Team UML

CSCI 360
Spring 2011

Jaime Miller
Shannon Fox
David Schirduan
Jeffrey Decker
Henry Dineen

Table of Contents

Vision

The alarm clock is to keep accurate time and display the time on a front-facing screen in a 12-hour rotation. The user is able to change the time displayed on the the screen and set two alarms to two separate designated times and alarm types. The user will have the option to have an alarm sound or the radio begin to play at the set alarm time. When the alarm sounds the user is able to turn the alarm off or have the alarm enter a 10-minute snooze mode at the end of which the alarm will activate again. The user will also have the option to simply play the radio at any time they wish and select what ever station they desire. There will also be the ability to adjust the sound volume via a dial on the side of the radio. If wall power is ever lost, the radio will switch its power source to a battery backup (if batteries are installed) to prevent failure of alarm going off and maintain functionality.

Use Case UC1: Setting the Time

Scope: Dual Alarm Clock/Radio
Level: User goal
Primary Actor: User
Stakeholders and Interests:

- User who wants to keep up with the current time and possibly as the standard to measure an alarm time against.

Preconditions:

- The clock is activated with a continuous power supply.

Postconditions:

- Current time is saved and continues to keep accurate time from that point as long as the unit does not lose power.

Main Success Scenario:

1. User first presses the set time button.
2. Clock display begins to blink 12:00 AM.
3. User presses and depresses the hour button until the desired hour (displayed on the clock) is reached.
   Note: Each time the user passes hour 12, the AM and PM indicator light will switch to the appropriate designation.
4. User presses and depresses the minute button until the desired minute (displayed on the clock) is reached.
5. User presses the set time button once more to end the setting process.
6. The clock will begin operating at the time as set by the user.

Extension:

1. User presses the set time button, pauses and then presses the set time button once again. The time will be set as 12:00 AM and begin operating with this as the time set by the user.

Failure Scenario:

1. User first presses the set time button.
2. Clock display begins to blink 12:00 AM.
3. User presses and depresses the hour button until the desired hour (displayed on the clock) is reached.
4. User presses and depresses the minute button until the desired minute (displayed on the clock) is reached.
5. User does not press the set time button to end the process.
6. The clock will not save and activate at the time that was entered by the user.

Use Case UC2: Setting Alarms

Scope: Dual Alarm Clock/Radio
Level: User goal
Primary Actor: User
Stakeholders and Interests:

- User: want to set the alarm, and have it saved for activation later.
- User-Dependents: requires the user to wake up at the correct time

Preconditions:

- The clock is turned on, with a continuous power supply, and set to the correct current time.

Postconditions:

- Alarm time is saved, and ready to be activated at the correct time.

Main Success Scenario:

1. User pushes the desired Alarm Button. (Note: the below processes are identical for both alarms.)
2. Clock displays the current alarm time (blinking).
3. User uses the Minute and Hour buttons to change the alarm time's minutes and hours, respectively.
   Note: To change the AM/PM setting, the user will need to cycle through to the next 12-hour period (e.g. to change the alarm from 4:00 AM to 6:00 PM, they will have to first cycle 4:00 AM to 4:00 PM, then add two more hours. The change in time will be shown by an AM/PM indicator light).
4. Clock's display updates to reflect these changes.
5. User pushes the respective alarm button again.
6. The clock will save the set time as the new alarm time.
7. Clock resumes displaying the current time, which is unaffected by the changing of the alarm time.
8. The user switches the appropriate Alarm toggle from Off to Buzzer.

Extensions:

3a. If the user does not press the alarm button when the desire alarm-time is reached, it will not save the new alarm-time.
    1. The user will have to start over from step 1 to set desired alarm time.

3b. To reset the alarm time to the default 12:00 AM, the user holds down both the hour and minute buttons.

6a. If the clock does not correctly save the new alarm time, it will default to 12:00AM.

8a. The user switches the appropriate Alarm toggle from Off to Radio.
    1. The user turns on the radio.
    2. The user adjusts the volume to the desired level.
    3. The user sets the station and band.
    4. The user turns off the radio when finished.

Use Case UC3: Responding to Alarm

Scope: Dual Alarm Clock/Radio
Level: Sub-function
Primary Actor: User
Stakeholders:

- User: Desires the alarm to sound at the set time to wake up when needed.
- User-Dependents: Desire the user to wake up to be able to do what they need to.
- Manufacturer: Wants to produce a well working product that does what it claims.

Preconditions:

- The clock is turned on, with a continuous power supply, and set to the correct current time.
- Alarm time is set to the desired time.
- Alarm is activated properly.
- The volume is at a reasonable level for the user to hear if radio is activated as alarm type.

Postconditions:

- The user wakes up in time.

Main Success Scenario:

1. The alarm is activated and the buzzer begins to ring.
2. The user presses the Off button.
3. The alarm stops ringing.

Extensions:

*a. Both alarms are set for the same time.
    1. Alarm 1 takes priority and all settings for Alarm 2 are ignored.
    2. The user toggles the alarm switch from buzzer to radio or the reverse:
        1. While the alarm is sounding:
            1. The alarm type will change also, even in the middle of sounding.
        2. Before the alarm sounds:
            1. When the alarm is activated the new alarm type will sound.
        3. During snooze mode:
            1. The 10 minute snooze time ends.
            2. The new alarm type sounds.

3a. The alarm is set to radio instead of buzzer.
    1. The alarm activates and the radio is turned on playing the set station.
    2. The user presses the stop button.
    3. The radio stops playing.

3b. The other alarm activates while the first to go off is still activated.
    1. The first alarm to go off is automatically reset to go off when the times match again.
    2. The second to go off begins sounding.
    3. The user then has the same options to either enter snooze mode or stop the alarm.

4a. The user presses the snooze button.
    1. The alarm (radio or buzzer) stops.
    2. The alarm is temporarily set to 10 minutes from the pressing of the snooze button.
    3. When the time matches the desired alarm type (radio or buzzer) will play again.

4a. The other Alarm activates while the first to go off is in snooze mode (during the 10 minute count down).
    1. The alarm in snooze mode is reset to go off at the original set time.
    2. The second to go off then takes over and begins sounding.
    3. The user then has the same options to either enter snooze mode or stop the alarm.
4. The cycle continues until the user presses the stop button.
5. When the stop button is pressed the alarm is reset to its original set time.

4b. The user is not present or does not do anything.
1. The alarm continues to go off on the desired setting (radio or buzzer).
2. After the alarm is continuously activated for 10 minutes (buzzer or radio) without intervention it automatically turns off.
3. The alarm is reset to sound at the original set time.

Frequency of Occurrence: Once daily per alarm.

Use Case UC4: Operating the Radio

Scope: Dual Alarm Clock/Radio
Level: User Goal
Primary Actor: User
Stakeholders and Interests:
- User: Desires the radio to play when wanted.
- User-Dependents: Desire the user to listen to the radio.

Preconditions:
- The Radio is turned on while having a power supply.
- The Radio is properly activated.
- The Volume is adjusted.

Postconditions:
- The last radio station played will be the first the next time the radio is turned on.

Main Success Scenario:
1. User presses the radio button.
2. User adjust the desired band from either AM or FM.
3. User adjusts the tuner to the desired radio station.
4. The radio is stopped when the Radio button is pressed again.

Extension:

1a. The Radio is activated by the alarm.
   1. The alarm activates and the radio is turned on playing the previously set station.
   2. The user presses the stop button.
   3. The radio stops playing.

1b. The Radio button is pressed while the (buzzing) Alarm is going off.
   1. The Alarm is disabled and the radio begins to play instead.
   2. The User presses the radio button again.
   3. The radio stops playing.

1c. The Radio button is pressed while the (radio) Alarm is going off.
   1. The Alarm feature is disabled and the radio begins to play as the radio feature.
   2. The User presses the radio button again.
   3. The radio stops playing.

Supplementary Specification

Revision History

Version	Date	Description	Author
Inception draft	January 22, 2011	First draft	Shannon Fox
Inception draft	January 29, 2011	Revision(1)	Shannon Fox

Introduction

This document is the repository of all requirements not captured in the use cases.

Functionality

Buzzer and Radio Function

- Clock can activate either a typical buzzer or radio at alarm time.
- Clock has two autonomous alarms that can be set to activate at different times.

Backup

A replaceable battery can be installed to keep internal time in the event of a power outage.

Usability

Readability

- LCD display attributes (the face of the unit):
  - Current Hour
  - Current Minute
  - AM/PM Indicator Light
  - Alarm 1 and Alarm 2 Indicator Lights
- Digital numbers for the clock should be visible from within 5 feet.
- Backlight can be adjusted from bright to dim.
- Radio face has a range of numbers and an indicator needle that can be easily read within 2 feet.

User Controls

Alarm/Radio Volume (along the right side of the unit)

- The volume for the alarm and the radio function is controlled by a single knob.

Time/Alarm Setting (along the top of the unit)

- Clock Button – 1st press enters the user into the time setting mode, 2nd press exits and activates clock.
- Alarm Set Button (one per alarm) – 1st press enters the user into the alarm setting mode, 2nd press exits and saves the time.
- Hour Button – Pressed to cycle to desired hour
- Minute Button – Pressed to cycle to desired minute

Alarm Switch (2 three-way slide switches along the top of the unit per alarm)

- Alarm/Buzzer on – Alarm activates the buzzer
- Alarm/Radio on – Alarm activates the radio
- Alarm off – Deactivates the alarm feature

Alarm Snooze Button (along the top of the unit)

- Initiates a 10 minute pause before the alarm sounds again

Alarm Off Button (along the top of the unit)

- Turns off the alarm when it is sounding

Radio ON/OFF Button (along the top of the unit)

- Pressing the button allows the user to turn on or turn off the radio.

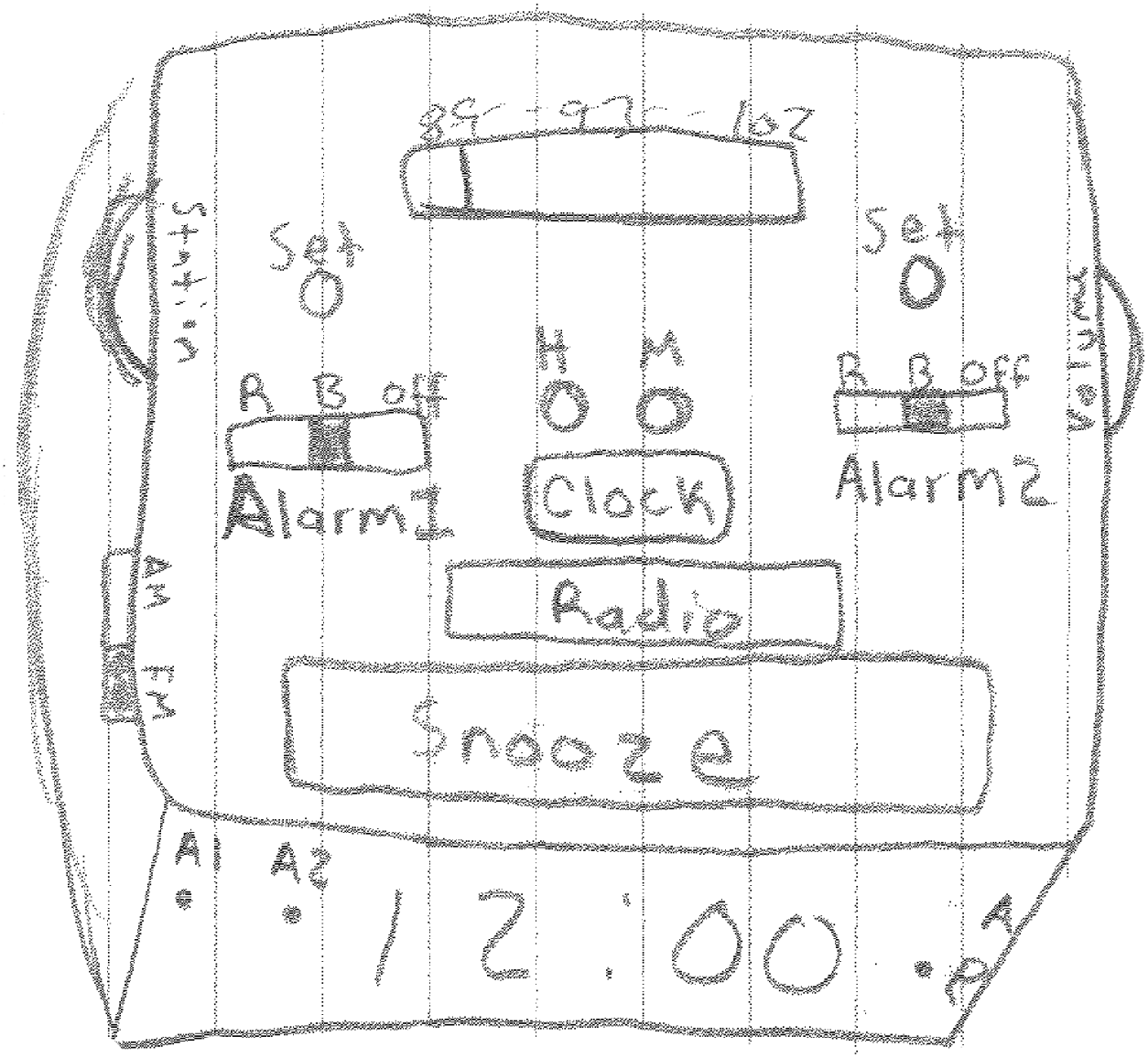
Radio adjustments

- Volume Knob (listed above)
- Tuning Dial (along the left side of the unit) – Different radio stations can be chosen by turning this knob.
- AM/FM switch (along the left side of the unit) – Activates the particular radio band the user chooses.

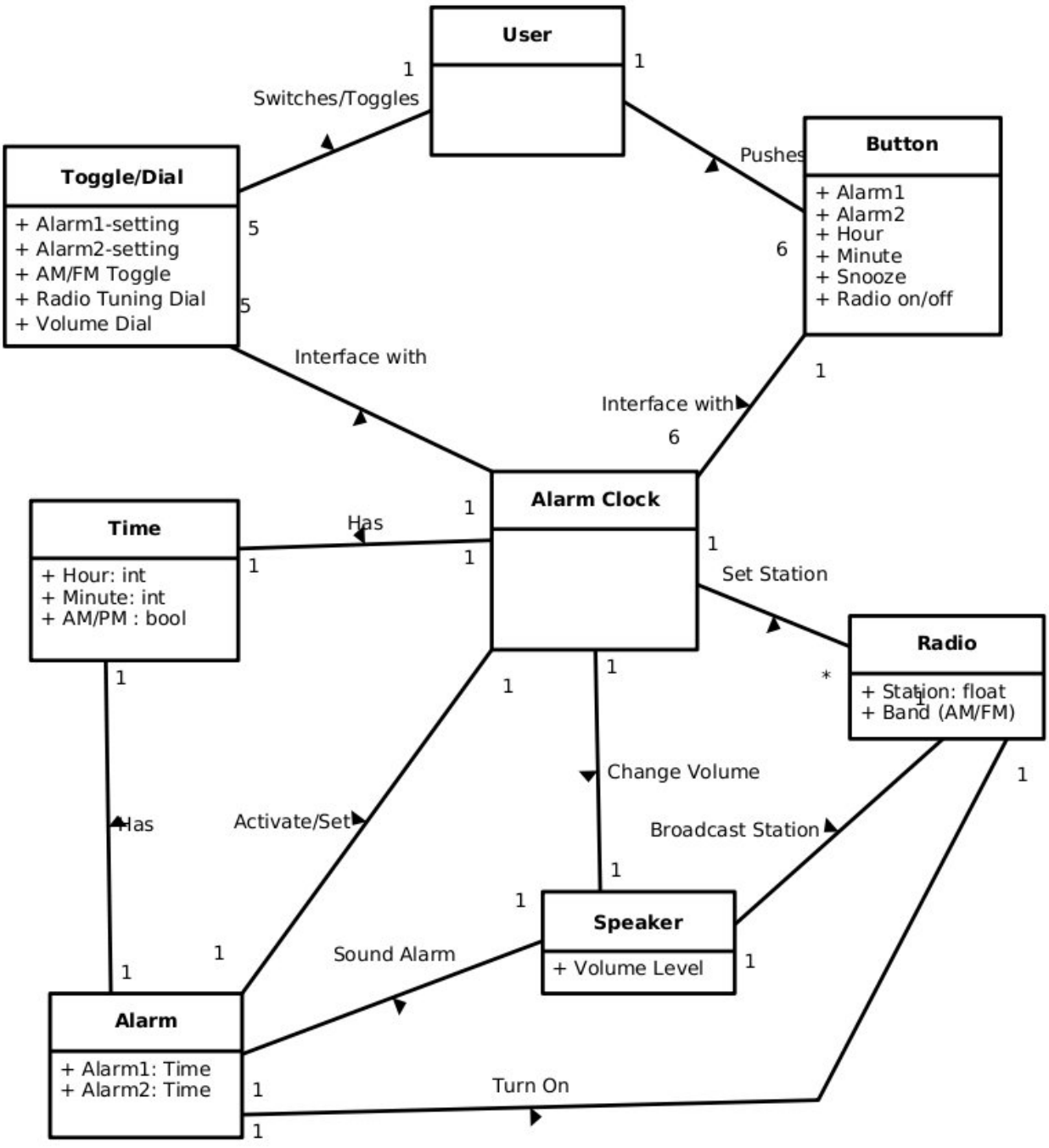
Reliability

As long as power is supplied (either AC or DC) to the unit, the time will remain correct in relation to the time that it was set. In addition (under the previous conditions) the alarm will consistently operate as set by the user.

Clock Sketch

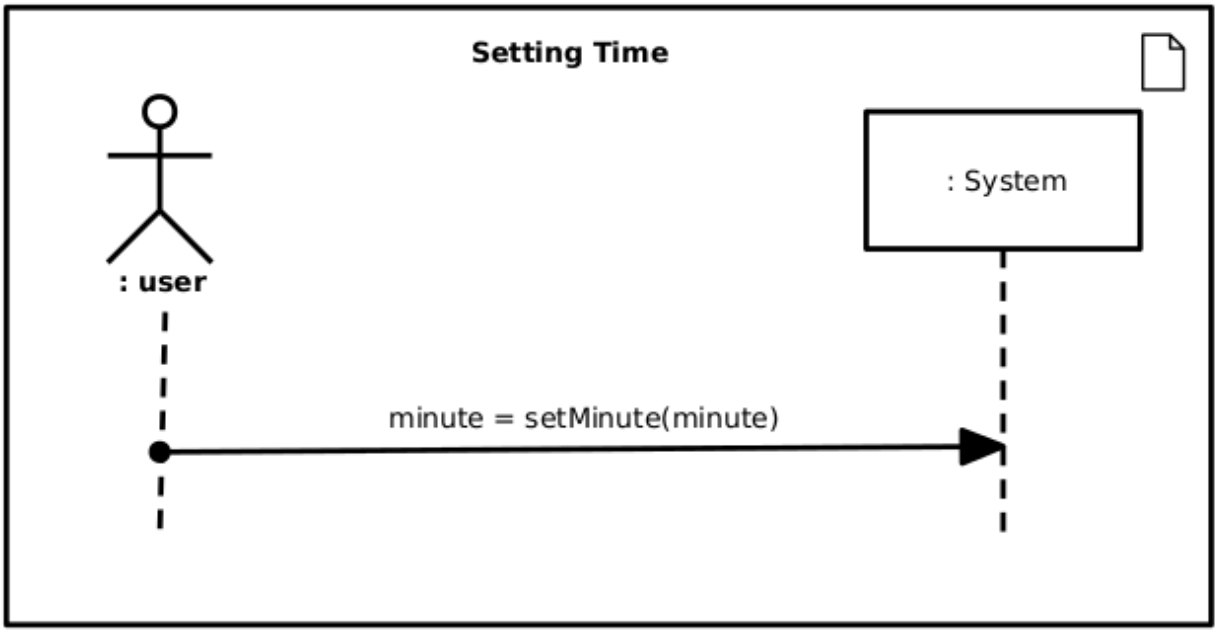


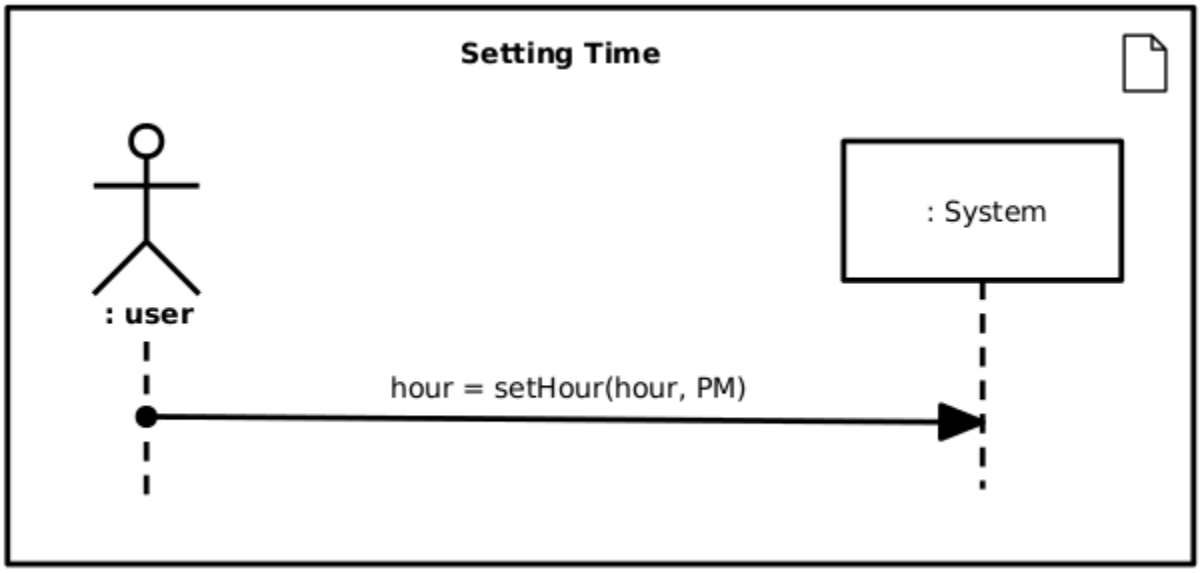
Domain Model

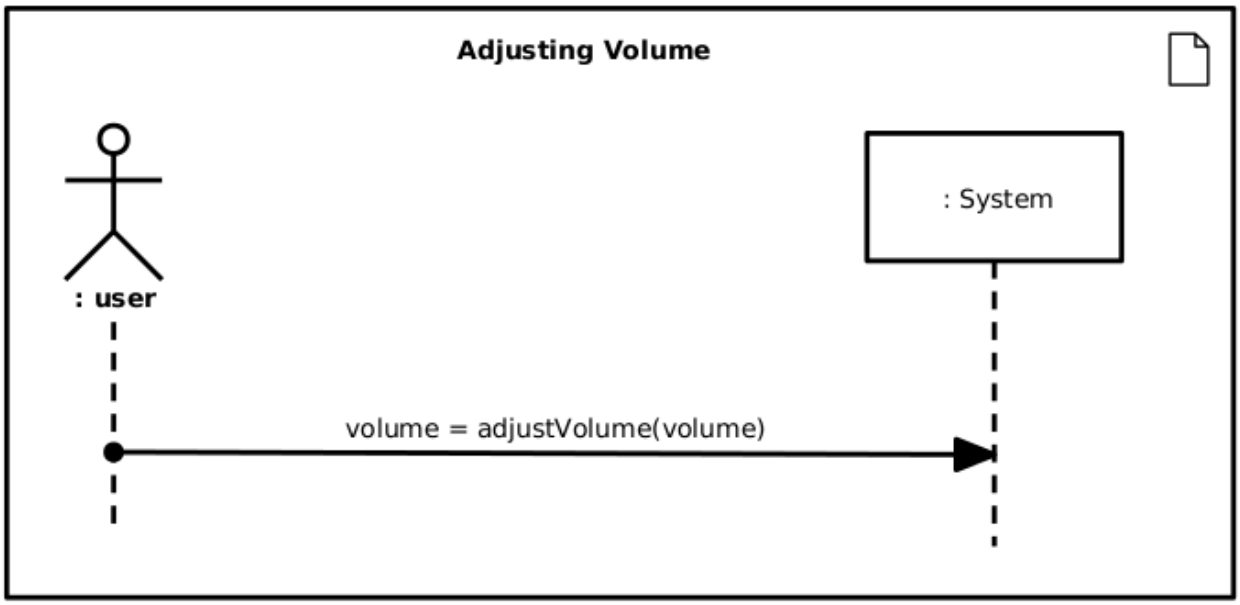


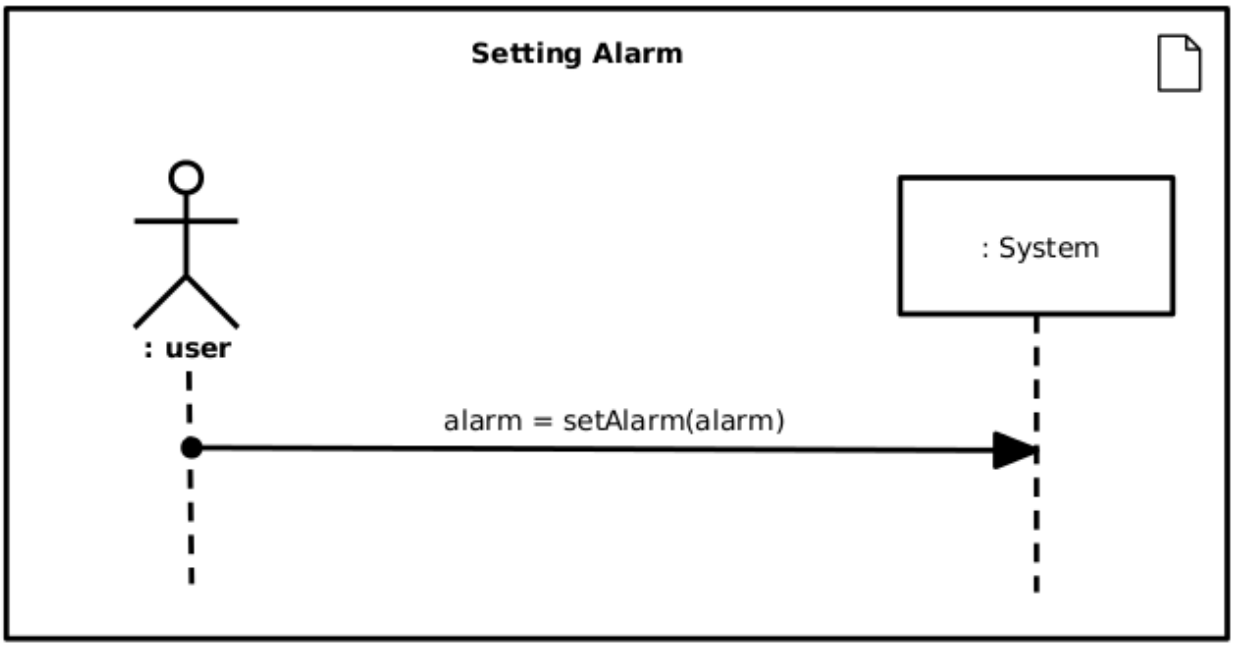
Note: All attributes in the diagram above with a plus-sign ('+') are meant to be private.

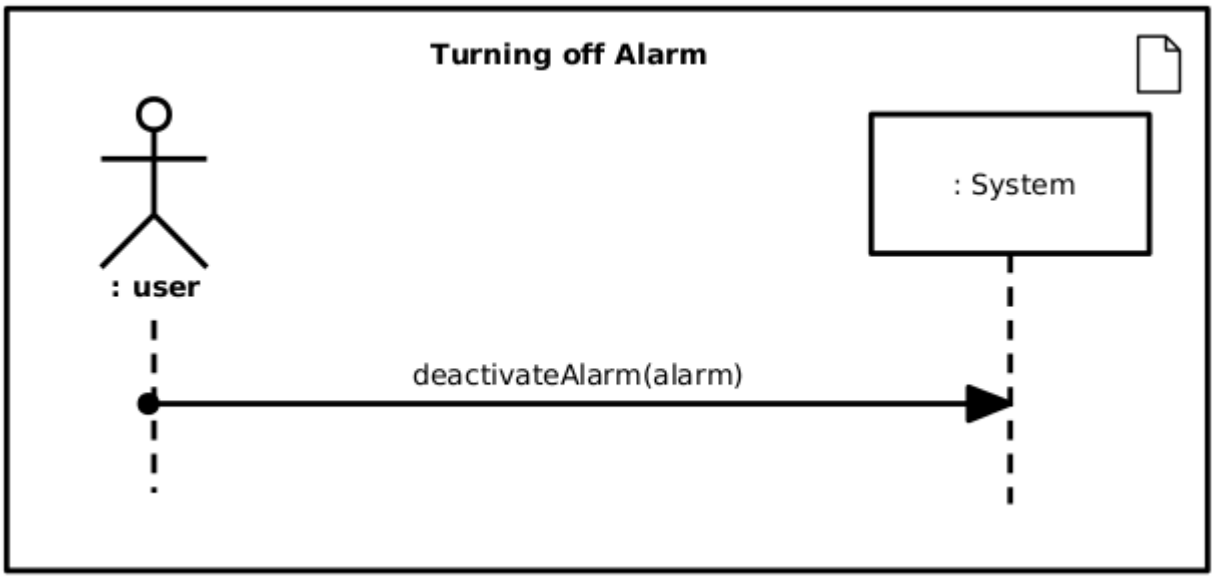
System Sequence Diagrams

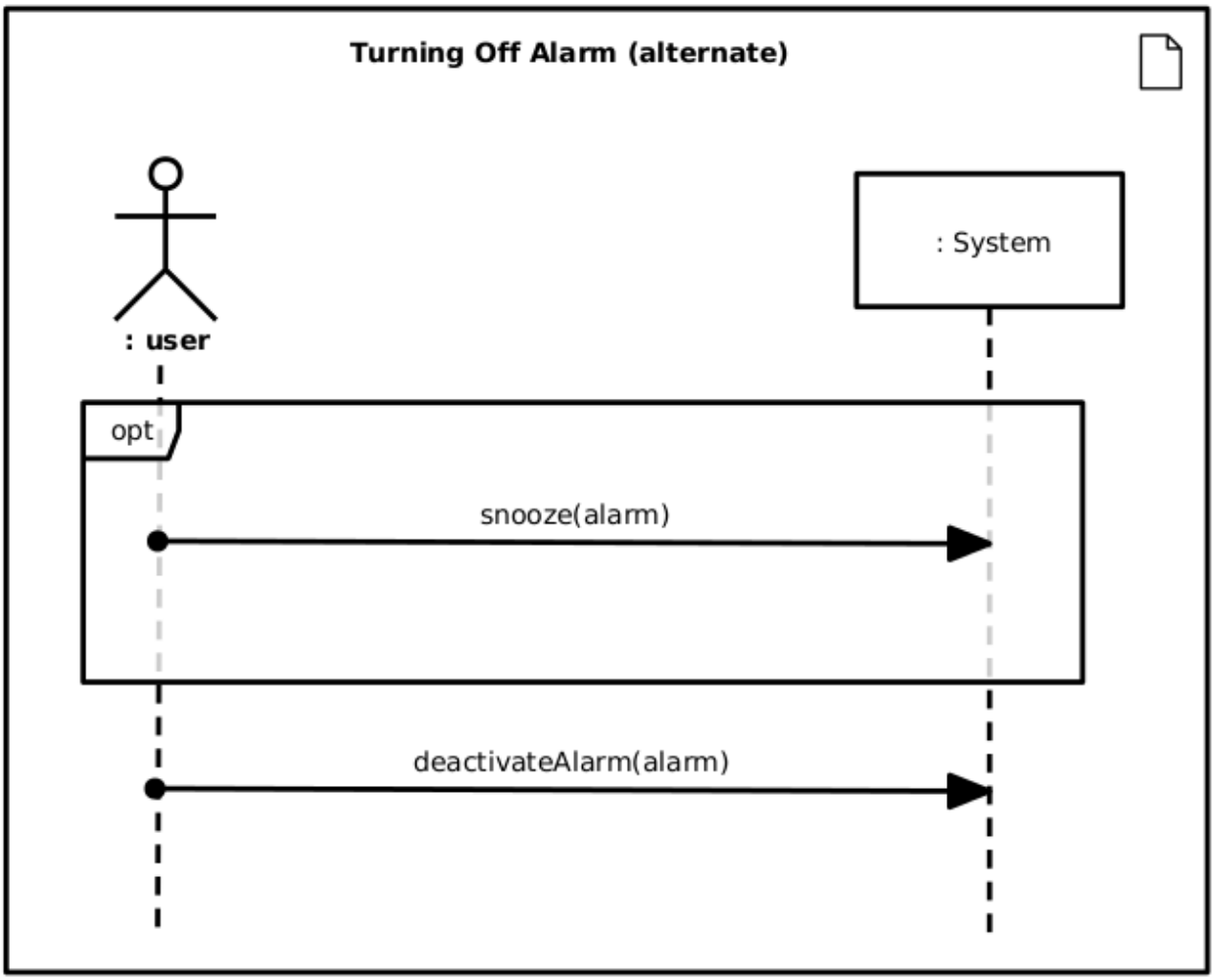


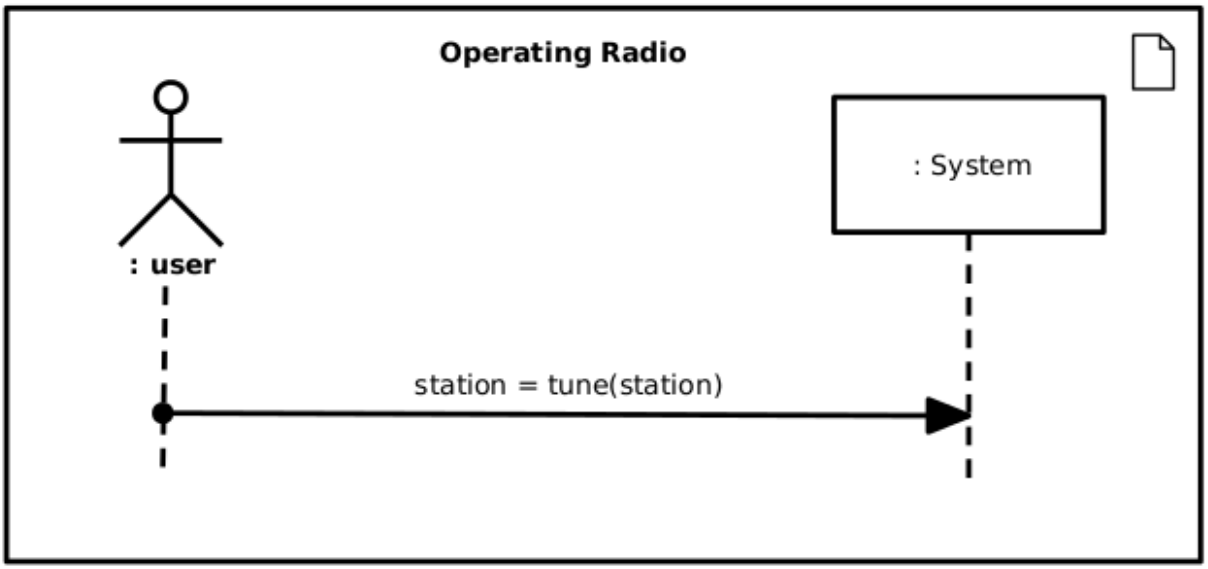


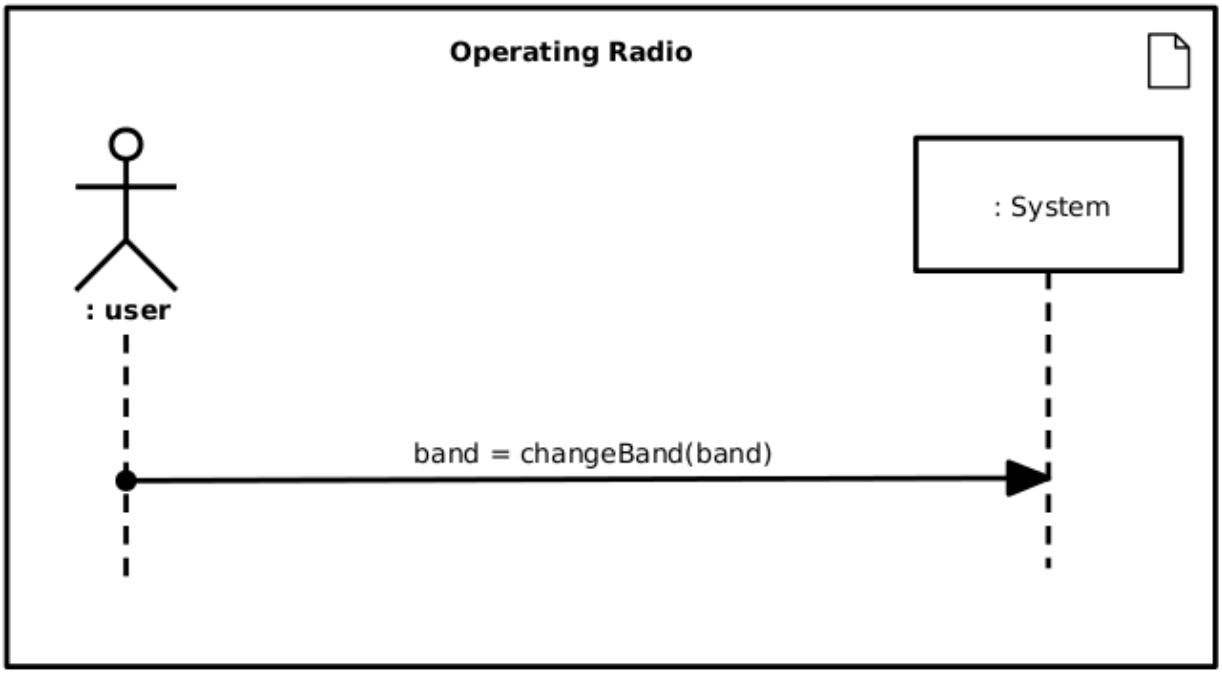

Adjusting Volume
: user
: System
volume = adjustVolume(volume)


Setting Alarm
: user
: System
alarm = setAlarm(alarm)


Turning off Alarm
: user
: System
deactivateAlarm(alarm)


Turning Off Alarm (alternate)
: user
: System
opt
snooze(alarm)
deactivateAlarm(alarm)


Operating Radio
: user
: System
station = tune(station)


Operating Radio
: user
: System
band = changeBand(band)

Operation Contracts

Operation:	setTime(int hour, int minute, bool AM)
Cross References:	Use Cases: UC1 Setting the Time
Preconditions:	The user wants to set the time with the specified parameters.
Postconditions:	The time is set.

Operation:	raiseVolume()
Cross References:	Use Case: UC4 Operating Radio, UC2 Setting Alarms
Preconditions:	- The user wants to raise the volume. - The volume is not at the maximum level.
Postconditions:	The volume level was incremented.

Operation:	lowerVolume()
Cross References:	Use Case: UC4 Operating Radio, UC2 Setting Alarms
Preconditions:	The user wants to lower the volume.
Postconditions:	The volume level was decremented.

Operation:	setAlarmTime(Alarm alarm, int hour, int minute, bool AM)
Cross References:	Use Case: UC2: Setting Alarms
Preconditions:	The user wants to set an alarm with the given parameters for Time.
Postconditions:	The alarm time was set.

Operation:	setAlarmTypeBuzzer(Alarm alarm)
Cross References:	Use Case: UC2: Setting Alarms
Preconditions:	The user wants to set an alarm to use the buzzer.
Postconditions:	The alarm's type was set to buzzer.

Operation:	setAlarmTypeRadio(Alarm alarm)
Cross References:	Use Case: UC2: Setting Alarms
Preconditions:	The user wants to set an alarm to use the radio.
Postconditions:	The alarm's type was set to radio.

Operation:	deactivateAlarm()
Cross References:	Use Case: UC3 Responding to Alarm
Preconditions:	- An alarm has sounded. - The user wants to quiet the alarm.
Postconditions:	The alarm was quieted.

Operation:	disableAlarm(Alarm alarm)
Cross References:	Use Case: UC3 Responding to Alarm
Preconditions:	The user wants to turn off an alarm.
Postconditions:	The alarm was turned off/disabled.

Operation:	snooze()
Cross References:	Use Case: UC3 Responding to Alarm
Preconditions:	- An alarm has sounded. - The user wants to sleep in.
Postconditions:	The alarm went into a temporary 10-minute delay.

Operation:	tuneUp()
Cross References:	Use Case: UC4 Operating the Radio
Preconditions:	- The radio is turned on. - The user wants to change the station to a higher station.
Postconditions:	The station was changed to a higher station.

Operation:	tuneDown()
Cross References:	Use Case: UC4 Operating the Radio
Preconditions:	- The radio is turned on. - The user wants to change the station to a lower station.
Postconditions:	The station was changed to a lower station.

Operation:	changeBand()
Cross References:	Use Case: UC4 Operating the Radio
Preconditions:	- The radio is turned on. - The user wants to change the band.
Postconditions:	The band was changed.

Class Diagrams

AlarmClock
- time : Time - alarm : Alarm [0..2] - radio : Radio - volume : Volume
+ setTime(int hour, int minute, bool AM) + setAlarmTime(Alarm alarm, int hour, int minute, bool AM) + snooze() + deactivateAlarm() + disableAlarm(Alarm alarm) + raiseVolume() + lowerVolume() + tuneUp() + tuneDown() + changeBand()

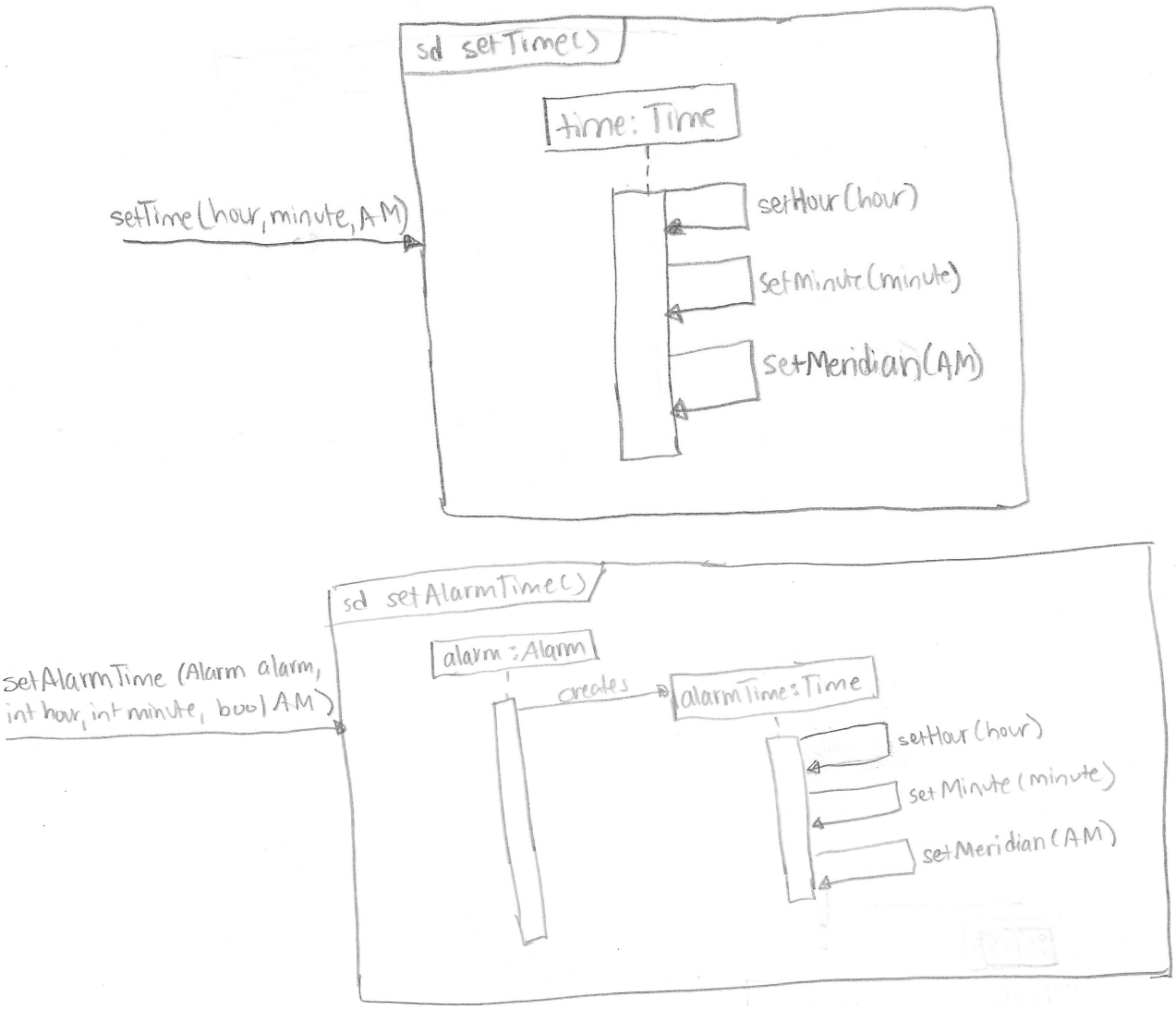
Volume
- max : int = 100 //final - min : int = 10 //final - level : int - increment : int = 10 //final
+ atMax() : bool + atMin() : bool + raiseVolume() + lowerVolume()

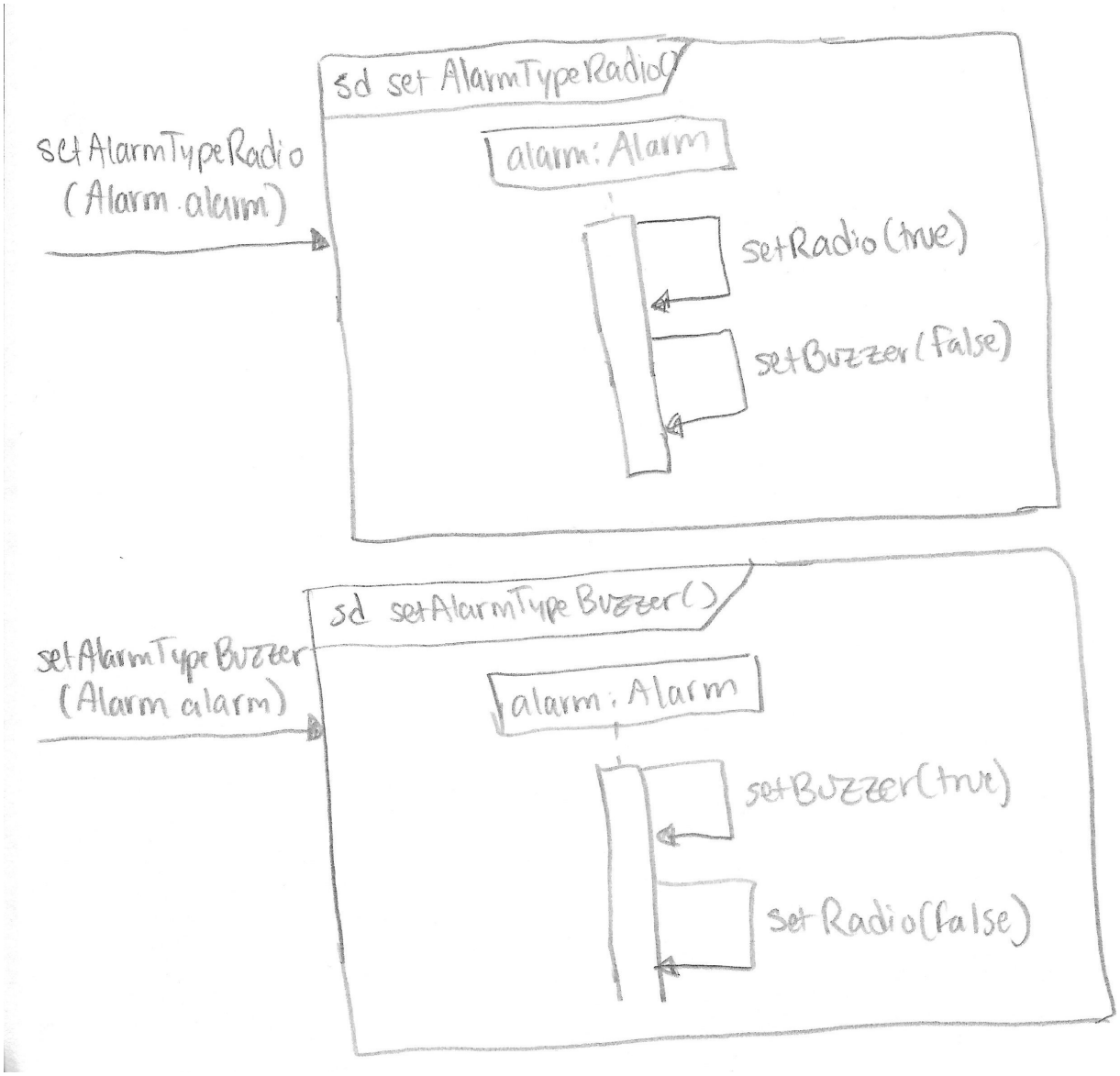
Time
- hour : int - minute : int - AM : bool //true=AM

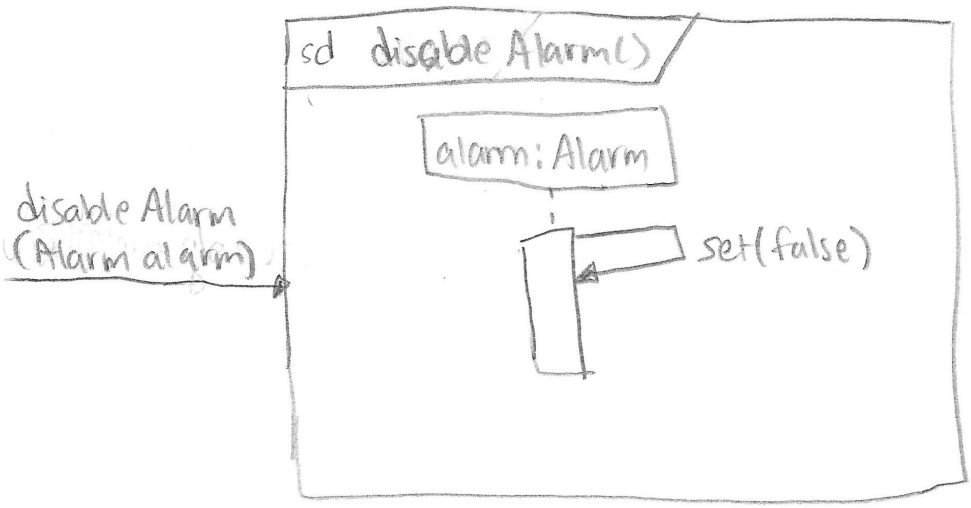
Radio
- amMax : float - amMin : float - fmMax : float - fmMin : float - station : float - playing : bool
+ tuneUp() + tuneDown() //loops around + changeBand()

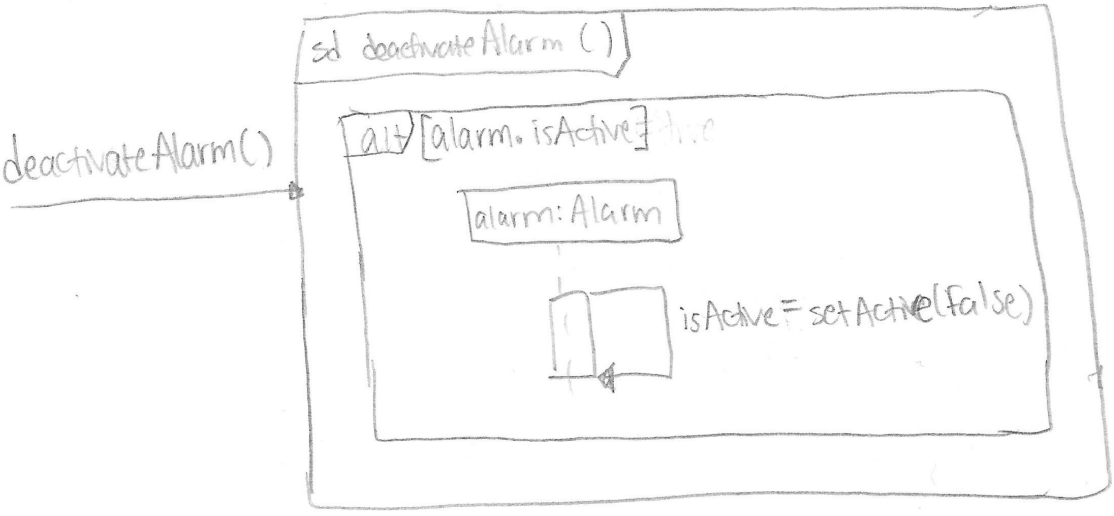
Alarm
- alarmTime : Time - isSet : bool - isActive: bool //true=sounding - radioSet : bool - buzzerSet : bool - tempTime : Time - snoozed : bool
+ set(bool) //on-off + snooze() + setRadio(bool) : bool + setBuzzer(bool) : bool

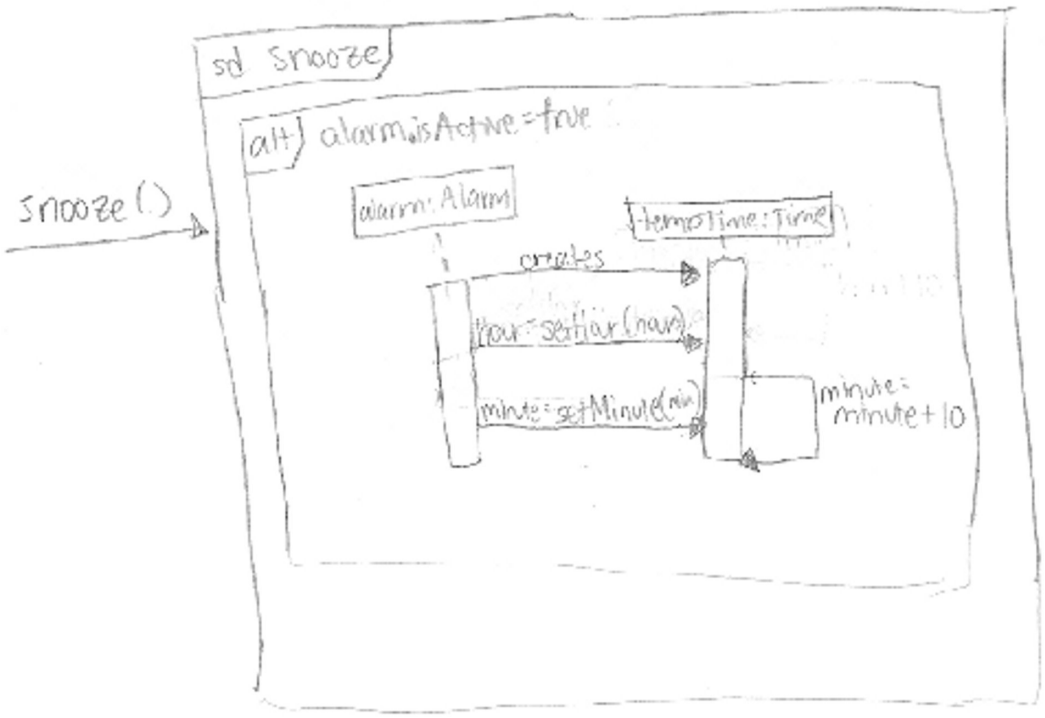
Sequence Diagrams

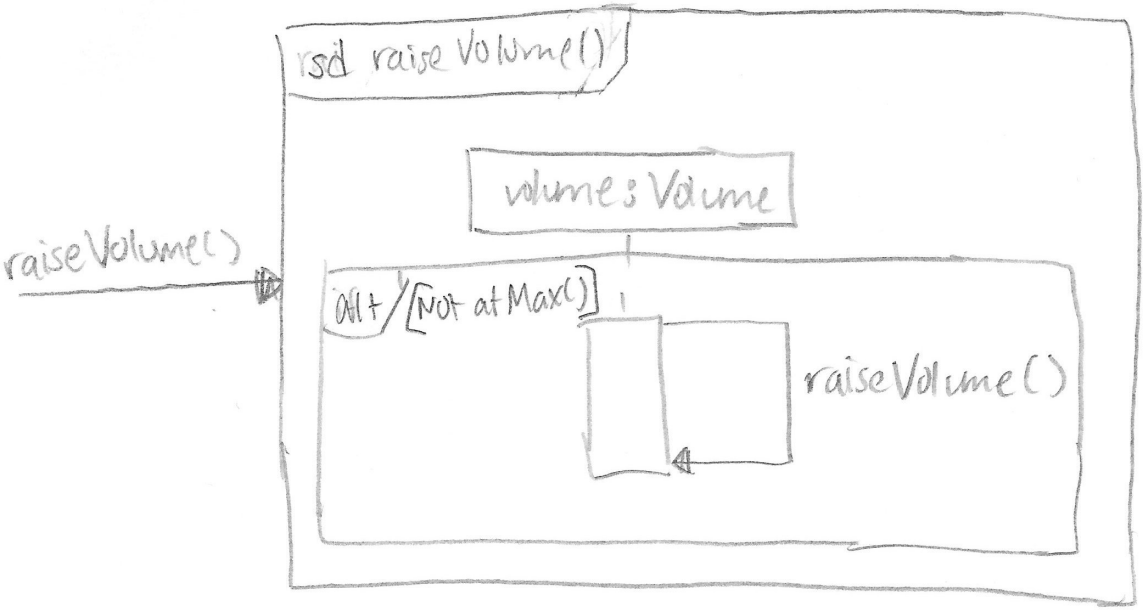


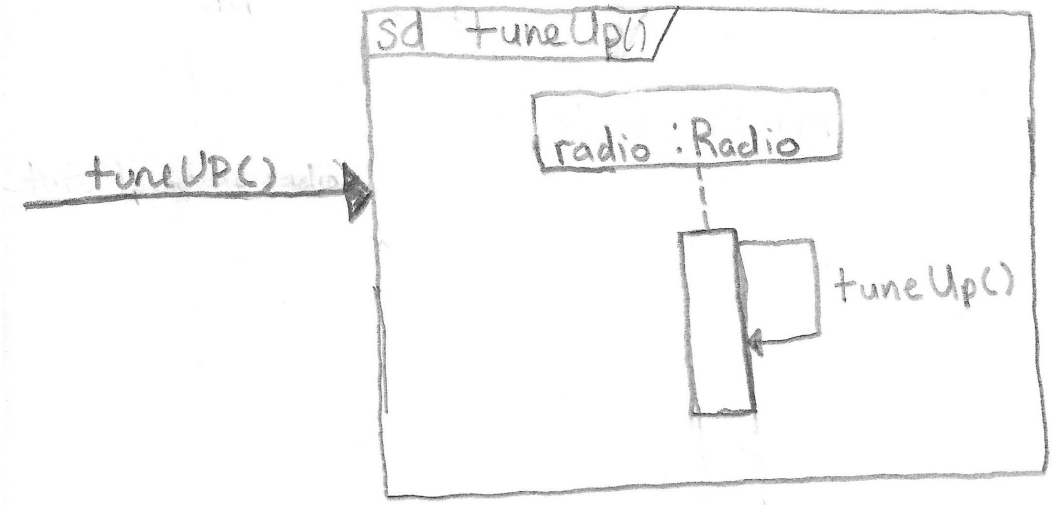

sd set AlarmTypeRadio()
setAlarmTypeRadio (Alarm alarm)
alarm: Alarm
setRadio(true)
setBuzzer(false)
sd setAlarmTypeBuzzer()
setAlarmTypeBuzzer (Alarm alarm)
alarm: Alarm
setBuzzer(true)
setRadio(false)

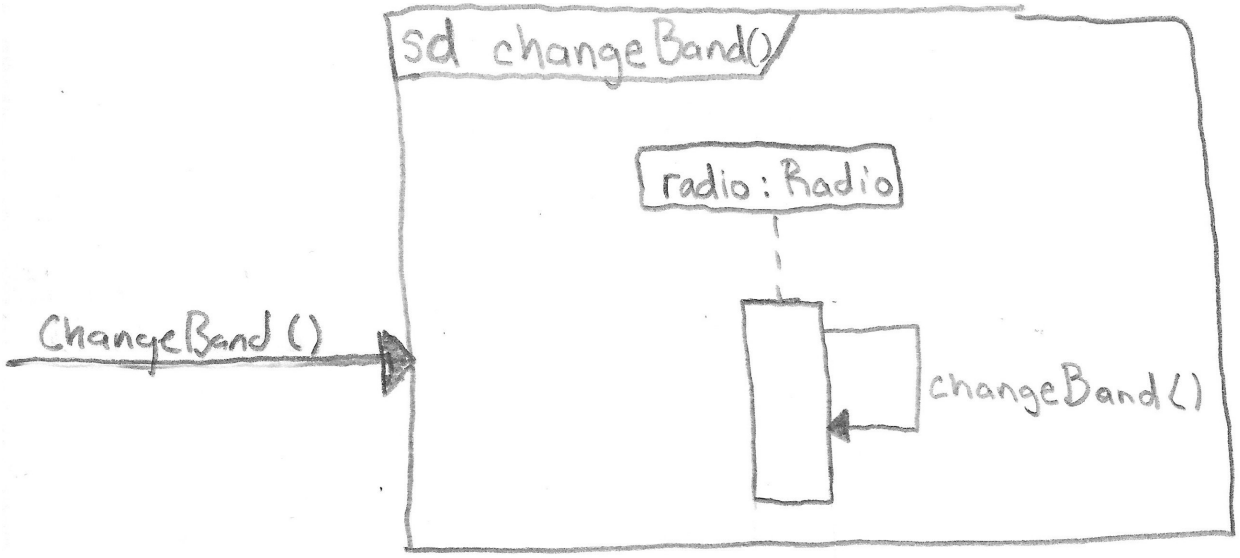

sd disableAlarm()
alarm: Alarm
disableAlarm
(Alarm alarm)
set(false)


sd deactivateAlarm()
deactivateAlarm()
alt
[alarm.isActive]
alarm: Alarm
isActive = setActive(false)


sd Snooze
alt alarmIsActive=true
snooze()
alarm:Alarm
tempTime:Time
creates
hour=setHour(hour)
minute=setMinute(min)
minute=
minute+10


sd raiseVolume()
volume:Volume
raiseVolume()
alt/[Not atMax()]
raiseVolume()


sd tuneUp()
radio : Radio
tuneUP()
tuneUp()


sd changeBand()
radio : Radio
ChangeBand()
changeBand()

Glossary

AC	Alternating Current, connected via power line to electricity source
adjustVolume (volume)	increases or decreases volume
Alarm	Time at which alarm is to go off; two alarms exist each with own settings
Alarm Indicator	Option for buzzer or radio for alarm sounding
AM	1. Abbreviation for amplitude modulation, varying the amplitude of a radio carrier wave 2. Abbreviation for ante meridiem, meaning before noon
Band	AM or FM setting for the radio
changeBand (band)	Changes the band from AM to FM or vice versa
DC	Direct Current, usually in the form of a battery
deactivateAlarm (alarm)	turns off the alarm until set at a later time
FM	Abbreviation for frequency modulation, varying the frequency of a radio carrier wave
LCD	Abbreviation for liquid crystal display
PM	Abbreviation for post meridiem, meaning after noon
setHour (hour, PM)	Accepts the users input and changes the hour and AM/PM indicator of the alarm or the current time
setMinute (minute)	Accepts the users input and changes the minute of the alarm or the current time
snooze (alarm)	To initiate an alarm to sound again later
soundAlarm (alarm)	causes the alarm to go off and make noise
station	Radio station
Time	Current hour and minutes, with AM/PM indicator
tune (station)	To change the station of the radio