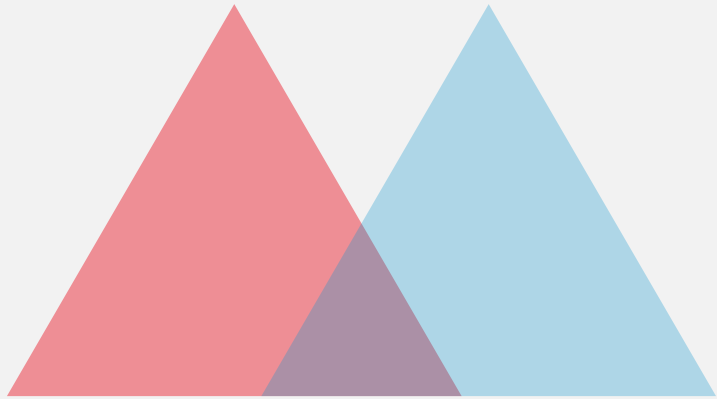
# Design OOPT Stage 2040



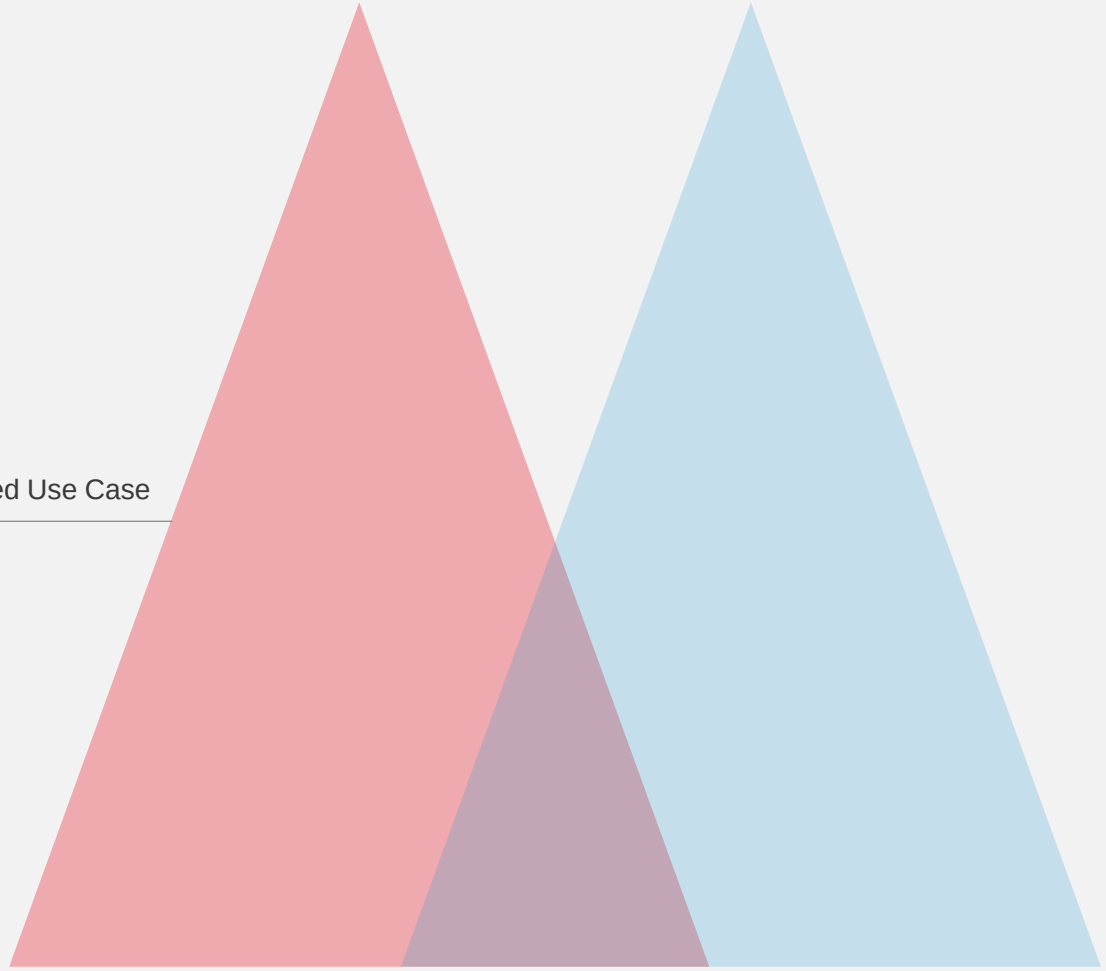
이현우 2015511288 김유진 2018111241 류수진 2018111249 서푸름 2018111265

# Contents

# 2041

Detailed Use Case

## Design Real Use Cases



# 2041

UseCase목록

## 수정된 UseCase목록

1

2041 UseCase

# Design Real Use Cases

| Number | Use case Name | Layer |
|---|---|---|
| R 1.1 | 1. InitWatchState | H |
| R 1.2 | 2. ModeConfig | E |
| R 1.3 | 3. NextMode | E |
| R 1.4 | 4. PrevMode | E |
| R 1.5 | 5. UseCurrentMode | E |
| R 1.6 | 6. BackToMainScreen | E |
| R 2.1 | 7. OnBuzzer | H |
| R 2.2 | 8. OffBuzzer | E |
| R 3.1 | 9. SetSegmentUpper | H |
| R 3.2 | 10. SetSegmentLower | H |
| R 4.1 | 11. SyncWithCurrentTime | H |
| R 5.1 | 12. Start StopWatch | E |
| R 5.2 | 13. Pause StopWatch | E |
| R 5.3 | 14. Continue StopWatch | E |
| R 5.4 | 15. Reset StopWatch | E |
| R 6.1 | 16. Start Timer | E |
| R 6.2 | 17. SetTimerMinute | E |
| R 6.3 | 18. SetTimerSeconds | E |
| R 6.4 | 19. PauseTimer | E |
| R 6.5 | 20. ContinueTimer | E |
| R 6.6 | 21. CancelTimer | E |
| R 6.7 | 22. TimerCheckTime | H |
| R 7.1 | 23. NextAlarm | E |
| R 7.2 | 24. DeleteAlarm | E |
| R 7.3.1 | 25. AddAlarm | E |
| R 7.3.2 | 26. SetAlarmHour | E |
| R 7.3.3 | 27. SetAlarmMinute | E |
| R 7.4 | 28. DecideAndBackToAlarmMode | E |
| R 7.5 | 29. RingAlarm | E |
| R 7.6 | 30. EnableAlarm | E |
| R 7.7 | 31. DisableAlarm | E |
| R 8.1 | 32. NextWorldTime | E |
| R 8.2 | 33. PrevWorldTime | E |
| R 8.3 | 34. HoldCurrentWorldTime | E |
| R 8.4 | 35. ReleaseCurrentWorldTimeLock | E |
| R 8.5 | 36. SyncWorldTime | H |
| R 9.1 | 37. NextTheme | E |
| R 9.2 | 38. PrevTheme | E |
| R 9.3 | 39. DecideTheme | E |
| R 10.1 | 40. SwapUsingMode | E |
| R 11.1 | 41. MappingButtonAction | E |

# 2041

수정되거나, 중요한 내용들 위주로

## 자세한 UseCase

1

2041 UseCase

# Design Real Use Cases

| UseCase | 1. InitWatchState |
|---|---|
| Actor | System |
| Purpose | 시계가 처음 부팅되었을 때, 시계의 상태를 초기화 시켜주어야 한다. |
| OverView | 시계가 부팅될 때 처음 1번만 실행되는 기능으로, 시계가 정상적으로 작동할 수 있도록, 시계의 모든 기능들을 초기화 시키고, 사용자가 사용할 수 있는 상태로 만들어 준다. |
| Type | Hidden |
| Cross Reference | R 1.1 |
| Pre – Requisites | 시계가 처음 부팅된 상태여야 한다. |
| Typical Courses of Events | (U) User (S) System<br>1. (U) User가 시계를 처음 부팅한다.<br>2. (S) 시계의 상태를 초기화 시킨다. (InitWatch)<br>3. (S) 외부의 클래스들이 ModeManager의 기능들을 사용할 수 있도록, ModeManager의 기능들을 등록한다. (RegisterCallBack)<br>4. (S) 모드들을 생성하고 각 모드들을 초기화 시켜준다 (BuildAndInitMode) |
| Alternative Courses of Events | N/A |
| Exceptional Courses of Events | N/A |

| UseCase | 5. UseCurrentMode |
|---|---|
| Actor | User |
| Purpose | 사용자가 현재 모드를 사용할 수 있게 해 준다. |
| OverView | 사용자가 useThisMode를 사용하면, 현재 화면에 표시되고 있는 모드로 진입해서 사용자가 사용할 수 있게 해 준다. |
| Type | Evident |
| Cross Reference | R 1.5 |
| Pre – Requisites | 현재 시계 상태는 MainScreenMode여야 한다. |
| Typical Courses of Events | (U) User (S) System<br>(U) User는 C버튼을 누른다.<br>(S) 현재 화면에 표시되고 있는 모드로 진입한다. (UseCurrentMode)<br>(S) 현재 사용중인 모드를 사용할 수 있는 환경으로 설정해준다. (UseThisMode)<br>1. (S) 사용할 모드가 현재 모드로 변경되면, 디스플레이에 표시되는 화면을 현재 모드의 화면으로 설정해준다.<br>2. (DisplayCurrentMode) |
| Alternative Courses of Events | N/A |
| Exceptional Courses of Events | - A, B, C버튼의 기능은 해당 모드의 기능으로, D버튼의 기능은 BackToMainScreen으로 변경된다. |

1

2041 UseCase

# Design Real Use Cases

| UseCase | 2. ModeConfig |
|---|---|
| Actor | User |
| Purpose | 사용자가 사용하지 않는 기능 2개를 선택하는 화면을 출력한다. |
| OverView | 사용자가 config를 누르면 사용자가 ModeConfig모드로 진입할 수 있도록 처리를 해 준다. |
| Type | Evident |
| Cross Reference | R 1.2 |
| Pre – Requisites | 현재 시계 상태는 MainScreenMode여야 한다. |
| Typical Courses of Events | (U) User (S)System<br>1. (U) User가 D 버튼을 누른다.<br>(S) 만약, 현재 시계 상태가 MainScreenMode 라면, 사용할 모드를 자기 자신으로 설정한다.(UseThisMode)<br>1. (S) 사용하고 있지 않은 모드 2개를 E세그먼트에 출력해 준다. (DisplayUpperSegment)<br>2. (S) 사용하지 않을 모드를 F세그먼트에 출력해준다.<br>3. (DisplayLowerSegment) |
| Alternative Courses of Events | N/A |
| Exceptional Courses of Events | - A 버튼, B 버튼 각각에는 SwapUsingMode가 매핑되고, C 버튼은 매핑 된 기능이 제거되며, D버튼의 기능은 BackToMainScreen으로 변경된다. |

| UseCase | 6. BackToMainScreen |
|---|---|
| Actor | User |
| Purpose | 사용자가 메인 화면으로 돌아갈 수 있게 해준다. |
| OverView | 사용자가 backToMainScreen을 요청하면, 현재 사용중인 모드를 메인 화면 모드로 설정해 주고, 각 버튼의 매핑을 메인 화면 모드에 맞는 매핑으로 바꾸어 준다. |
| Type | Evident |
| Cross Reference | R 1.6 |
| Pre – Requisites | 현재 시계 상태는 MainScreenMode가 아닌 모드 이어야 한다. |
| Typical Courses of Events | (U) User (S) System<br>(U) User가 D 버튼을 누른다.<br>(S) 현재 시계 상태가 MainScreenMode가 아니라면, MainScreenMode로 진입한다. (UseConfigMode)<br>(S) 현재 시계의 상태를 MainScreenMode로 설정하고, 각 버튼의 매핑을 해당 모드의 매핑으로 변경해준다. (UseThisMode) |
| Alternative Courses of Events | N/A |
| Exceptional Courses of Events | - A, B 버튼은 각각, DecreaseIndex/IncreaseIndex로 변화하고, C 버튼은 ActiveUseCurrentMode에, D버튼은 ModeConfig에 매핑된다. |

1

2041 UseCase

# Design Real Use Cases

| UseCase | 7. OnBuzzer |
|---|---|
| Actor | System |
| Purpose | 시스템이 사용자에게 무언가 알려야 할 때, 버저를 울린다. |
| OverView | 타이머나 알람이 일정 시간이 되었을 때, 버저가 울리게 된다.<br>타이머와 알람의 울림이 중복된다면, 먼저 발생한 쪽의 울림을 중지시키고, 새로 울리는 쪽의 울림을 발생시킨다.<br>(단, 근소한 시간동안 꺼졌다 켜지므로, 사용자는 이 끊김을 느낄 수 없다,) |
| Type | Hidden |
| Cross Reference | R 2.1 |
| Pre – Requisites | N/A |
| Typical Courses of Events | (S) System<br>(S) 사용자가 설정한 일정 시간이 되면(혹은 지나면) OnBuzzer 기능이 불린다.<br>(S) 소리를 재생시킨다. (OnSoundPlayer)<br>(S) 30초 후에 OffBuzzer가 자동으로 불리게 된다(ReserveBuzzerOff) |
| Alternative Courses of Events | N/A |
| Exceptional Courses of Events | - 알람이 울리고 있을 때, 모든 버튼에 매핑되는 기능을 버저 정지로 바꾼다. (setMappingDisable) |

| UseCase | 8. OffBuzzer |
|---|---|
| Actor | System |
| Purpose | 시스템이 사용자에게 알릴 버저를 울리고 있을 때, 버저를 끈다. |
| OverView | 버저가 울리고 있는 상태일 때(사용자), 사용자가 offBuzzer를 누르면 버저를 끈다.<br>버저가 울리고 있는 상태일 때(시스템), 30초가 지나면 자동으로 버저가 꺼진다.<br>버저가 울리지 않고 있었다면, 아무 일도 일어나지 않는다. |
| Type | Evident |
| Cross Reference | R 2.2 |
| Pre – Requisites | N/A |
| Typical Courses of Events | (S) System<br>(S) 알람이 울리고 있을 때, 사용자가 offBuzzer를 누르면 알람을 종료하고 (OffBuzzer), 소리를 끈다. (OffSoungPlayer)<br>(S) 알람이 울리고 있을 때, onBuzzer가 불리고 난 뒤에 30초가 지나면 버저가 자동적으로 종료된다. (ReserveBuzzerOff)<br>(S) Off알람이 예약되어 있었다면 예약을 해제한다. |
| Alternative Courses of Events | N/A |
| Exceptional Courses of Events | - 이 기능이 호출되었을 때, 버저가 울리고 있지 않았다면 아무 일도 일어나지 않는다. |

2041 UseCase

# Design Real Use Cases

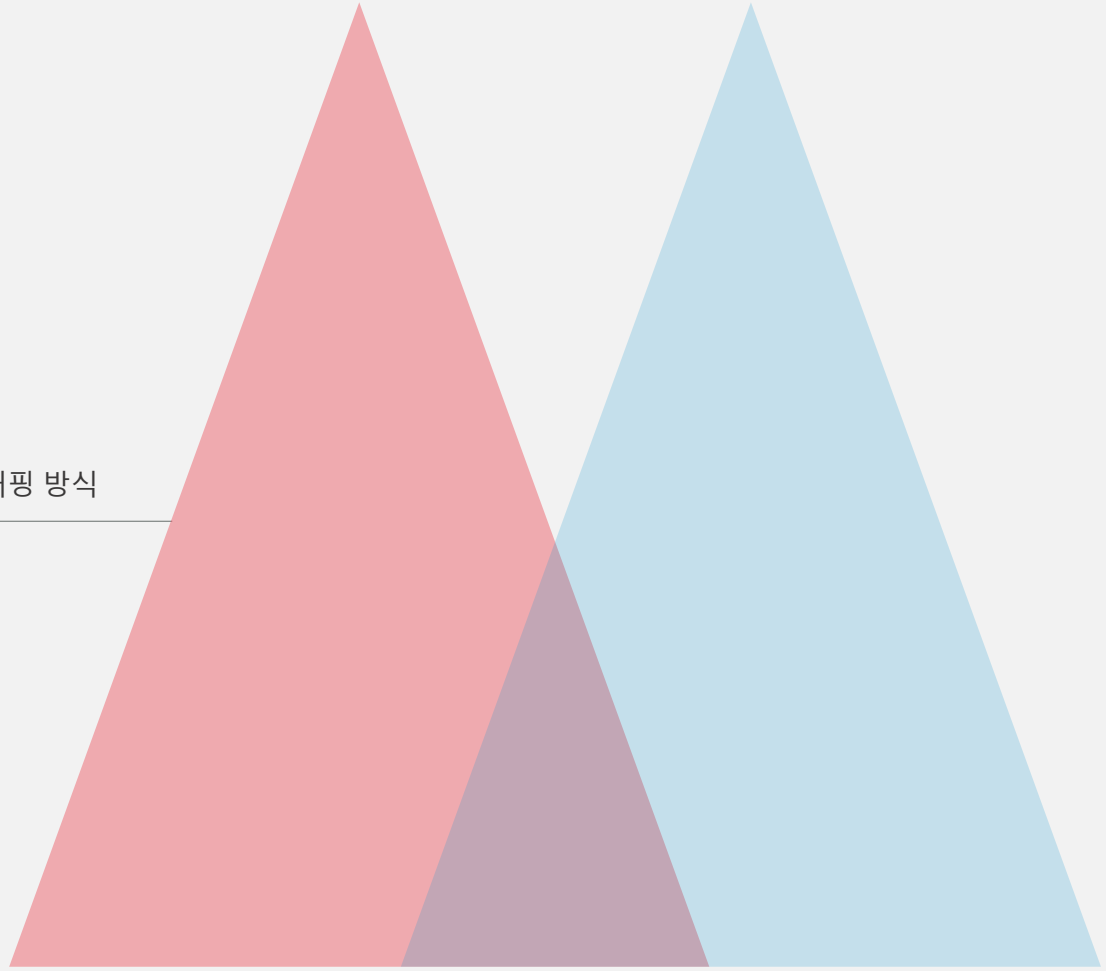
| UseCase | 9. SetSegmentUpper |
|---|---|
| Actor | System |
| Purpose | 상단 세그먼트에 표시되는 내용을 설정한다. |
| OverView | 사용자가 선택한 Mode에 관한 입력 받은 문자열로, 세그먼트에 표시되는 내용을 변경한다. |
| Type | Hidden |
| Cross Reference | R 3.1 |
| Pre – Requisites | N/A |
| Typical Courses of Events | (S) System<br>(S) SetSegmentUpper가 호출된다.<br>(S) 설정 인자로 들어온 값의 길이를 계산해서, 길다면 잘라주고, 작다면, 0을 붙여서 길이를 늘려준다. (TrimUpperElement)<br>(S) 수정된 값을, 현재 세그먼트에 출력되고 있는 값에 반영한다. (StoreUpperData) |
| Alternative Courses of Events | N/A |
| Exceptional Courses of Events | N/A |

| UseCase | 10. SetSegmentLower |
|---|---|
| Actor | System |
| Purpose | 하단 세그먼트에 표시되는 내용을 설정한다. |
| OverView | 사용자가 선택한 Mode에 관한 입력 받은 문자열로, 세그먼트에 표시되는 내용을 변경한다. |
| Type | Hidden |
| Cross Reference | R 3.2 |
| Pre – Requisites | N/A |
| Typical Courses of Events | (S) System<br>(S) SetSegmentLower가 호출된다.<br>(S) 설정 인자로 들어온 값의 길이를 계산해서, 길다면 잘라주고, 작다면, 0을 붙여서 길이를 늘려 준다. (TrimLowerElement)<br>(S) 수정된 값을, 현재 세그먼트에 출력되고 있는 값에 반영한다. (StoreLowerData) |
| Alternative Courses of Events | N/A |
| Exceptional Courses of Events | N/A |

# 2042

버튼 & UI의 매핑 방식

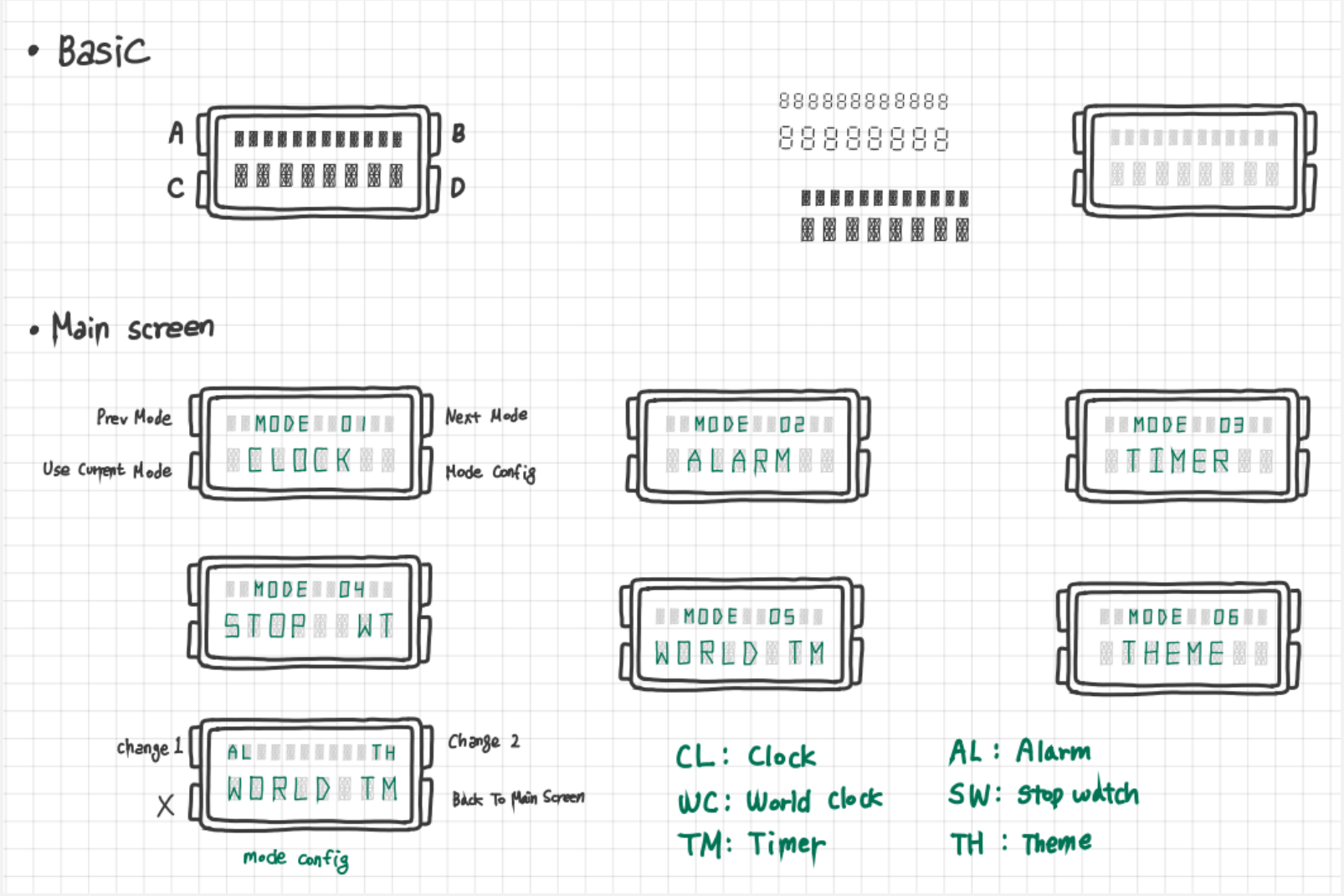
## Define Reports, UI, and Storyboards



2

2042 Mapping

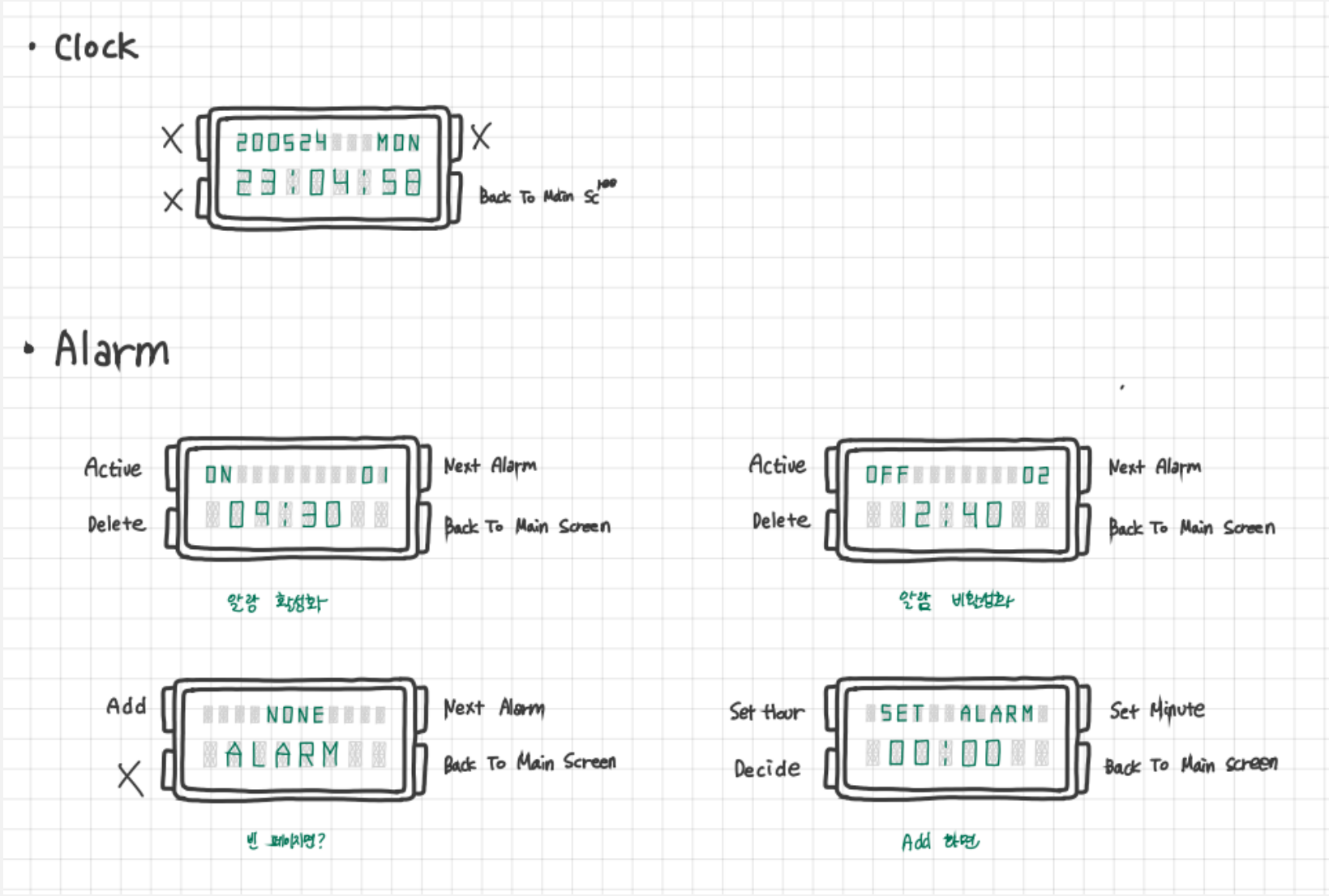
# Define Reports, UI, and Storyboards



2

2042 Mapping

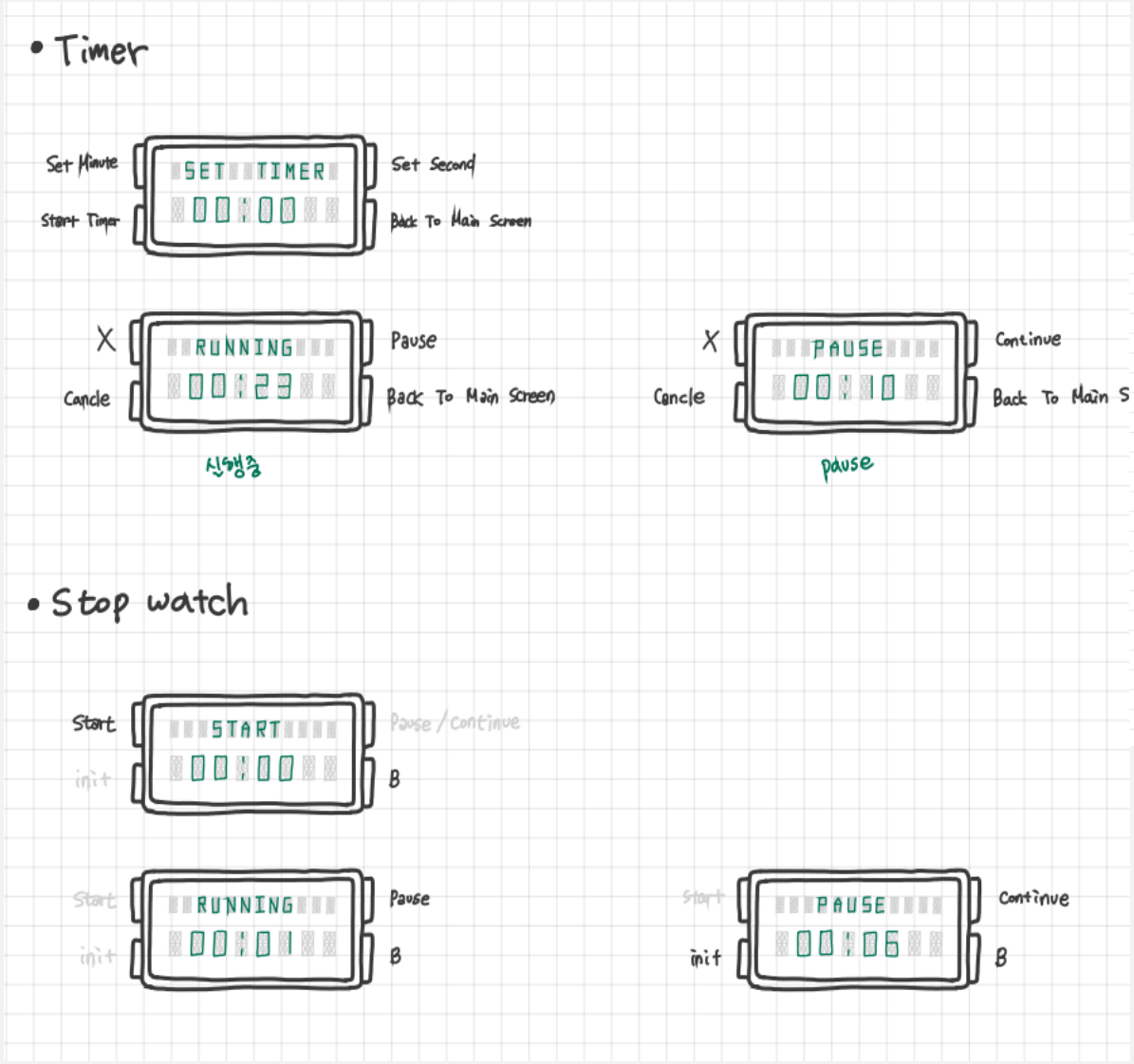
# Define Reports, UI, and Storyboards

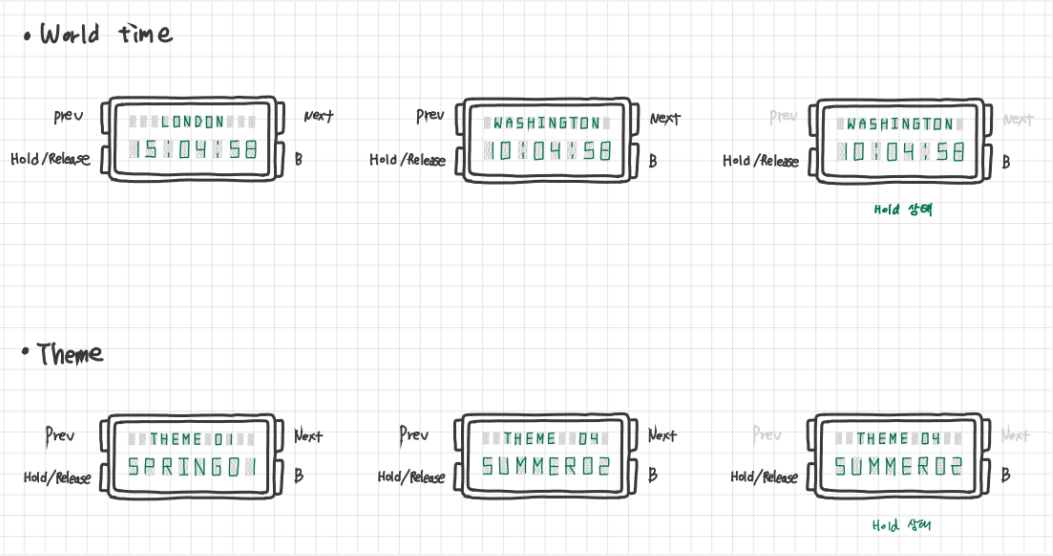


2

2042 Mapping

# Define Reports, UI, and Storyboards



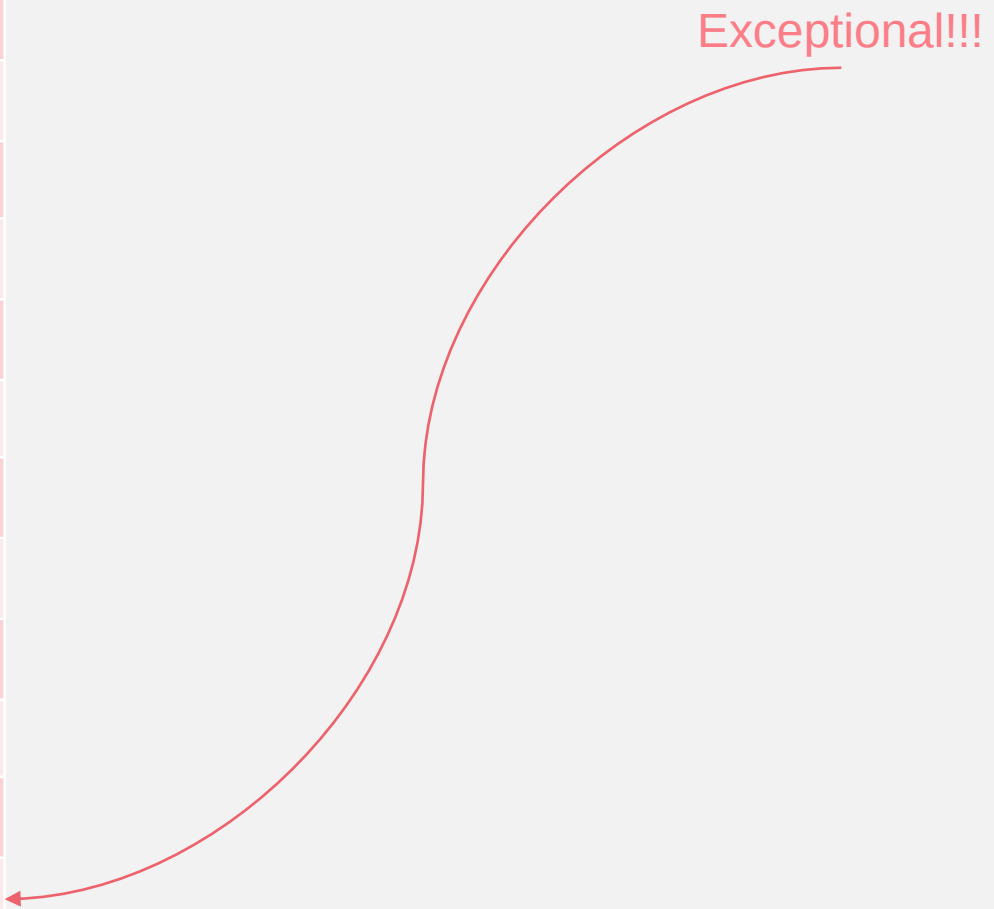



2

2042 Mapping

# Define Reports, UI, and Storyboards

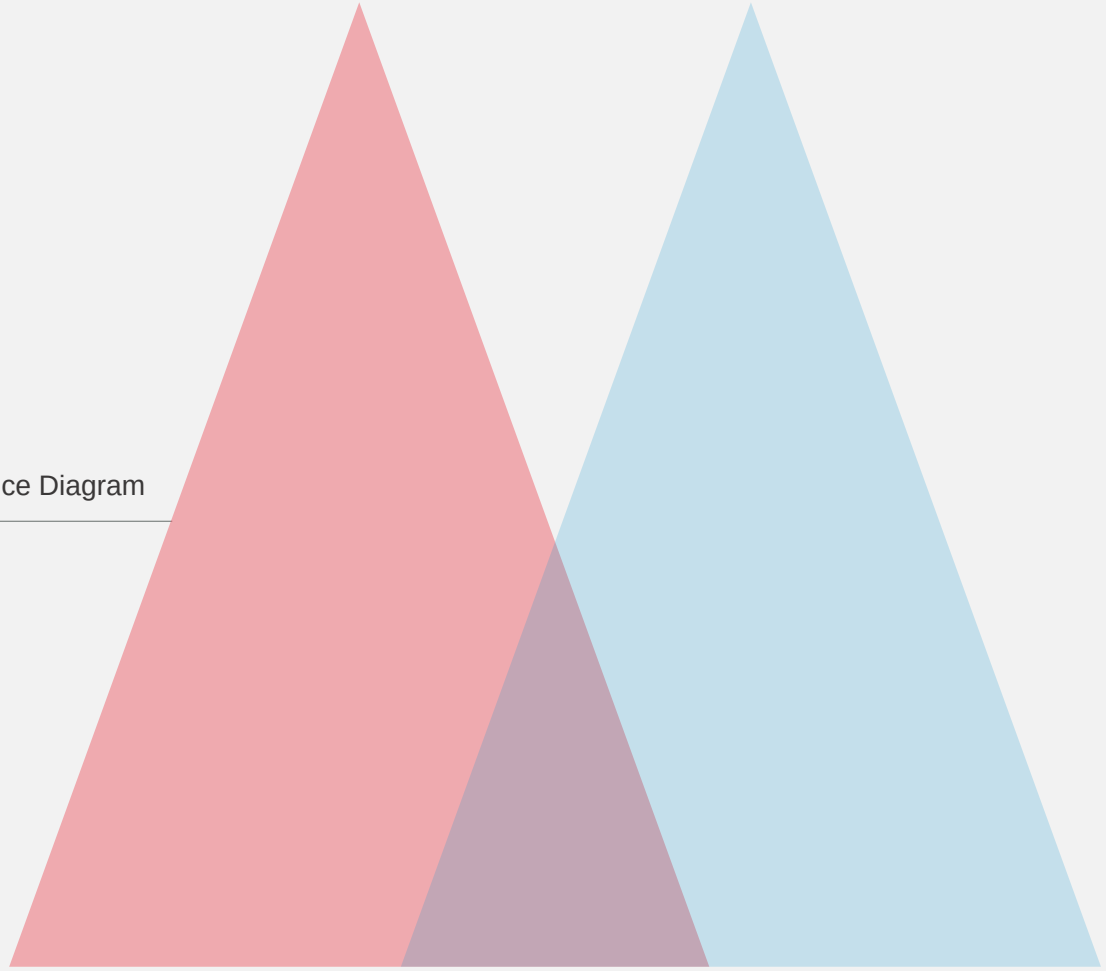
| Mode | ButtonA | ButtonB | ButtonC | ButtonD |
|---|---|---|---|---|
| **MainScreen** | PrevMode | NextMode | UseThisMode | ModeConfig |
| **ModeConfig** | SwapUsingMode | SwapUsingMode | N/A | BackToMain |
| **Colck** | N/A | N/A | N/A | BackToMain |
| **Alarm(Exist)** | Active/Disable | NextAlarm | Delete | BackToMain |
| **Alarm(Empty)** | AddAlarm | NextAlarm | Delete | BackToMain |
| **AlarmTimeSetting** | SetHour | SetMinute | Decide | BackToMain |
| **Timer** | SetMinute | SetSeconds | StartTimer | BackToMain |
| **Timer(Started)** | N/A | Pause/StopTimer | Cancel | BackToMain |
| **StopWatch** | Start | Pause/Continue | Init | BackToMain |
| **WorldTime** | PrevWorld | NextWorld | Hold/Release | BackToMain |
| **Theme** | PrevTheme | NextTheme | Decide | BackToMain |
| **BuzzerActived** | OffBuzzer | OffBuzzer | OffBuzzer | OffBuzzer |



# 2043

Sequence Diagram

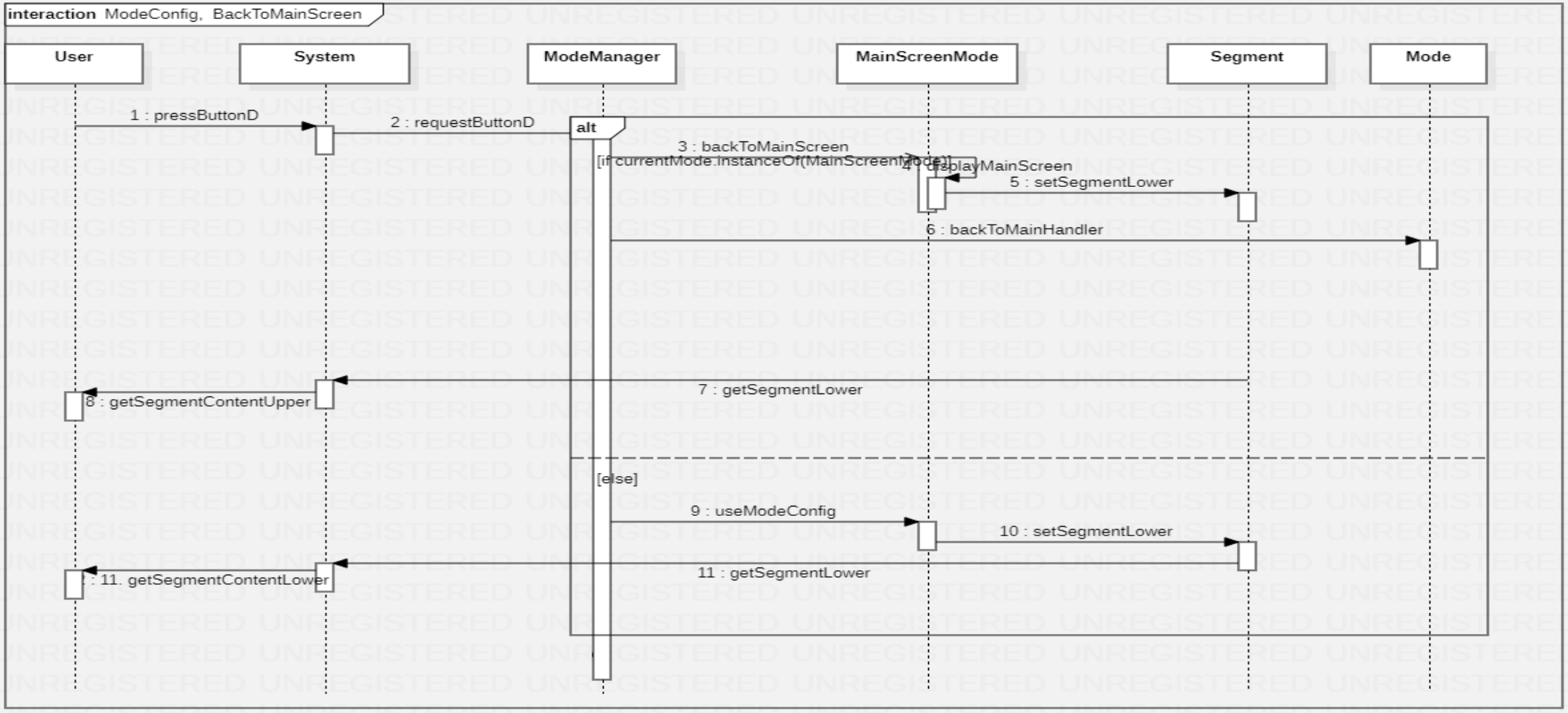
## Refine System Architecture



3

2043 Mapping

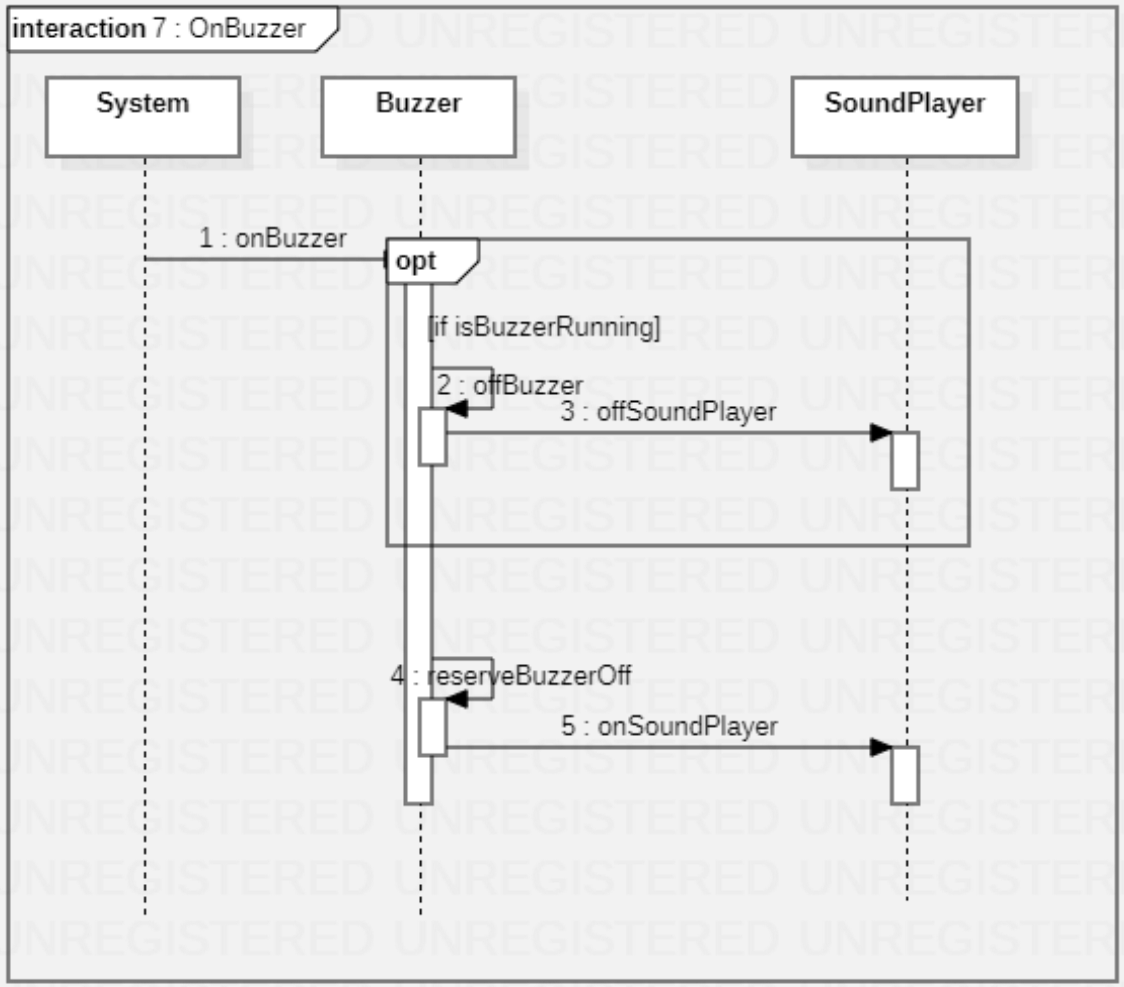
# Refine System Architecture

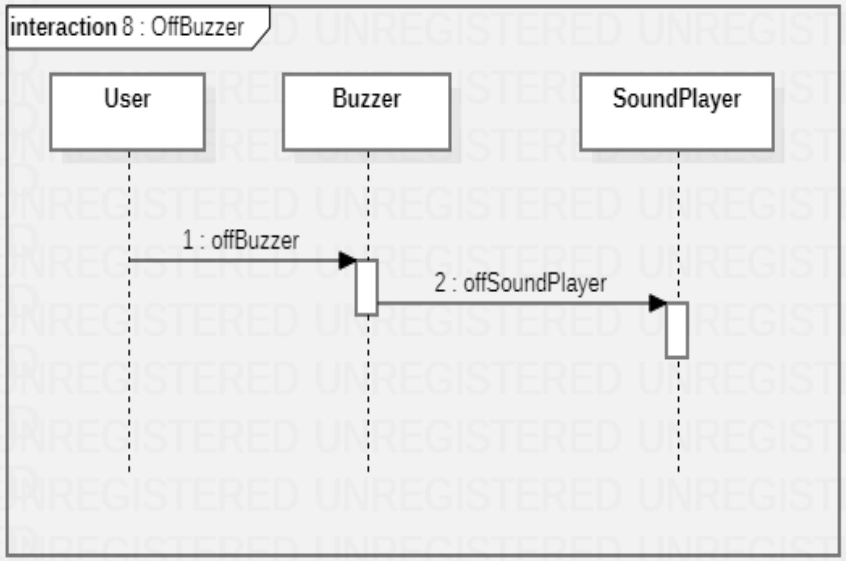


3

2043 Mapping

# Refine System Architecture

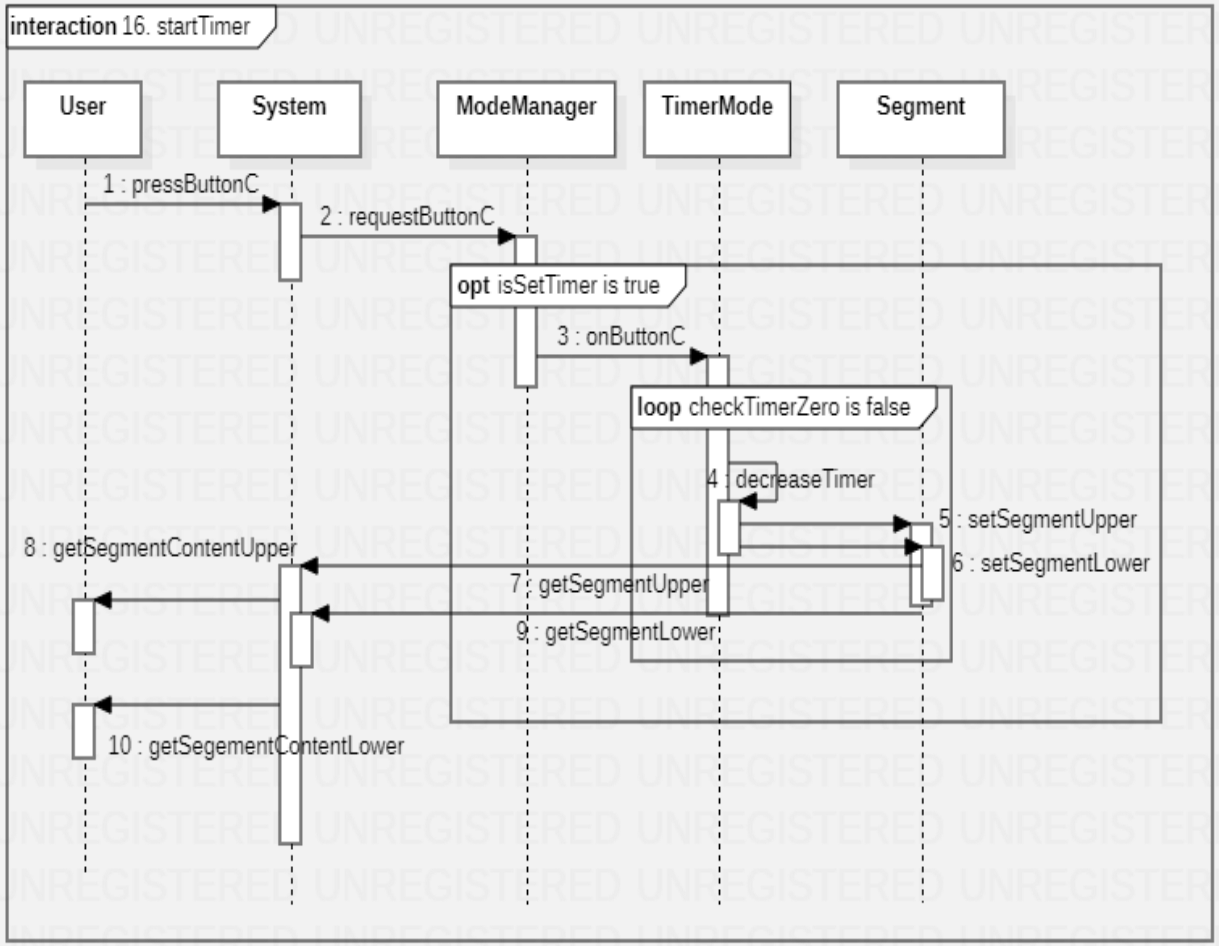


3

2043 Mapping

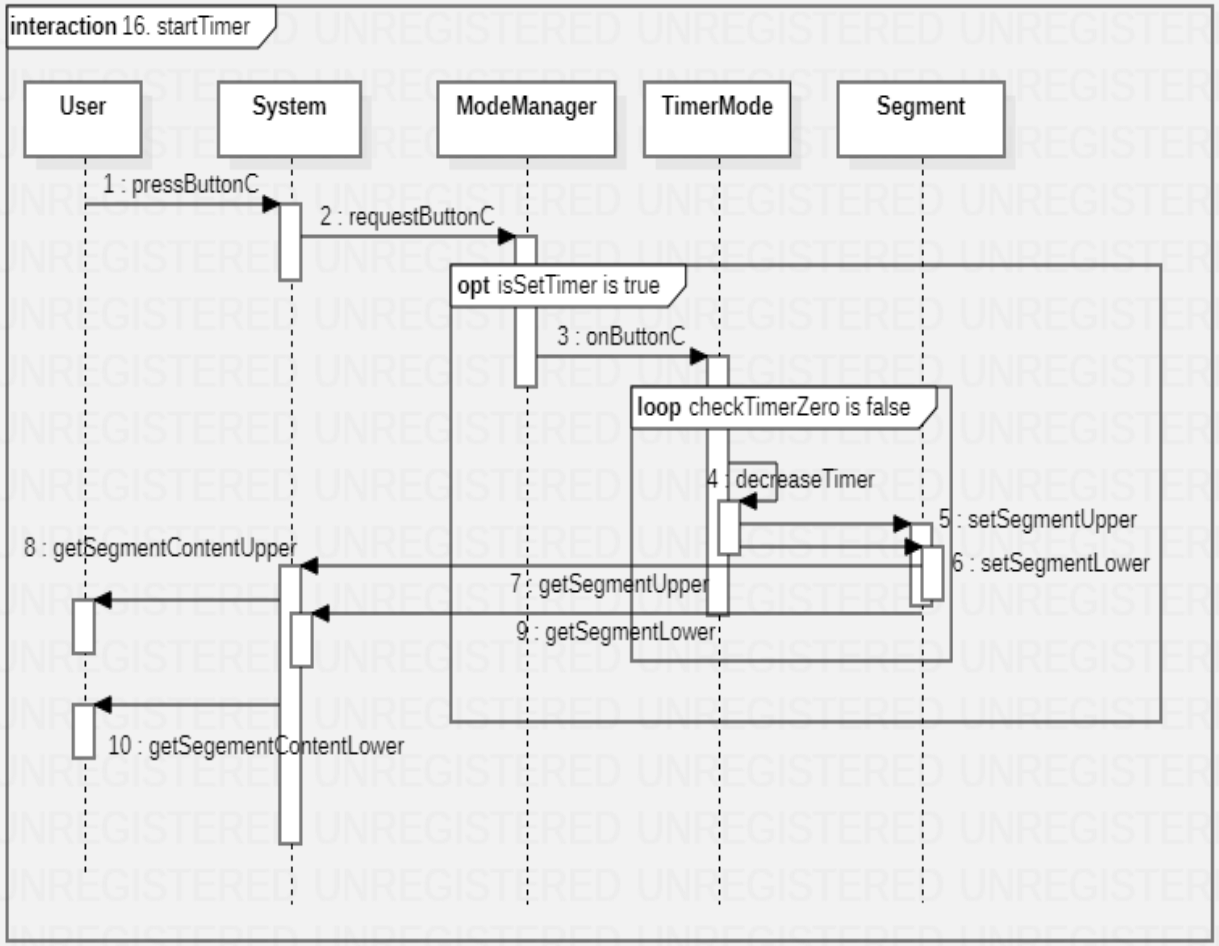
# Refine System Architecture



3

2043 Mapping

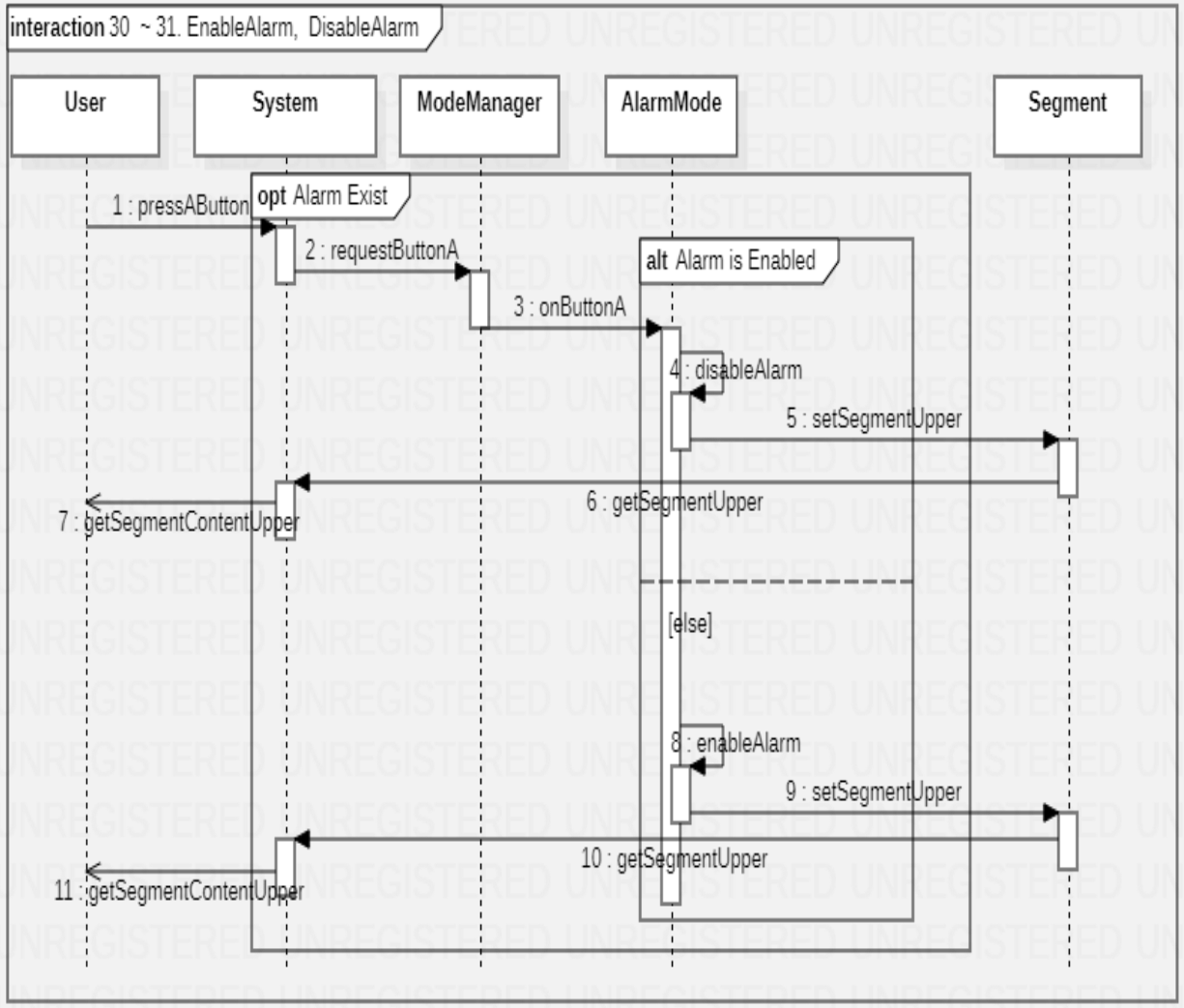
# Refine System Architecture



3

2043 Mapping

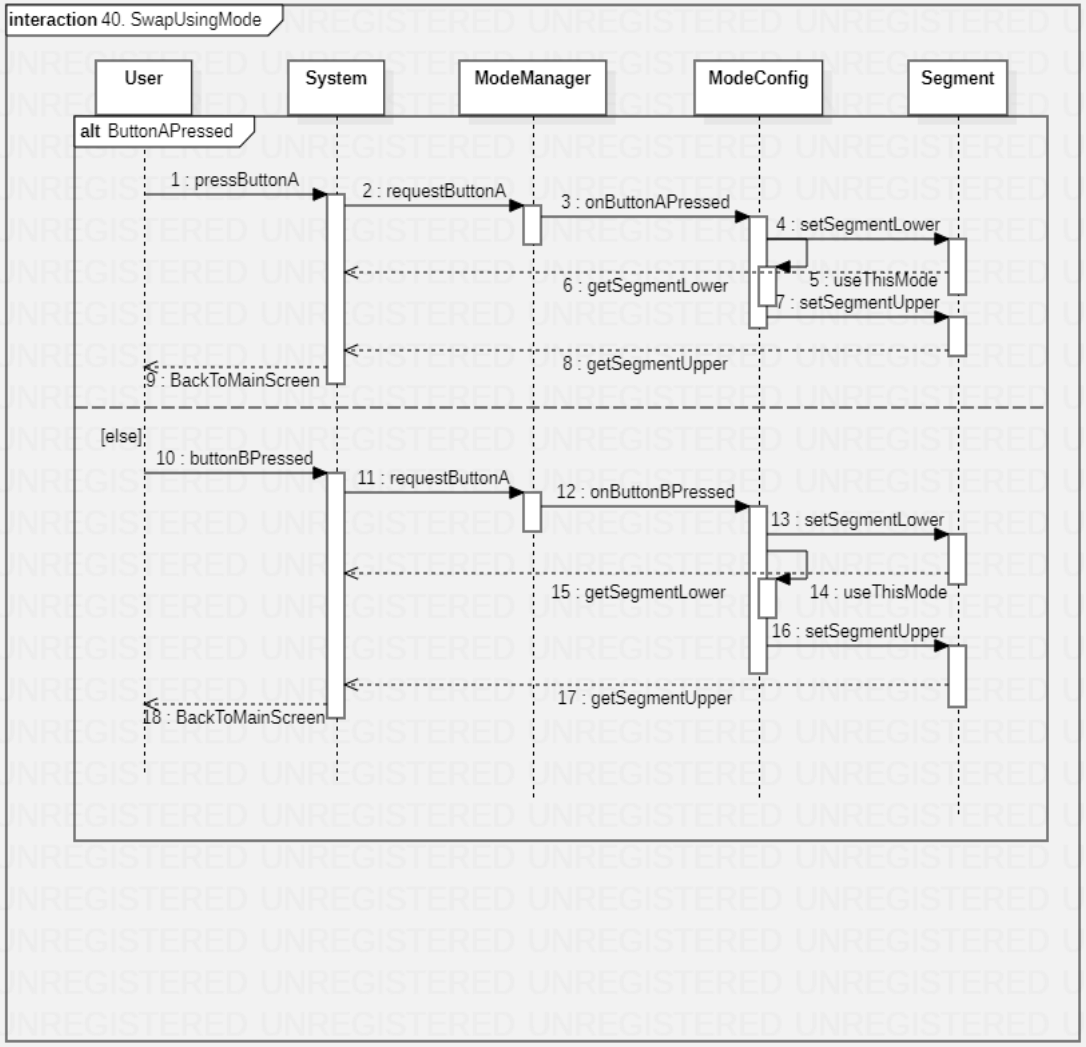
# Refine System Architecture



3

2043 Mapping

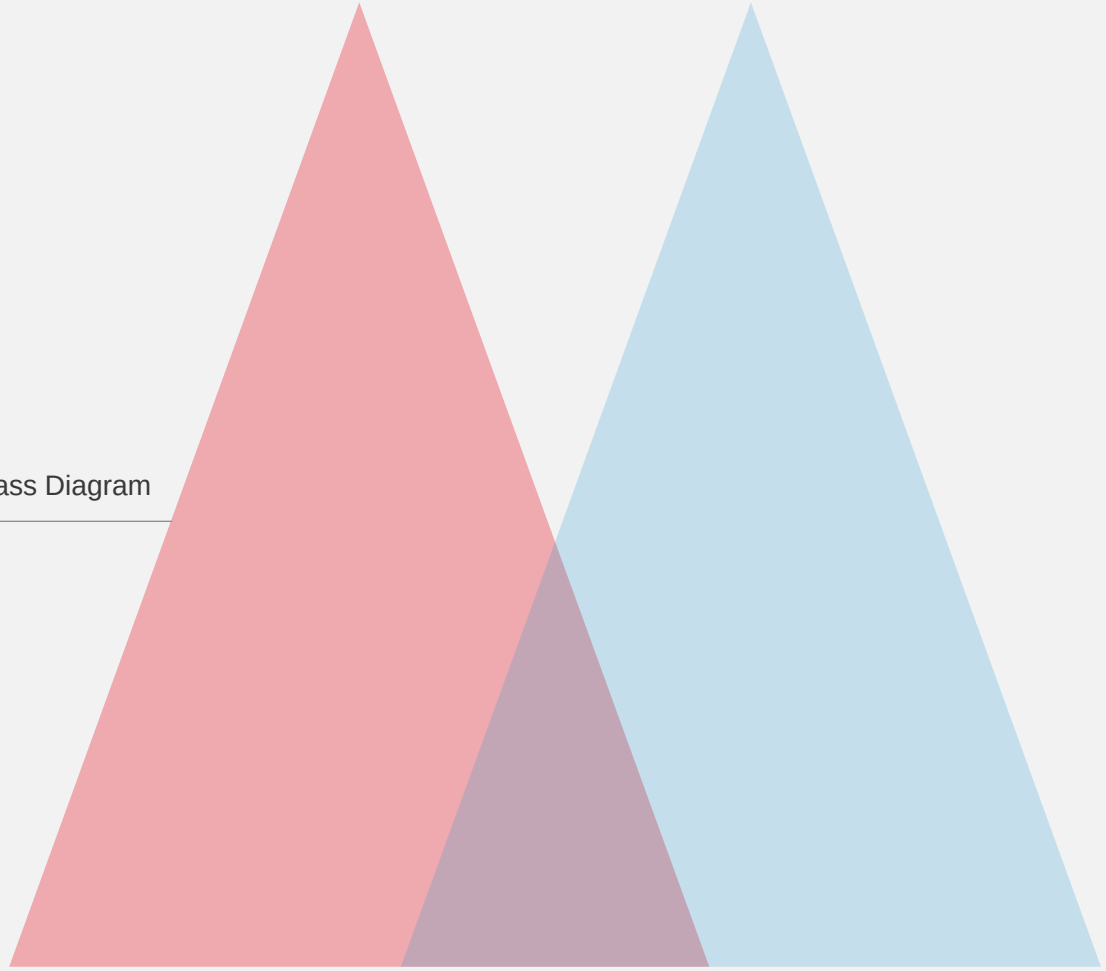
# Refine System Architecture



# 2044

Class Diagram

## Define Design Class Diagram



2044 ClassDiagram

# Define Design Class Diagram

## 설계 철학

- 변화가 가능한 부분은 인터페이스로 처리하고, 인터페이스 마저도, 다른 인터페이스의 구성 요소가 될 수 있다.
- 중복 참조가 발생한다면, 인터렉터 클래스를 통해서, 두 클래스의 콜백 메소드를 등록하고, 해당 메소드를 호출하는 것으로 해결한다.
- 클래스를 최대한 세분화 하되, 이유없이 클래스를 분리하지는 않는다
- 함수 역시도 최대한 작은 단위로 나누어서, 각각의 함수의 복잡도를 낮추자.

4

2044 ClassDiagram

# Define Design Class Diagram

## 설계 방식

- 시퀀스 다이어그램 이전에, 비즈니스 오브젝트 다이어그램을 참조하여, 대략적인 형태의 클래스 다이어그램을 작성한다.
- 해당 클래스 다이어그램은 그대로 두고, UseCase및 시퀀스 다이어그램을 작성하되, 각 UseCase에서 공통적으로 필요한 객체들을 뽑아내서 계속적으로 클래스 다이어그램에 추가한다.
- 이후, 클래스 다이어그램에 시퀀스 다이어그램을 반영하고, 다시 시퀀스 다이어그램에 클래스 다이어그램을 반영하는 과정을, 형태가 갖추어 질 때 까지 반복한다.

4

2044 ClassDiagram

# Define Design Class Diagram

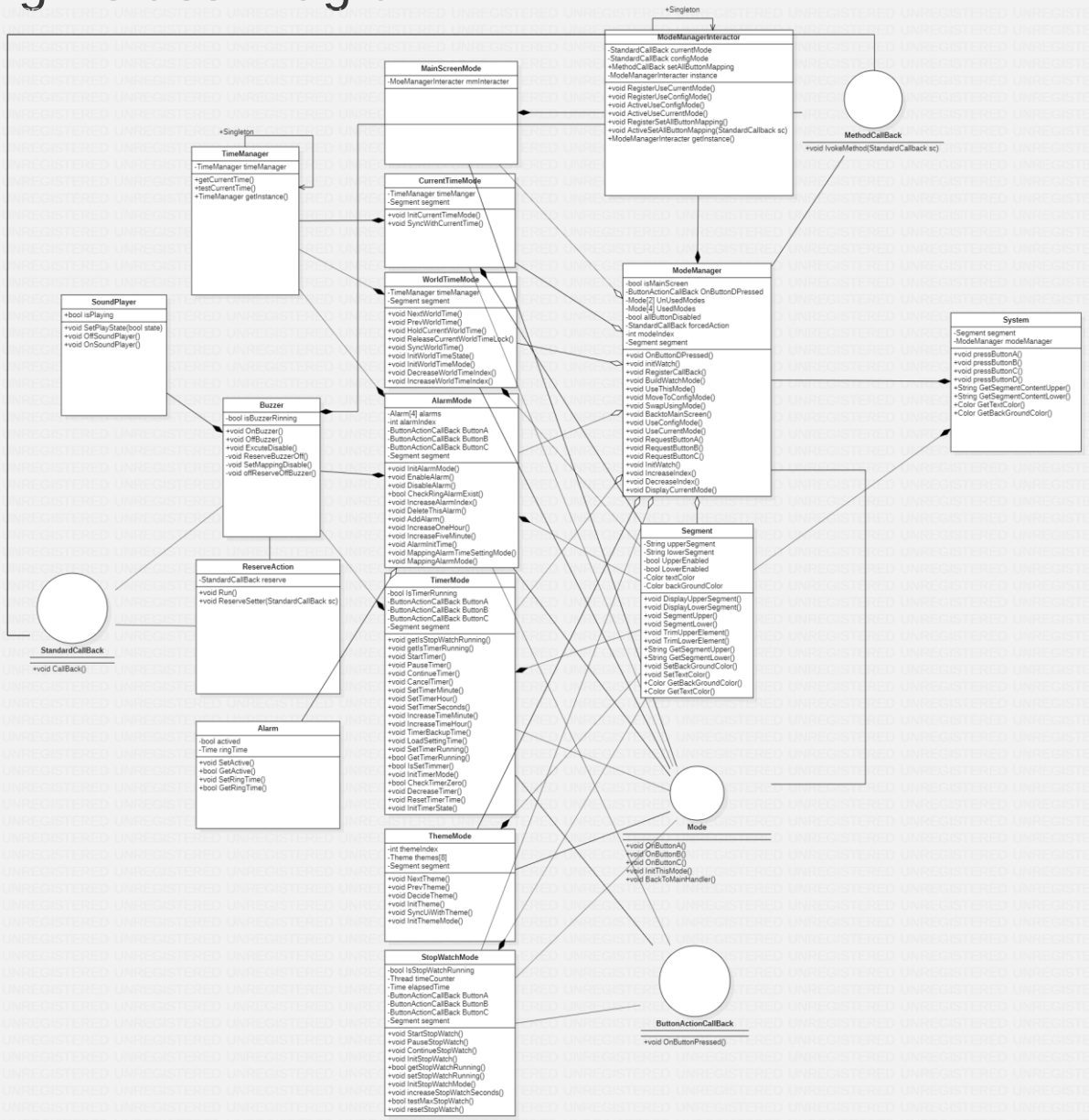
## 도전해 본 점들

- 작년의 결과물을 보면, Model과 View를 연결해주는 클래스로, System클래스를 많이 잡았음(System, WatchSystem 등등)
- View와 Controller의 UseCase요청이 System에 집중되는 구조여서, System이 거대해짐
- 이런 부분에 대해서 System을 경량화 할 수 없을까?
- Interface를 통해서, 현재 사용중인 모드가, Model과 View와 연결되도록 하고, System은 중계만 하자!

4

2044 ClassDiagram

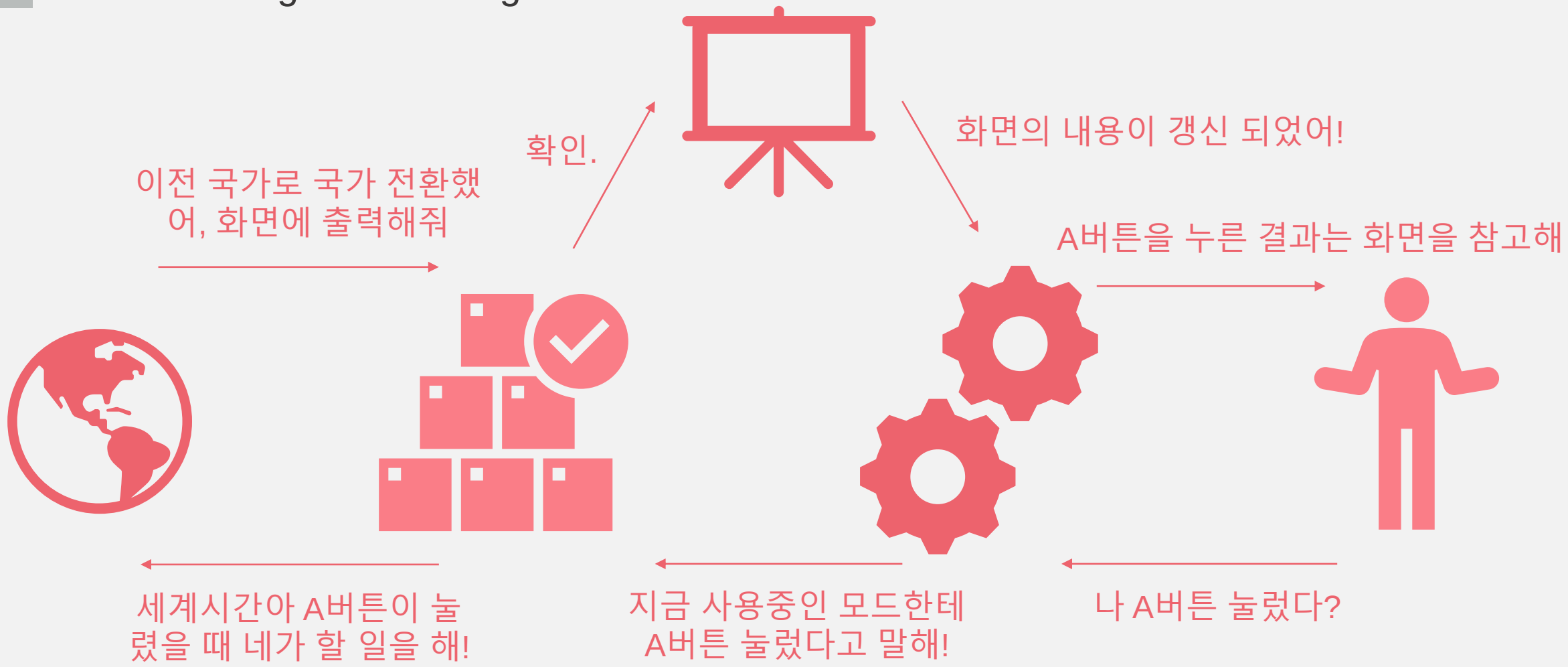
# Define Design Class Diagram



4

2044 ClassDiagram

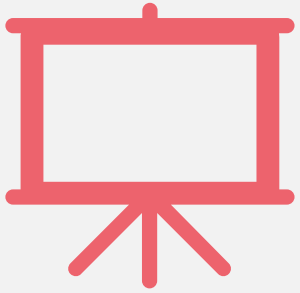
# Define Design Class Diagram



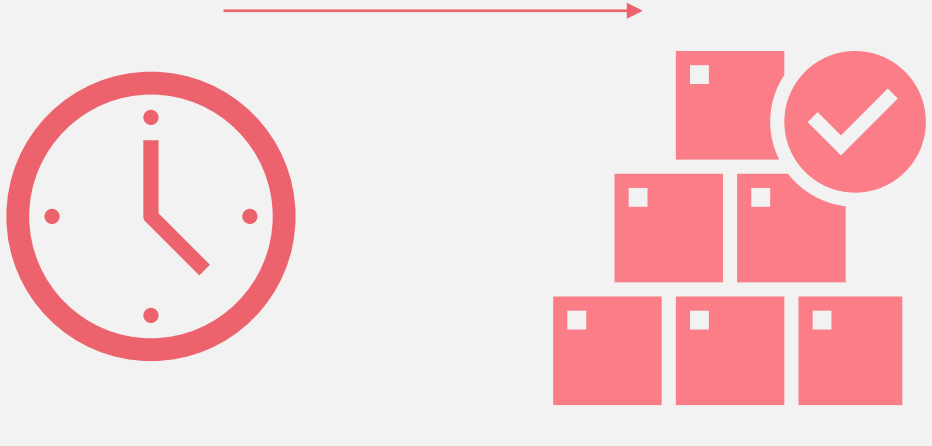
4

2044 ClassDiagram

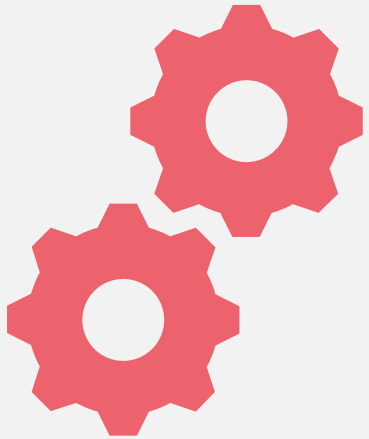
# Define Design Class Diagram



나는 A버튼을 눌렀을 때
아무것도 안 하지롱~



현재시간아 A버튼이 눌
렸을 때 네가 할 일을 해!

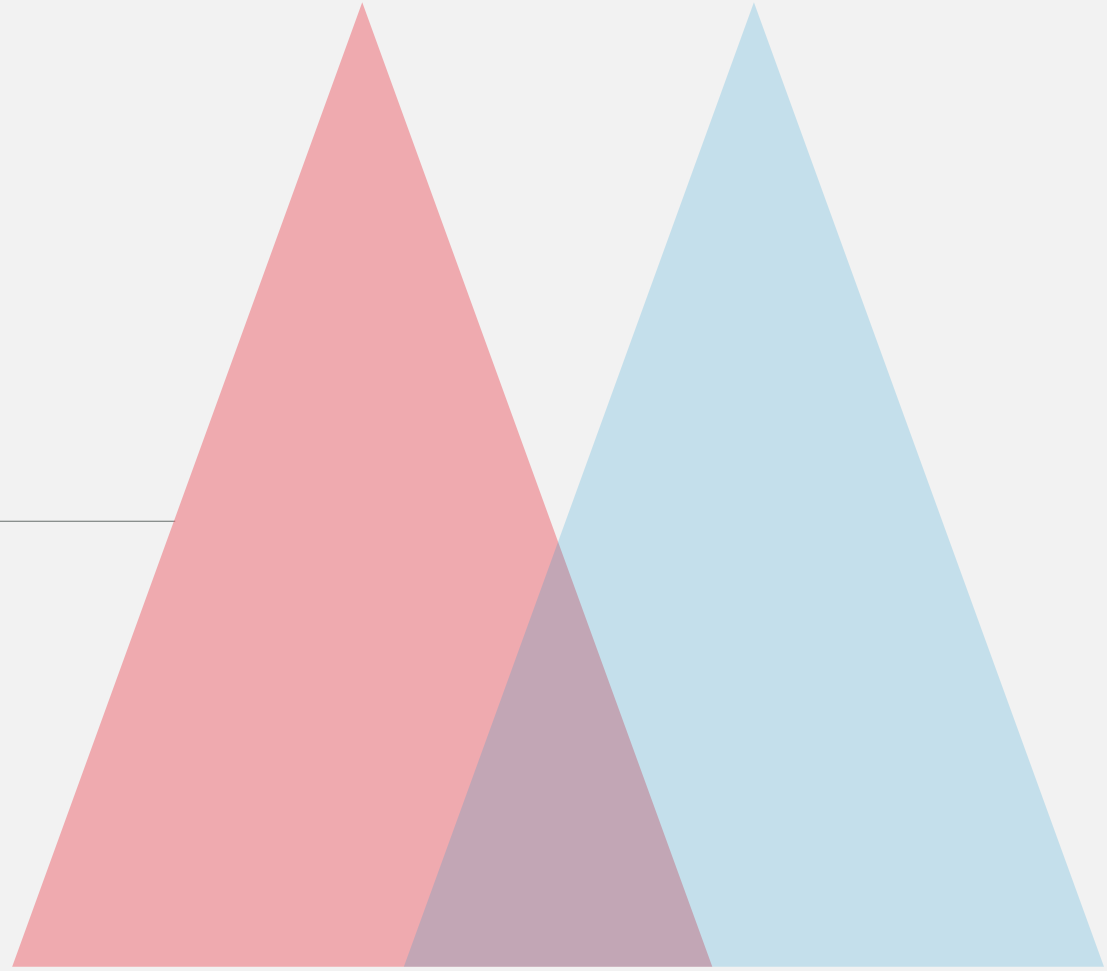


지금 사용중인 모드한테
A버튼 눌렀다고 말해!

나 A버튼 눌렀다?

# 2045

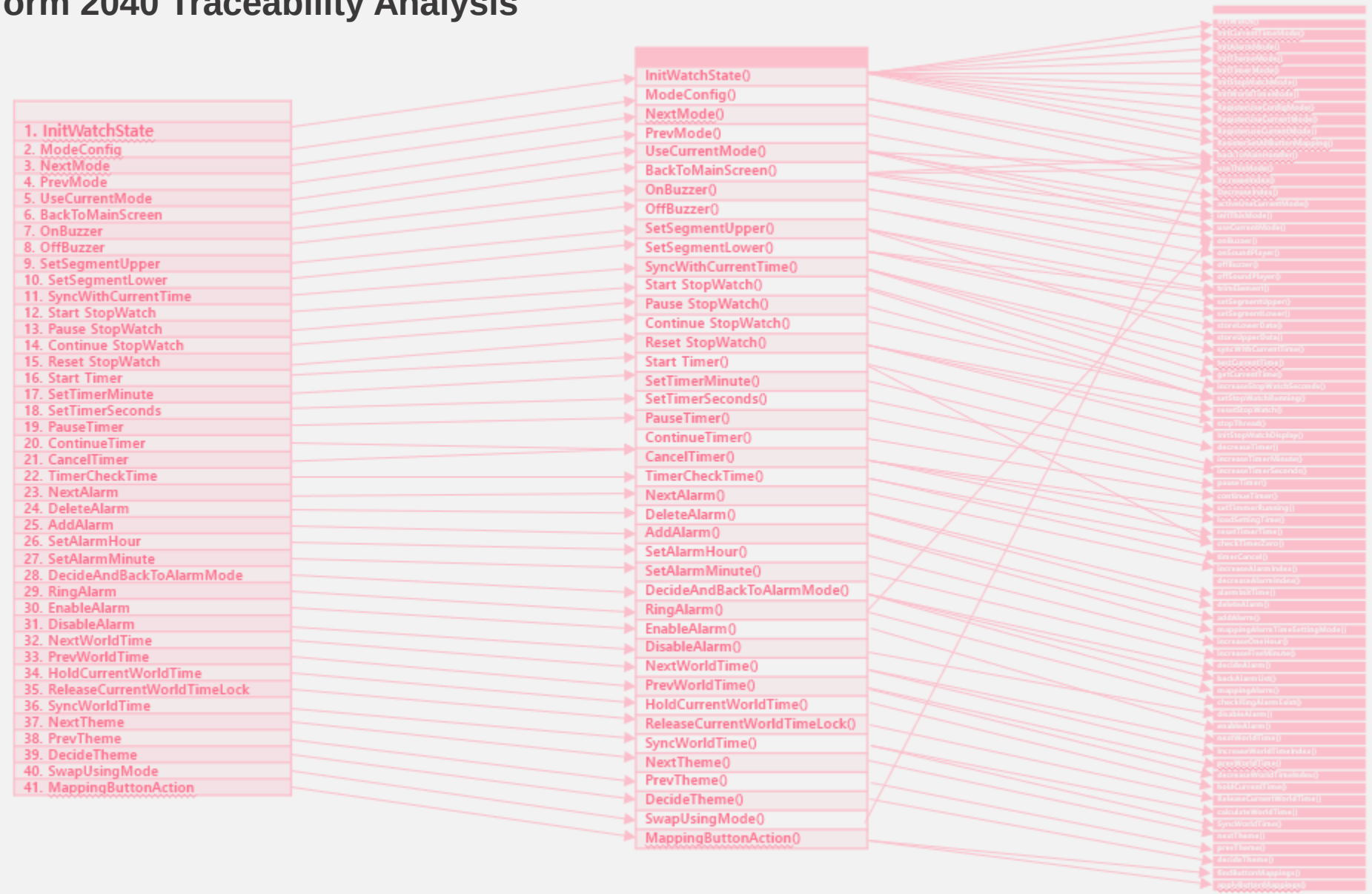
## Perform 2040 Traceability Analysis



5

2047 ClassDiagram

# Perform 2040 Traceability Analysis



# 감사합니다!

질문 받습니다.