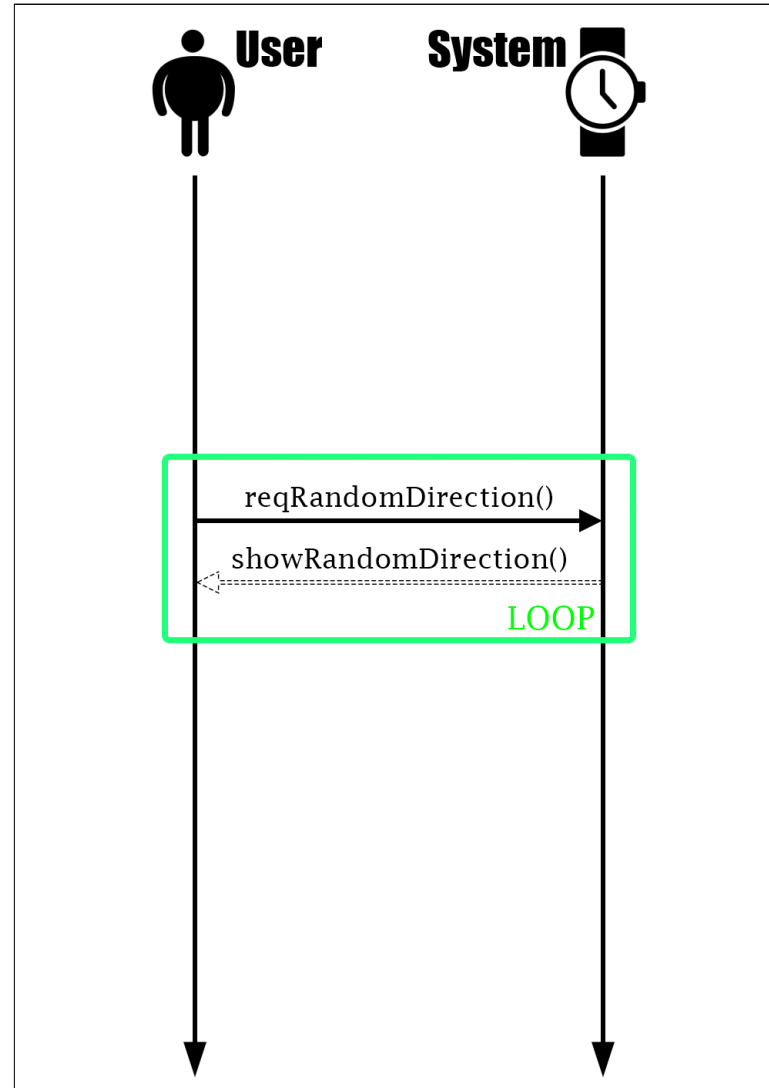
### Use Case: Generate Rand. Numb.

1. 사용자가 Set Scope 진행 중 Start 버튼을 눌러 난수 생성을 요청한다.
2. 시스템이 현재 시각을 Seed로 하는 임의의 수를 생성해 출력한다(1~2를 반복한다).
3. 결과가 나온 상태에서 사용자가 Reset 버튼을 눌러 다시 Set Scope 기능으로 돌아가는 것을 요청한다.
4. 시스템이 Set Scope 기능으로 화면을 전환시킨다.



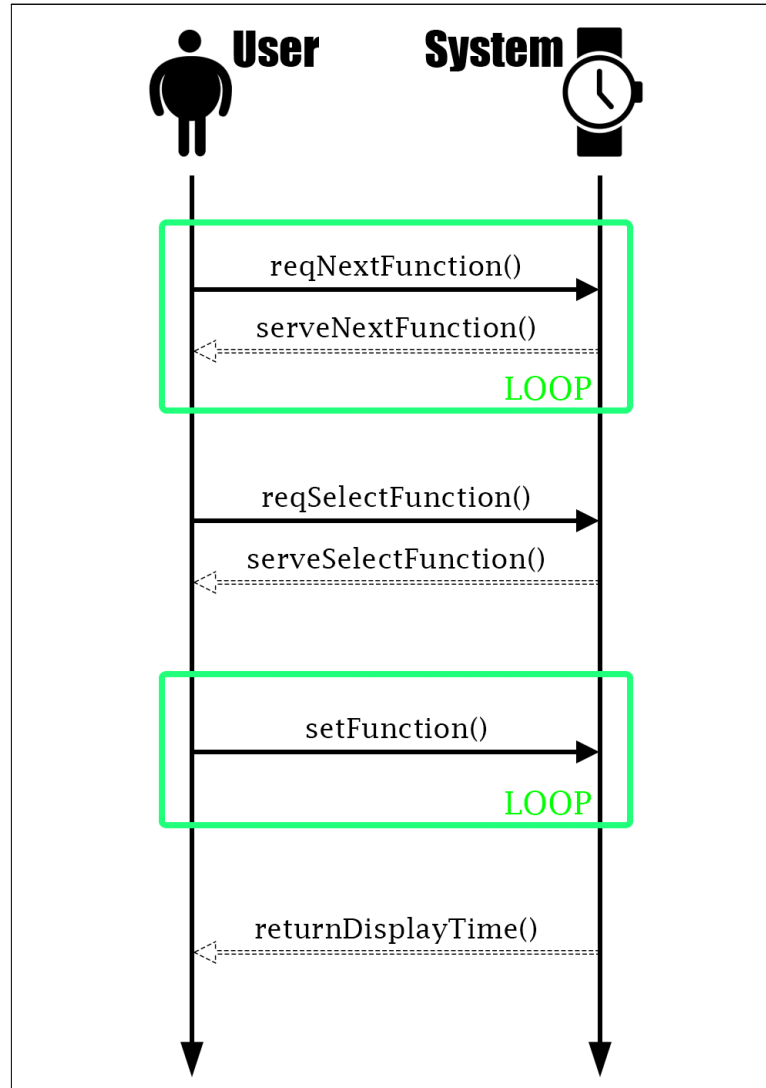
### Use Case: Generate Rand. Direct.

1. 사용자가 Random Direction Generator 기능을 선택한 상태로 Start 버튼을 눌러 임의의 방향 출력을 요청한다.
2. 시스템이 1~60 범위의 난수를 생성 후 해당 번호에 맞게 테두리 LCD에 결과를 출력한다(1~2를 반복한다).



### Use Case: Select Function

1. 기본적으로 사용자가 Mode 버튼을 눌러 다음 기능으로 넘어가도록 요청한다.
2. 시스템이 현재 On으로 설정 되어있는 다음 기능으로 화면을 전환시킨다.
3. 사용자가 Mode 버튼을 2초간 눌러 기능 선택을 요청한다.
4. 시스템이 기능 선택 기능을 제공한다.
5. 사용자가 Timer, Stopwatch, Alarm, Random Number Generator, Random Direction Generator 순서로 각 기능들의 On 여부를 체크하는 것을 반복한다.
6. 3개의 기능이 On 되면 시스템이 즉시 설정을 저장하고 고정된 0번 기능인 Time Keeping 화면으로 전환시킨다.



## 2035. Define Operation Contracts

### 1. Identify System Operations From System Sequence Diagrams

Use Case	Name of Actor-Activated Event	System Operation
1. Set Time	1: reqSetTime()	1: reqSetTime()
	2: setTimePart()	2: setTimePart()
	3: reqStoreTime()	3: reqStoreTime()
2. Display Time	Hidden	Hidden
3. Set Timer	4: reqSetTimer()	4: reqSetTimer()
	5: setTimerPart()	5: setTimerPart()
	6: reqStartTimer()	6: reqStartTimer()
	7: reqResetTimer()	7: reqResetTimer()
4. Beep Timer Buzzer	Hidden	Hidden
5. Stop Timer Buzzer	8: reqStopTimerBuzzer()	8: reqStopTimerBuzzer()
6. Start Stopwatch	9: reqStartStopwatch()	9: reqStartStopwatch()
7. Store Lap & Split	10: reqStoreLnS()	10: reqStoreLnS()
8. Pause Stopwatch	11: reqPauseStopwatch()	11: reqPauseStopwatch()
	12: reqResumeStopwatch()	12: reqResumeStopwatch()
9. Check Lap & Split	13: reqCheckLnS()	13: reqCheckLnS()
	14: reqNextLnS()	14: reqNextLnS()
	15: reqBackToStopwatch()	15: reqBackToStopwatch()
10. Reset Stopwatch	16: reqResetStopwatch()	16: reqResetStopwatch()
11. Set Alarm	17: reqSetAlarm()	17: reqSetAlarm()
	18: reqDetailSet()	18: reqDetailSet()
	19: setAlarmPart()	19: setAlarmPart()
	20: reqStoreAlarm()	20: reqStoreAlarm()
	21: reqDeleteAlarm()	21: reqDeleteAlarm()
12. Beep Alarm Buzzer	Hidden	Hidden
13. Stop Alarm Buzzer	22: reqStopAlarmBuzzer()	22: reqStopAlarmBuzzer()
14. Set Snooze	23: reqSetSnooze()	23: reqSetSnooze()
15. Set Scope	24: reqSetScope()	24: reqSetScope()
16. Generate Random Number	25: reqRandomNumber()	25: reqRandomNumber()
	26: reqBackToSetScope()	26: reqBackToSetScope()
17. Generate Random Direction	27: reqRandomDirection()	27: reqRandomDirection()
18. Select Function	28: reqNextFuction()	28: reqNextFuction()
	29: reqSelectFuction()	29: reqSelectFuction()
	30: setFunction()	30: setFunction()

## 2. Operation Contracts

<b>Name</b>	1: reqSetTime()
<b>Responsibilities</b>	사용자가 시간 설정 기능을 요청한다.
<b>Type</b>	Syetem
<b>Cross References</b>	R1.1
<b>Notes</b>	
<b>Exceptions</b>	N/A
<b>Output</b>	N/A
<b>Pre-Conditions</b>	Time Keeping 기능이 선택되어 있어야 한다.
<b>Post-Conditions</b>	시스템이 시간 설정 기능을 제공한다.

<b>Name</b>	2: setTimePart()
<b>Responsibilities</b>	사용자가 12/24h 포맷, 분, 시, 요일 등의 부분 설정을 한다.
<b>Type</b>	Syetem
<b>Cross References</b>	R1.1
<b>Notes</b>	
<b>Exceptions</b>	N/A
<b>Output</b>	N/A
<b>Pre-Conditions</b>	시간 설정 기능이 작동중이어야 한다.
<b>Post-Conditions</b>	시스템이 각 설정값을 저장한다.

<b>Name</b>	3: reqStoreTime()
<b>Responsibilities</b>	사용자가 설정의 최종 저장을 요청한다.
<b>Type</b>	Syetem
<b>Cross References</b>	R1.1
<b>Notes</b>	
<b>Exceptions</b>	N/A
<b>Output</b>	N/A
<b>Pre-Conditions</b>	시간 설정 기능이 작동중이어야 한다.
<b>Post-Conditions</b>	시스템이 설정된 값을 적용시킨 뒤 시간 표시 화면을 띄운다.

<b>Name</b>	4: reqSetTimer()
<b>Responsibilities</b>	사용자가 타이머 설정 기능을 요청한다.
<b>Type</b>	Syetem
<b>Cross References</b>	R2.1
<b>Notes</b>	
<b>Exceptions</b>	N/A
<b>Output</b>	N/A
<b>Pre-Conditions</b>	Timer 기능이 선택되어 있어야 한다.
<b>Post-Conditions</b>	시스템이 타이머 설정 기능을 제공한다.

<b>Name</b>	5: setTimerPart()
<b>Responsibilities</b>	사용자가 타이머의 초, 분, 시에 대한 부분 설정을 한다.
<b>Type</b>	Syetem
<b>Cross References</b>	R2.1
<b>Notes</b>	Loop
<b>Exceptions</b>	N/A
<b>Output</b>	N/A
<b>Pre-Conditions</b>	타이머 설정 기능이 작동중이어야 한다.
<b>Post-Conditions</b>	시스템이 각 설정값을 저장한다.

<b>Name</b>	6: reqStartTimer()
<b>Responsibilities</b>	사용자가 설정 중 타이머 시작을 요청한다.
<b>Type</b>	Syetem
<b>Cross References</b>	R2.1
<b>Notes</b>	Alt
<b>Exceptions</b>	N/A
<b>Output</b>	N/A
<b>Pre-Conditions</b>	타이머 설정 기능이 작동중이어야 한다.
<b>Post-Conditions</b>	시스템이 설정된 값을 적용시킨 뒤 시간 표시 화면을 띄운다.

<b>Name</b>	7: reqResetTimer()
<b>Responsibilities</b>	사용자가 설정 중 타이머 설정 초기화를 요청한다.
<b>Type</b>	Syetem
<b>Cross References</b>	R2.1
<b>Notes</b>	Alt
<b>Exceptions</b>	N/A
<b>Output</b>	N/A
<b>Pre-Conditions</b>	타이머 설정 기능이 작동중이어야 한다.
<b>Post-Conditions</b>	시스템이 타이머를 초기화한다.

<b>Name</b>	8: reqStopTimerBuzzer()
<b>Responsibilities</b>	타이머의 버저가 울릴 때 사용자가 아무 버튼을 눌러 버저 멈춤을 요청한다.
<b>Type</b>	Syetem
<b>Cross References</b>	R2.3
<b>Notes</b>	
<b>Exceptions</b>	N/A
<b>Output</b>	N/A
<b>Pre-Conditions</b>	타이머의 버저가 작동중이어야 한다.
<b>Post-Conditions</b>	시스템이 버저를 즉시 멈춘다.

<b>Name</b>	9: reqStartStopwatch()
<b>Responsibilities</b>	사용자가 Stopwatch 기능을 선택한 상태로 스톱워치의 시작을 요청한다.
<b>Type</b>	Syetem
<b>Cross References</b>	R3.1
<b>Notes</b>	
<b>Exceptions</b>	N/A
<b>Output</b>	N/A
<b>Pre-Conditions</b>	Stopwatch 기능이 선택되어 있어야 한다.
<b>Post-Conditions</b>	시스템이 스톱워치를 작동시킨다.

<b>Name</b>	10: reqStoreLnS()
<b>Responsibilities</b>	사용자가 스톱워치가 진행 중일 때 Lap & Split 저장을 요청한다.
<b>Type</b>	Syetem
<b>Cross References</b>	R3.2
<b>Notes</b>	
<b>Exceptions</b>	N/A
<b>Output</b>	N/A
<b>Pre-Conditions</b>	스톱워치가 진행중이어야 한다.
<b>Post-Conditions</b>	시스템이 Lap & Split 데이터를 저장한다.

<b>Name</b>	11: reqPauseStopwatch()
<b>Responsibilities</b>	사용자가 스톱워치가 진행 중일 때 일시 정지를 요청한다.
<b>Type</b>	Syetem
<b>Cross References</b>	R3.3
<b>Notes</b>	
<b>Exceptions</b>	N/A
<b>Output</b>	N/A
<b>Pre-Conditions</b>	스톱워치가 진행중이어야 한다.
<b>Post-Conditions</b>	시스템이 스톱워치를 일시 정지 시킨다.

<b>Name</b>	12: reqResumeStopwatch()
<b>Responsibilities</b>	사용자가 스톱워치가 일시 정지 중일 때 재개를 요청한다.
<b>Type</b>	Syetem
<b>Cross References</b>	R3.3
<b>Notes</b>	
<b>Exceptions</b>	N/A
<b>Output</b>	N/A
<b>Pre-Conditions</b>	스톱워치가 일시정지 상태여야 한다.
<b>Post-Conditions</b>	시스템이 스톱워치를 재개시킨다.



<b>Name</b>	13: reqCheckLnS()
<b>Responsibilities</b>	사용자가 Lap & Split 데이터 조회를 요청한다.
<b>Type</b>	Syetem
<b>Cross References</b>	R3.4
<b>Notes</b>	
<b>Exceptions</b>	N/A
<b>Output</b>	N/A
<b>Pre-Conditions</b>	스톱워치가 일시정지 상태여야 한다. Lap & Split 데이터가 하나 이상 존재해야 한다.
<b>Post-Conditions</b>	시스템이 Lap & Split 데이터 조회를 제공한다.

<b>Name</b>	14: reqNextLnS()
<b>Responsibilities</b>	사용자가 다음(또는 이전) 데이터 조회를 요청한다.
<b>Type</b>	Syetem
<b>Cross References</b>	R3.4
<b>Notes</b>	Loop
<b>Exceptions</b>	N/A
<b>Output</b>	N/A
<b>Pre-Conditions</b>	Lap & Split 데이터를 조회중이어야 한다.
<b>Post-Conditions</b>	시스템이 요청에 맞는 데이터를 보여준다.

<b>Name</b>	15: reqBackToStopwatch()
<b>Responsibilities</b>	사용자가 조회를 마치고 다시 스톱워치로 돌아가는 것을 요청한다.
<b>Type</b>	Syetem
<b>Cross References</b>	R3.4
<b>Notes</b>	
<b>Exceptions</b>	N/A
<b>Output</b>	N/A
<b>Pre-Conditions</b>	Lap & Split 데이터를 조회중이어야 한다.
<b>Post-Conditions</b>	시스템이 스톱워치로 화면을 전환시킨다.

<b>Name</b>	16: reqResetStopwatch()
<b>Responsibilities</b>	사용자가 스톱워치의 초기화를 요청한다.
<b>Type</b>	Syetem
<b>Cross References</b>	R3.5
<b>Notes</b>	
<b>Exceptions</b>	N/A
<b>Output</b>	N/A
<b>Pre-Conditions</b>	스톱워치가 일시 정지된 상태여야 한다.
<b>Post-Conditions</b>	시스템이 Lap & Split 데이터를 삭제하고 스톱워치를 초기화한다.

<b>Name</b>	17: reqSetAlarm()
<b>Responsibilities</b>	사용자가 알람 설정을 요청한다.
<b>Type</b>	Syetem
<b>Cross References</b>	R4.1
<b>Notes</b>	저장되어있는 알람이 없으면 세부 설정으로 바로 넘어간다.
<b>Exceptions</b>	N/A
<b>Output</b>	N/A
<b>Pre-Conditions</b>	Alarm 기능이 선택되어 있어야 한다. 저장되어있는 알람이 하나 이상 존재해야 한다.
<b>Post-Conditions</b>	시스템이 알람 리스트를 제공한다.

<b>Name</b>	18: reqDetailSet()
<b>Responsibilities</b>	사용자가 알람의 세부 설정을 요청한다.
<b>Type</b>	Syetem
<b>Cross References</b>	R4.1
<b>Notes</b>	
<b>Exceptions</b>	N/A
<b>Output</b>	N/A
<b>Pre-Conditions</b>	알람 리스트 조회 또는 알람이 0개인 상태여야 한다.
<b>Post-Conditions</b>	시스템이 알람 세부 설정을 제공한다.

<b>Name</b>	19: setAlarmPart()
<b>Responsibilities</b>	사용자가 알람의 On/Off, 분, 시, 요일 등을 부분적으로 설정한다.
<b>Type</b>	Syetem
<b>Cross References</b>	R4.1
<b>Notes</b>	Loop
<b>Exceptions</b>	N/A
<b>Output</b>	N/A
<b>Pre-Conditions</b>	알람 세부 설정이 작동중이어야 한다.
<b>Post-Conditions</b>	시스템이 세부 설정 내용을 저장한다.

<b>Name</b>	20: reqStoreAlarm()
<b>Responsibilities</b>	사용자가 알람 설정 저장 및 적용을 요청한다.
<b>Type</b>	Syetem
<b>Cross References</b>	R4.1
<b>Notes</b>	Alt
<b>Exceptions</b>	N/A
<b>Output</b>	N/A
<b>Pre-Conditions</b>	알람 세부 설정이 작동중이어야 한다.
<b>Post-Conditions</b>	시스템이 알람의 전체 설정을 저장하고 적용시킨다. 이후 알람 리스트 화면으로 전환시킨다.

<b>Name</b>	21: reqDeleteAlarm()
<b>Responsibilities</b>	사용자가 알람 삭제를 요청한다.
<b>Type</b>	Syetem
<b>Cross References</b>	R4.1
<b>Notes</b>	Alt
<b>Exceptions</b>	N/A
<b>Output</b>	N/A
<b>Pre-Conditions</b>	알람 세부 설정이 작동중이어야 한다.
<b>Post-Conditions</b>	시스템이 현재 설정중인 알람을 리스트에서 삭제한다. 이후 알람 리스트 화면으로 전환시킨다.

<b>Name</b>	22: reqStopAlarmBuzzer()
<b>Responsibilities</b>	알람의 버저가 울릴 때 사용자가 Start 또는 Reset 버튼을 눌러 버저 멈춤을 요청한다.
<b>Type</b>	Syetem
<b>Cross References</b>	R4.3
<b>Notes</b>	
<b>Exceptions</b>	N/A
<b>Output</b>	N/A
<b>Pre-Conditions</b>	알람의 버저가 작동중이어야 한다.
<b>Post-Conditions</b>	시스템이 버저를 즉시 멈춘다.

<b>Name</b>	23: reqSetSnooze()
<b>Responsibilities</b>	알람의 버저가 울릴 때 사용자가 Mode 또는 Funct 버튼을 눌러 버저 멈춤 및 5분 Snooze를 요청한다.
<b>Type</b>	Syetem
<b>Cross References</b>	R4.4
<b>Notes</b>	
<b>Exceptions</b>	N/A
<b>Output</b>	N/A
<b>Pre-Conditions</b>	알람의 버저가 작동중이어야 한다.
<b>Post-Conditions</b>	시스템이 버저를 즉시 멈춘다. 그리고 5분 뒤 임시 알람이 울리도록 설정한다.

<b>Name</b>	24: reqSetScope()
<b>Responsibilities</b>	사용자가 Random Number Generator 기능을 선택하여 난수 범위 설정을 요청한다.
<b>Type</b>	Syetem
<b>Cross References</b>	R5.1
<b>Notes</b>	
<b>Exceptions</b>	N/A
<b>Output</b>	N/A
<b>Pre-Conditions</b>	Random Number Generator 기능을 선택해야 한다.
<b>Post-Conditions</b>	시스템이 난수 범위 설정 기능을 제공한다.

<b>Name</b>	25: reqRandomNumber()
<b>Responsibilities</b>	사용자가 설정된 범위 안의 난수 생성을 요청한다.
<b>Type</b>	Syetem
<b>Cross References</b>	R5.2
<b>Notes</b>	Loop
<b>Exceptions</b>	N/A
<b>Output</b>	사용자가 설정한 범위 안의 난수
<b>Pre-Conditions</b>	난수 범위가 설정되어 있어야 한다.
<b>Post-Conditions</b>	시스템이 현재 시간을 Seed로 해 범위 내 난수를 생성해 출력한다.

<b>Name</b>	26: reqBackToSetScope()
<b>Responsibilities</b>	사용자가 난수 범위 설정으로 돌아가는 것을 요청한다.
<b>Type</b>	Syetem
<b>Cross References</b>	R5.2
<b>Notes</b>	
<b>Exceptions</b>	N/A
<b>Output</b>	N/A
<b>Pre-Conditions</b>	난수가 생성 및 출력 완료되어야 한다.
<b>Post-Conditions</b>	시스템이 난수 범위 설정 기능으로 화면을 전환한다.

<b>Name</b>	27: reqRandomDirection()
<b>Responsibilities</b>	사용자가 Random Direction Generator 기능을 선택한 상태로 임의의 방향 출력을 요청한다.
<b>Type</b>	Syetem
<b>Cross References</b>	R6.1
<b>Notes</b>	Loop
<b>Exceptions</b>	N/A
<b>Output</b>	1~60 범위 중의 난수에 해당하는 테두리 LCD의 표시
<b>Pre-Conditions</b>	Random Direction Generator 기능이 작동 중이어야 한다.
<b>Post-Conditions</b>	시스템이 현재 시간을 Seed로 해 1~60 범위 내 난수를 생성하고 그에 맞는 테두리 LCD 세그먼트를 점등시킨다.

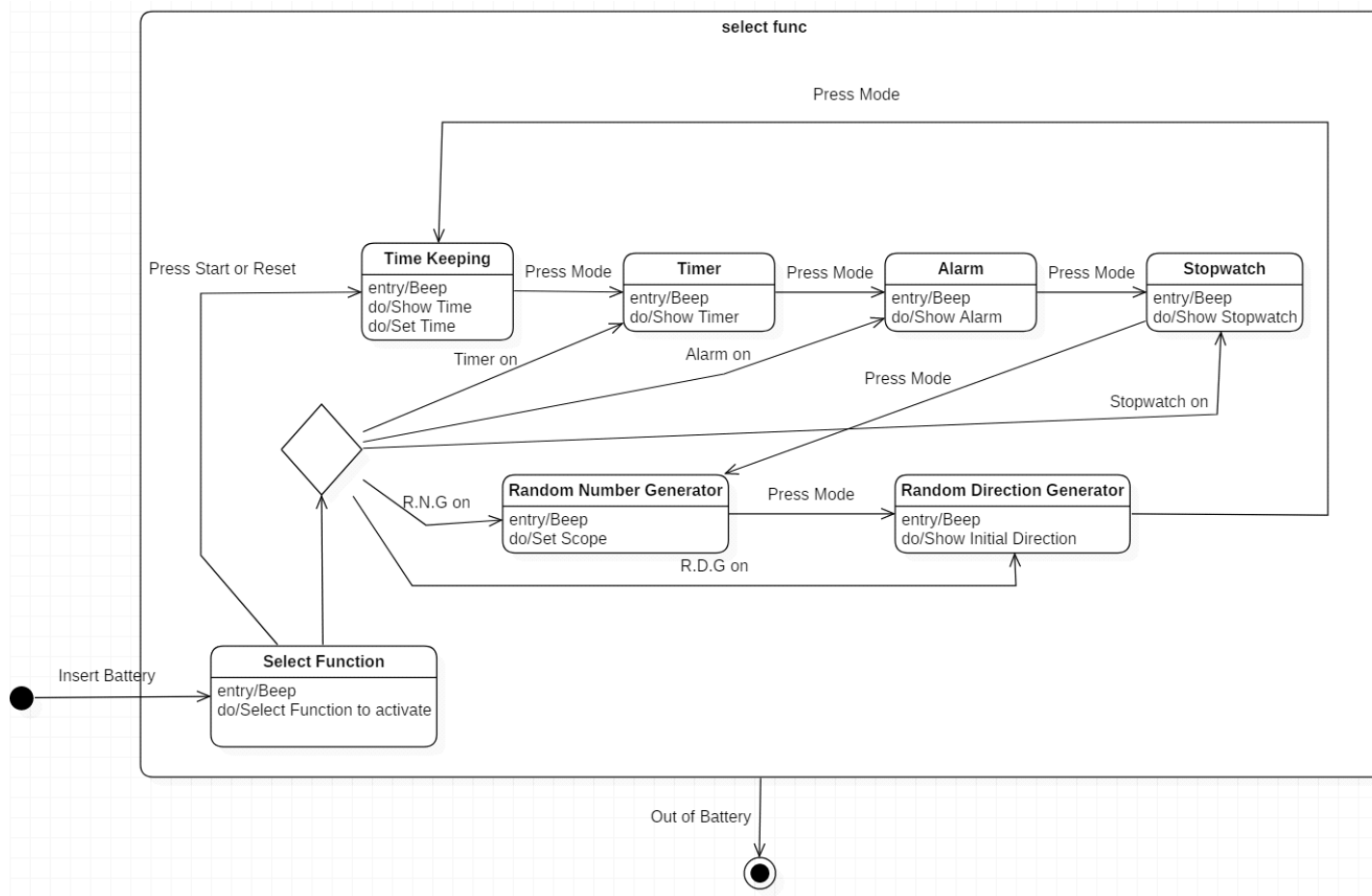
<b>Name</b>	28: reqNextFuction()
<b>Responsibilities</b>	사용자가 다음 기능으로 넘어가는 것을 요청한다.
<b>Type</b>	Syetem
<b>Cross References</b>	R7.1
<b>Notes</b>	Loop
<b>Exceptions</b>	N/A
<b>Output</b>	N/A
<b>Pre-Conditions</b>	N/A
<b>Post-Conditions</b>	시스템이 On으로 설정되어 있는 다음 기능으로 화면을 전환시킨다.

<b>Name</b>	29: reqSelectFuction()
<b>Responsibilities</b>	사용자가 기능 선택 기능을 요청한다.
<b>Type</b>	Syetem
<b>Cross References</b>	R7.1
<b>Notes</b>	
<b>Exceptions</b>	N/A
<b>Output</b>	N/A
<b>Pre-Conditions</b>	N/A
<b>Post-Conditions</b>	시스템이 기능 선택 기능을 제공한다.

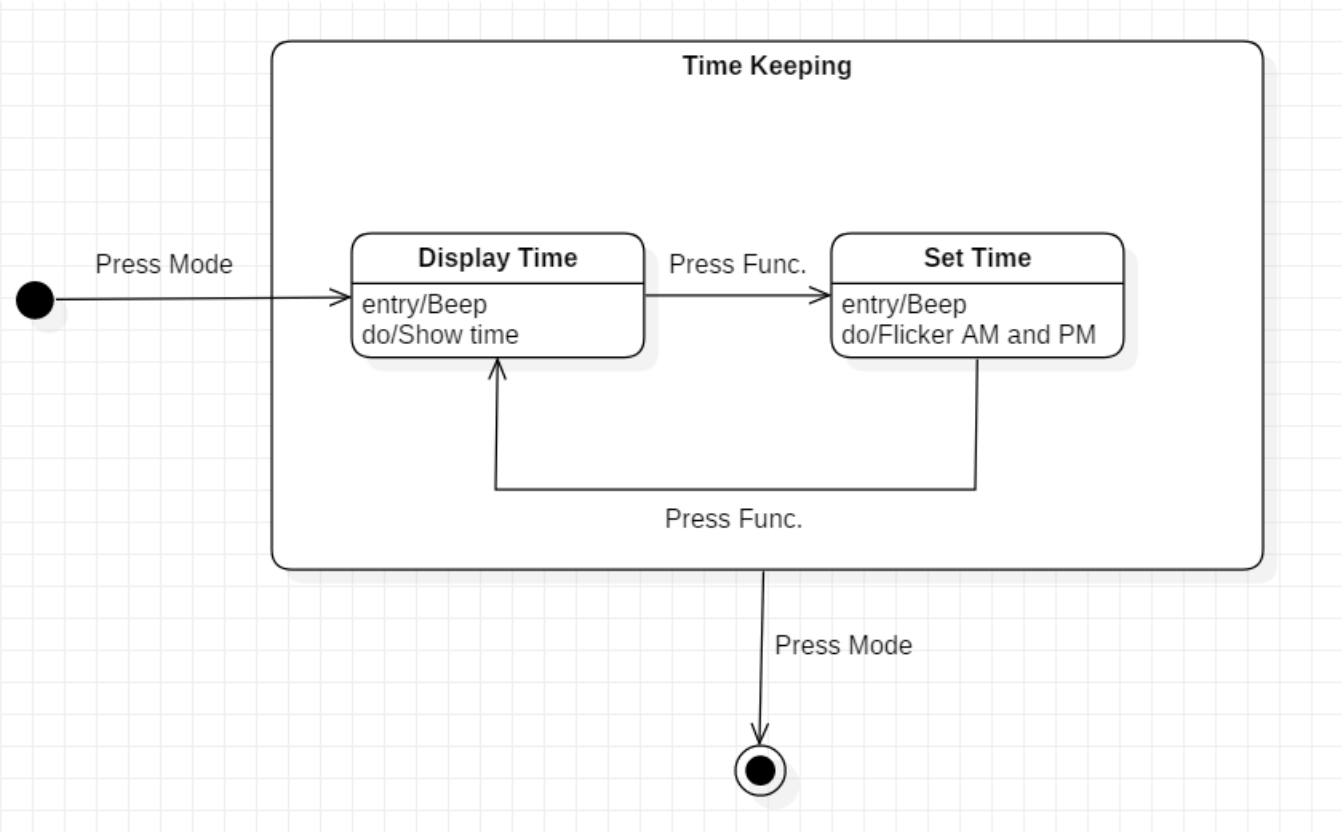
<b>Name</b>	30: setFunction()
<b>Responsibilities</b>	사용자가 Timer, Stopwatch, Alarm, Random Number Generator, Random Direction Generator 순서로 각 기능들의 On 여부를 체크(설정)한다.
<b>Type</b>	Syetem
<b>Cross References</b>	R7.1
<b>Notes</b>	Loop
<b>Exceptions</b>	N/A
<b>Output</b>	N/A
<b>Pre-Conditions</b>	기능 선택 기능이 작동중이어야 한다.
<b>Post-Conditions</b>	설정 중 3개 이상의 기능이 On되면 시스템이 즉시 설정을 저장하고 Time Keeping 화면으로 전환한다.

# 2036. Define State Diagrams

## 1. Overall

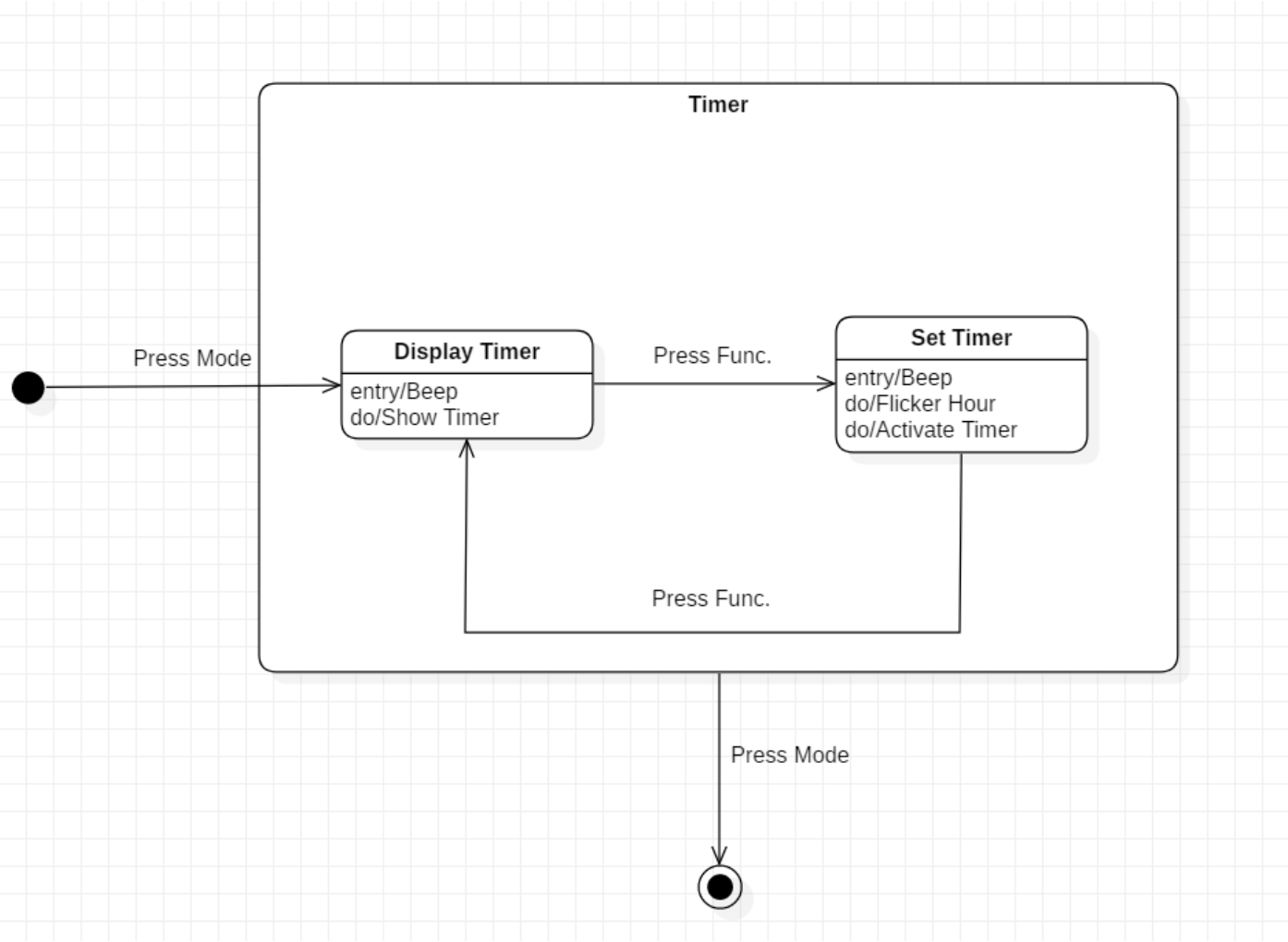


## 2. Time Keeping

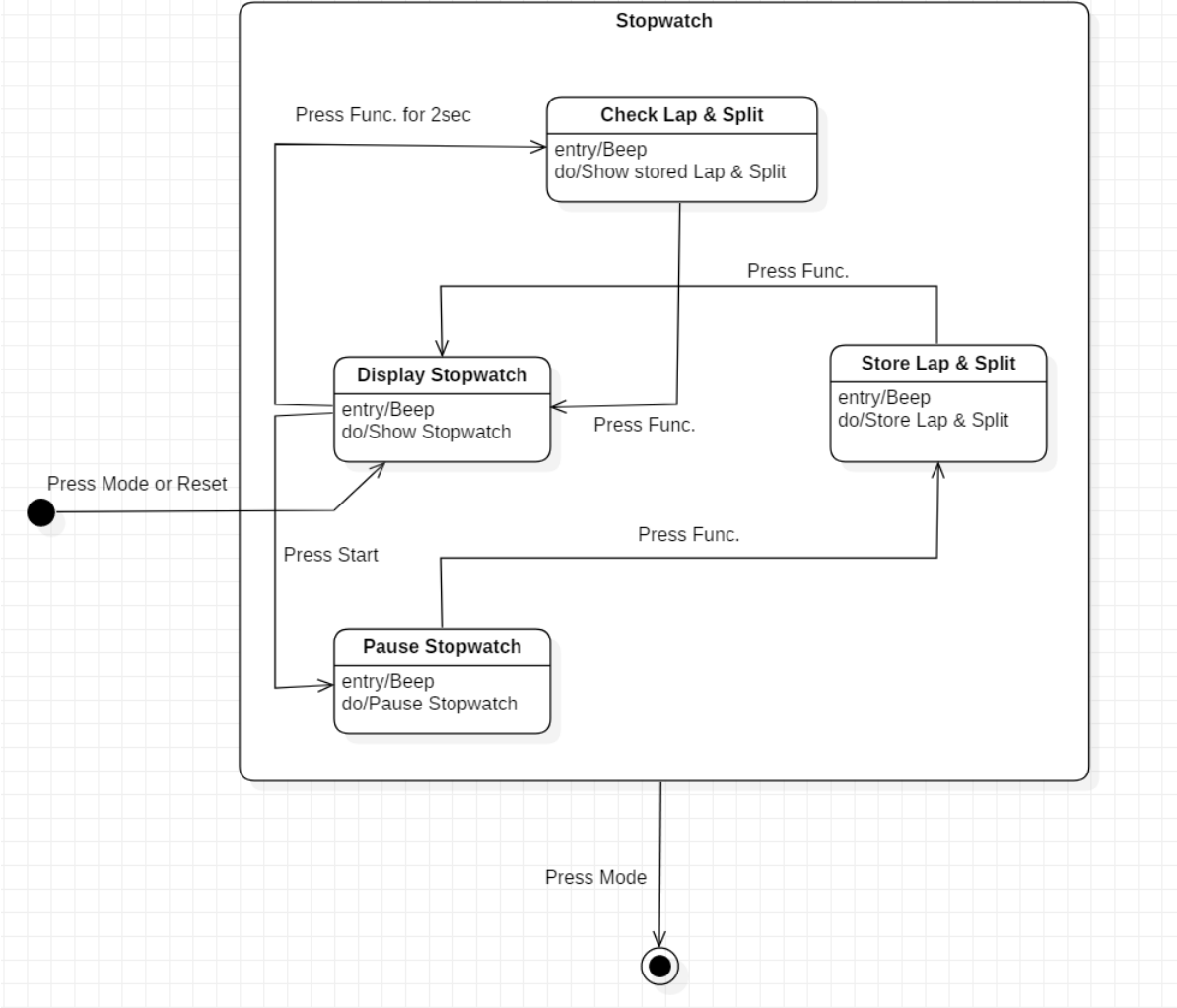




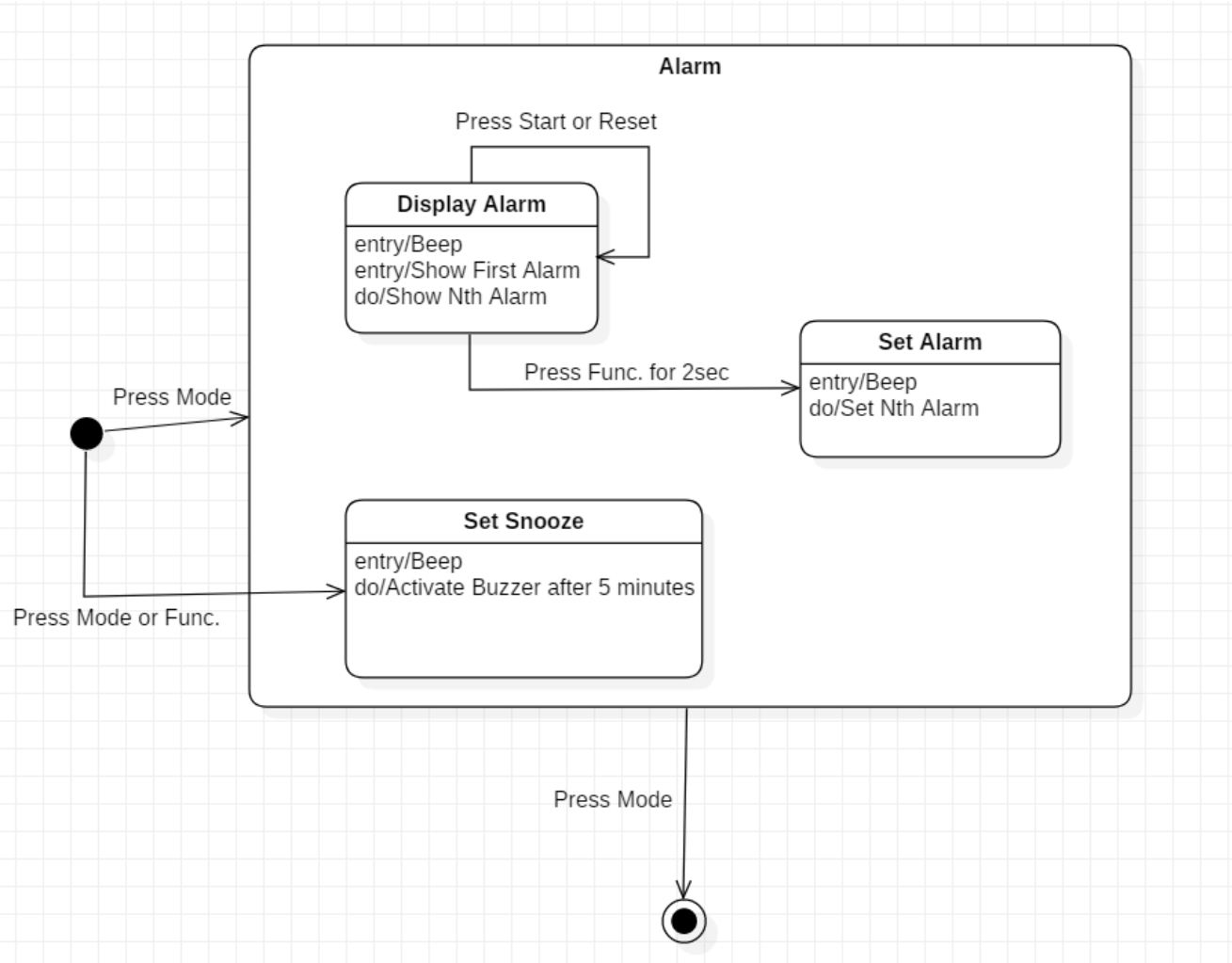
### 3. Timer



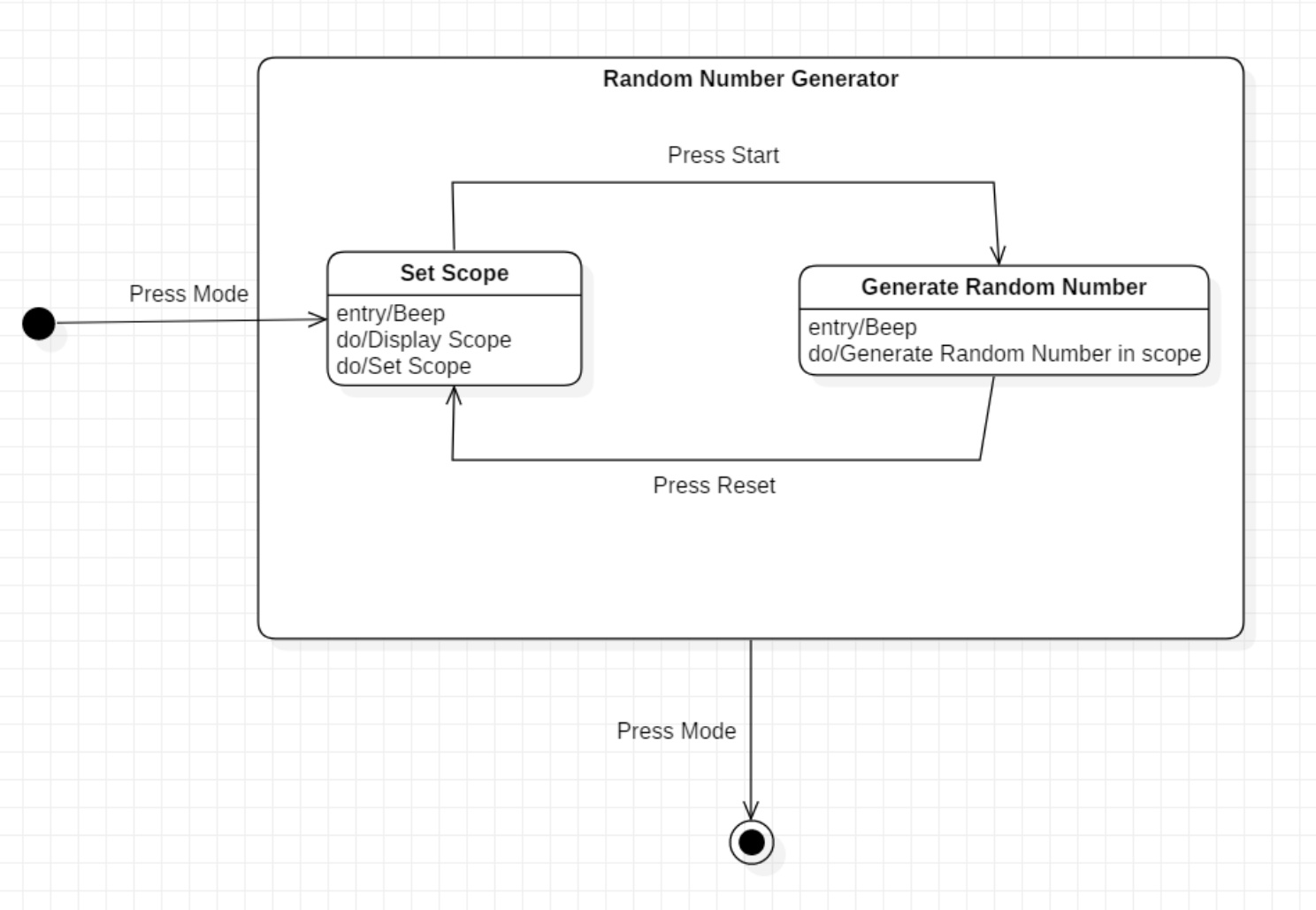
# 4. Stopwatch



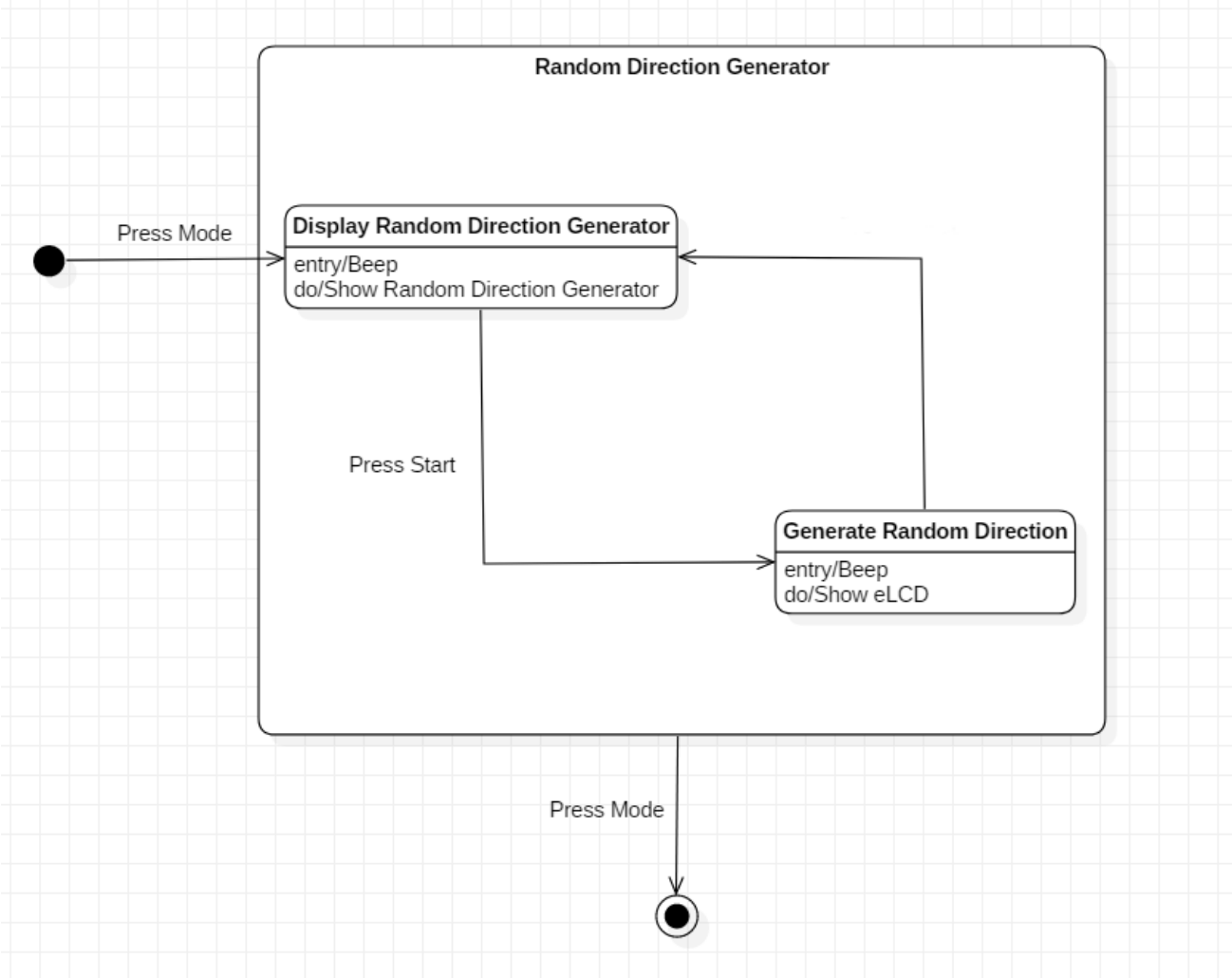
# 5. Alarm



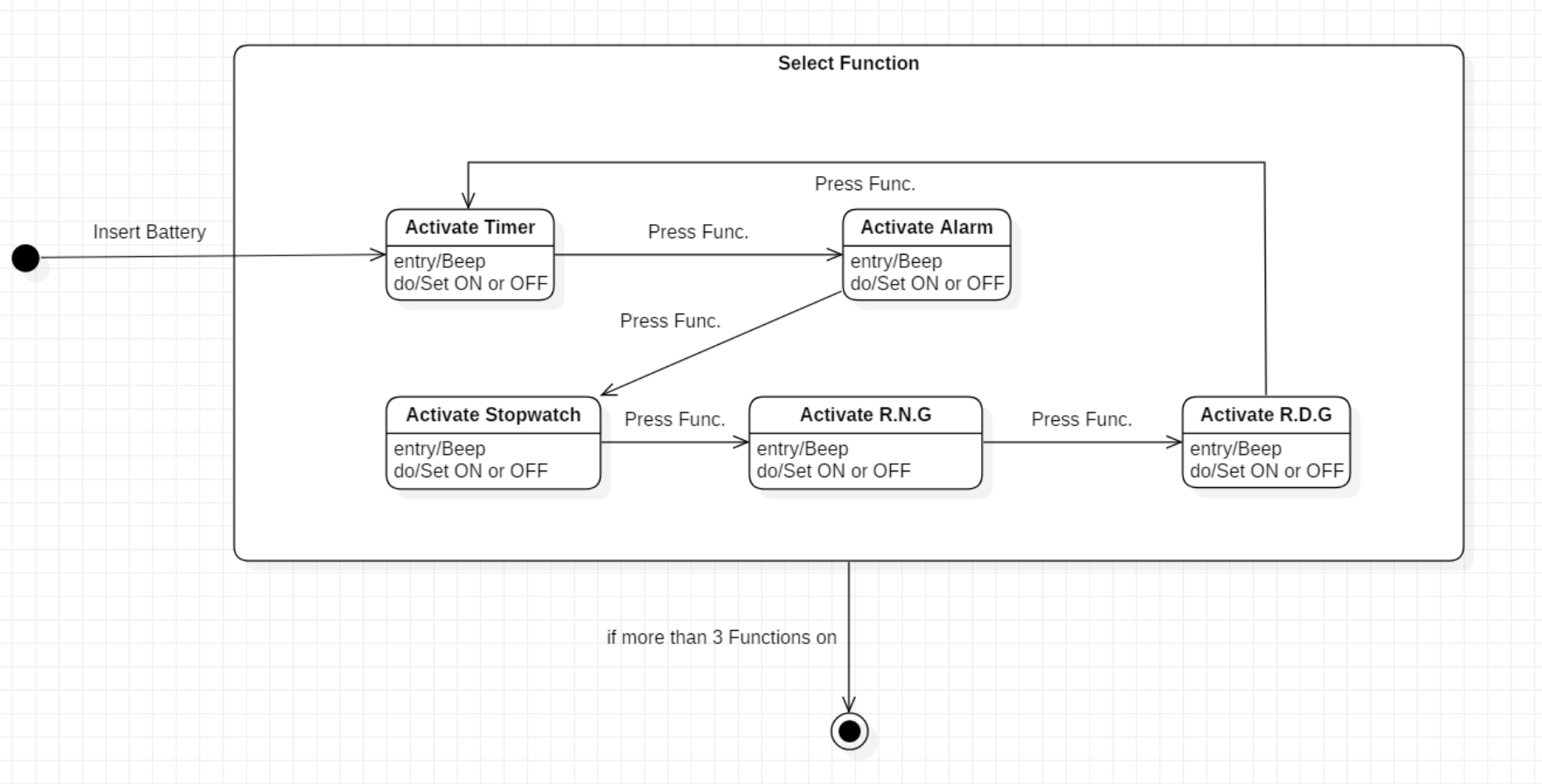
# 6. Random Number Generator



# 7. Random Direction Generator



# 8. Select Function



## 2037. Refine System Test Case

Test #	Test 항목	Description	Use Case	Sys. Func.
1	시간 설정 시험	- User에게 시간을 입력 받는 기능 Test - 입력 후 저장되었는지 Test	1. Set Time	R1.1
2	시간 출력 시험	- 저장되어 있는 시간이 LCD에 출력되는지 Test - Southfinder가 정확한 방향을 가리키고 있는지 Test	2. Display Time	R1.2
3	타이머 설정 시험	- User에게 Timer 설정을 입력 받는 기능 Test - 입력 후 저장되었는지 Test	3. Set Timer	R2.1
4	타이머 버저 시험	- Timer가 0이되면 5초동안 버저를 울리는지 Test	4. Beep Timer Buzzer	R2.2
5	타이머 버저 중지 시험	- Timer의 버저가 울릴 때 User가 버튼을 누르면 버저가 정지하는지 Test - Timer가 초기화 되는지 Test	5. Stop Timer Buzzer	R2.3
6	스톱워치 시험	- User에게 Stopwatch 설정을 입력 받는 기능 Test - 입력 후 성공적으로 저장되었는지 Test	6. Start Stopwatch	R3.1
7	랩, 스플릿 시험	- User가 버튼을 누를 때 마다 Lap, Split이 저장되는지 Test - 최대 30개까지 저장이 되는지 Test - 30개가 넘어가면 저장이 안되는지 Test	7. Store Lap & Split	R3.2
8	스톱워치 중지 시험	- User에게 스톱워치 일시정지를 입력받으면 정지하는지 Test - User에게 스톱워치 재개를 요청받으면 재개하는지 Test	8. Pause Stopwatch	R3.3
9	랩, 스플릿 확인 시험	- User에게 버튼을 입력받아 저장된 Lap, Split이 출력되는지 Test - 제일 마지막 Lap, Split이 출력된 후 다시 맨 처음 Lap, Split이 출력되는지 Test - Lap, Split 데이터가 없으면 기능이 제대로 블락되는지 Test	9. Check Lap & Split	R3.4
10	스톱워치 초기화 시험	- User에게 버튼을 입력받아 실행중인 스톱워치가 초기화 되는지 Test - 저장되어있던 Lap, Split이 삭제되는지 Test	10. Reset Stopwatch	R3.5
11	알람 설정 시험	- User에게 버튼을 입력받아 알람이 저장되는지 Test - 최대 10까지 저장이 되며 10개가 넘어가면 저장이 안되는지 Test - 설정한 알람이 삭제 되는지 Test - 알람 기능을 껐다 켤 시 알람 정보가 삭제되지 않는지 Test	11. Set Alarm	R4.1

12	알람 버저 시험	- 알람에 저장된 시간이 되면 버저가 1분 동안 울리는지 Test	12. Beep Alarm Buzzer	R4.2
13	알람 버저 중지 시험	- 알람 버저가 울릴 때 User가 중지 요청 시 버저가 중지되는지 Test	13. Stop Alarm Buzzer	R4.3
14	스누즈 시험	- 알람 버저가 울릴 때 User가 Snooze 요청 시 5분 후 다시 버저가 울리는지 Test	14. Set Snooze	R4.4
15	범위 조절 시험	- User의 버튼 입력에 따라 1~999,999가 입력 되는지 Test	15. Set Scope	R5.1
16	난수 생성 시험	- 현재 시각을 Seed로 하여 완전히 Random한 1~999,999 사이의 난수가 생성되는지 Test - 생성된 난수가 화면에 출력되는지 Test	16. Generate Random Number	R5.2
17	임의 방향 생성 시험	- 현재 시각을 Seed로 하여 완전히 Random한 1~60 사이의 난수가 생성되는지 Test - 생성된 난수에 해당하는 LCD가 출력되는지 Test	17. Generate Random Direction	R6.1
18	기능 선택 시험	- User가 원하는 기능을 순서대로 선택할 수 있는지 Test - User가 원하는 기능을 설정하고 제대로 적용되는지 Test	18. Select Function	R7.1



## 2038. Analyze(2030) Traceability Analysis

