Dual-Alarm Clock Radio

AJ, William, and Luke (DaLAW)

Our Vision (*other than get an A)*

- Create software to model the logic of a dual-alarm FM/AM Radio Clock.

- Learn the process of Software Development while completing this task.

The Process: Use Cases

UC 1: Using Alarms

Pre-Conditions: *The clock time is correctly set*

Success Guarantee: *The alarm goes off at the specified time*

UC 2: Listening to the radio

Pre-Conditions: *Radio can recieve transmissions and respond to software*

Success Guarantee: *The station is playing and user controlled*

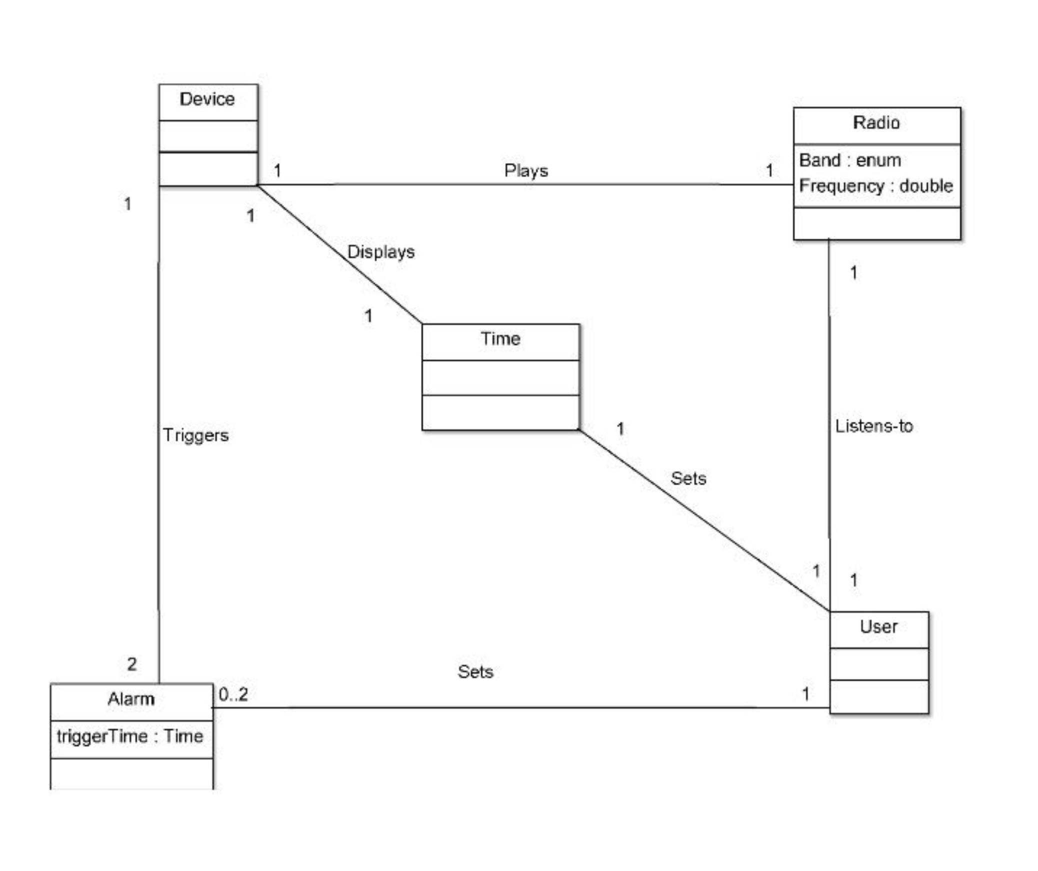
UC 3 : Setting Clock Time

Pre-Conditions: *There is already some time set*

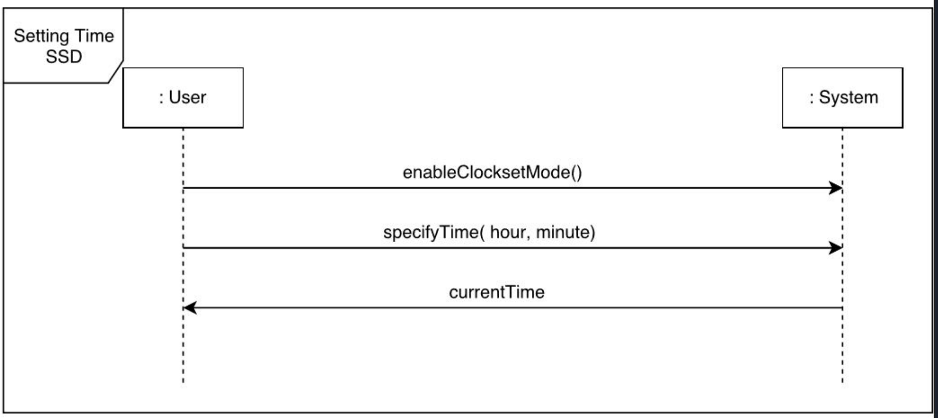
Success Guarantee *The clock time is changed*

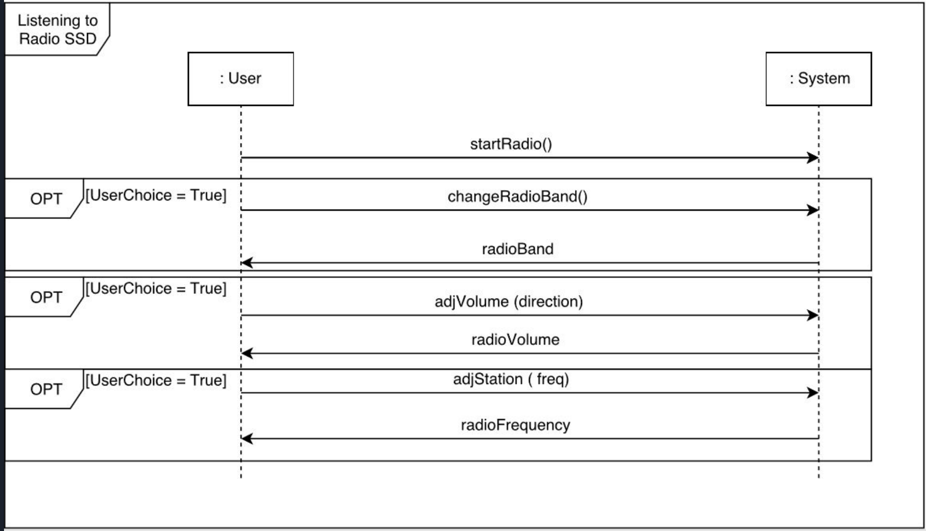
The Process: Modeling...

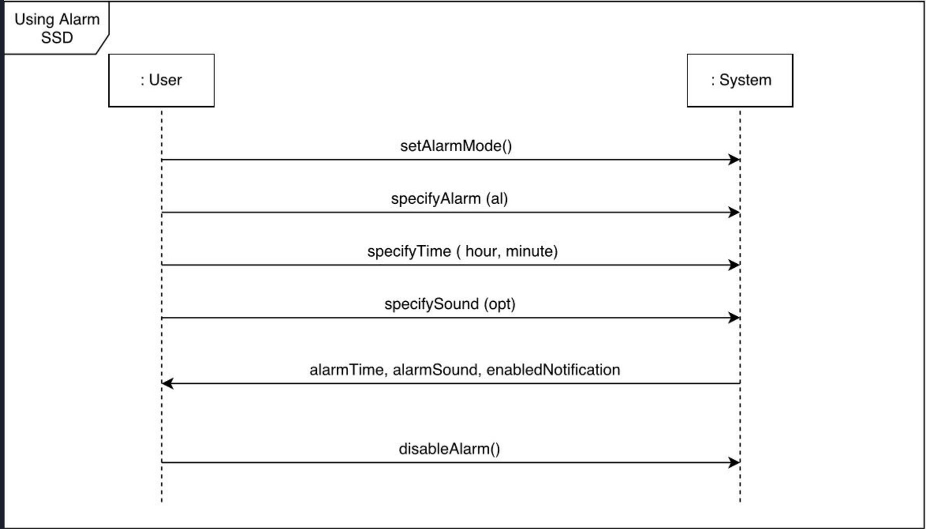
Domain Model



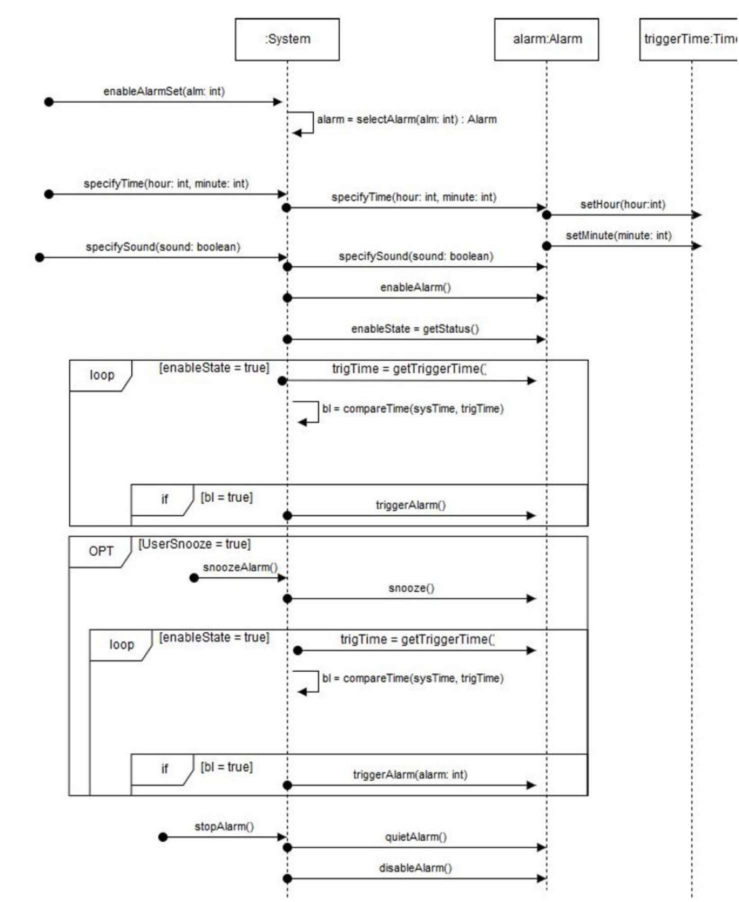
Sequence Diagram



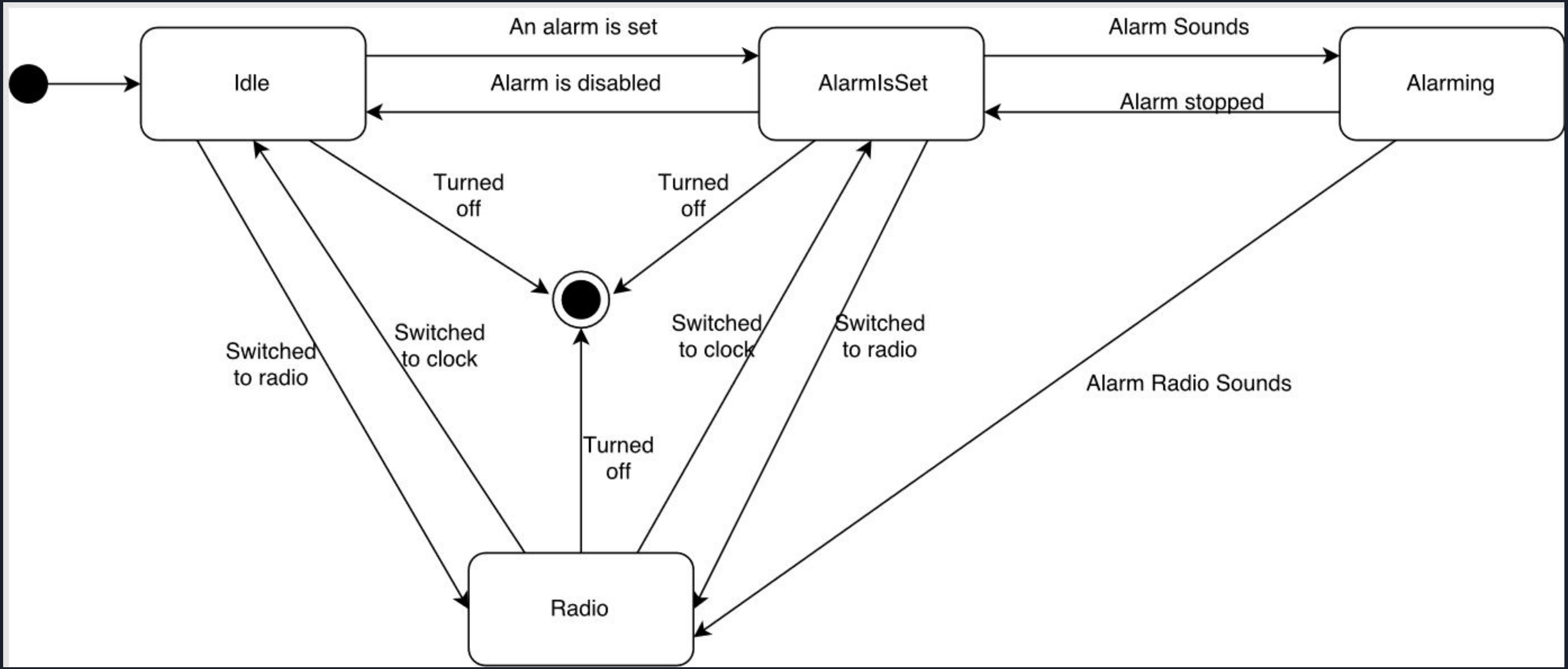




Use Case Realization: Dual-Alarm


:System
alarm:Alarm
triggerTime:Tim
enableAlarmSet(alm: int)
alarm = selectAlarm(alm: int) : Alarm
specifyTime(hour: int, minute: int)
specifyTime(hour: int, minute: int)
setHour(hour:int)
setMinute(minute: int)
specifySound(sound: boolean)
specifySound(sound: boolean)
enableAlarm()
enableState = getStatus()
loop
[enableState = true]
trigTime = getTriggerTime(
bl = compareTime(sysTime, trigTime)
if
[bl = true]
triggerAlarm()
OPT
[UserSnooze = true]
snoozeAlarm()
snooze()
loop
[enableState = true]
trigTime = getTriggerTime(
bl = compareTime(sysTime, trigTime)
if
[bl = true]
triggerAlarm(alarm: int)
stopAlarm()
quietAlarm()
disableAlarm()

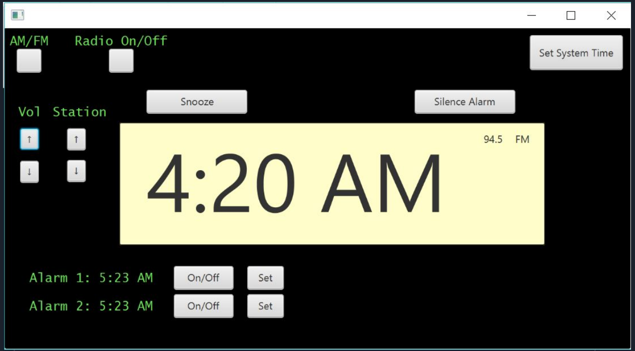
UI State Diagram



The Process: Implementation

- Coded in Java in the Netbeans IDE, using JavaFX for UI through Scenebuilder.
- GitHub for collaboration and code storage
- GroupMe for team communication

The Process: Presenting Implementation

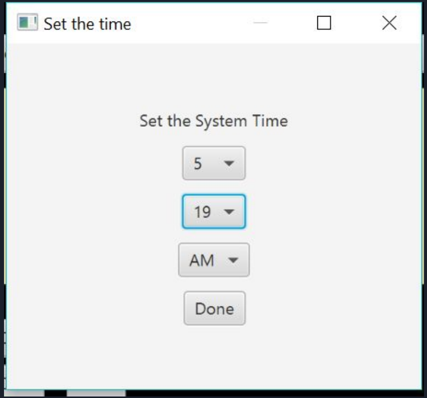


94.5 FM

4:20 AM

Alarm 1: 5:23 AM

Alarm 2: 5:23 AM

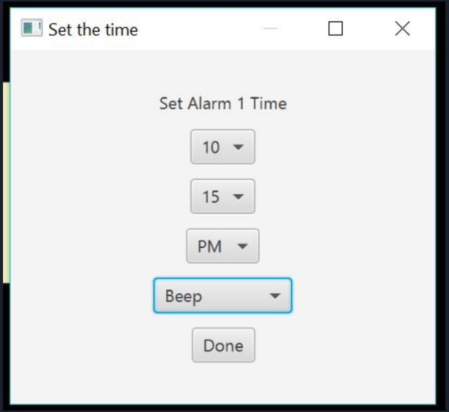


Set the System Time

5

19

AM



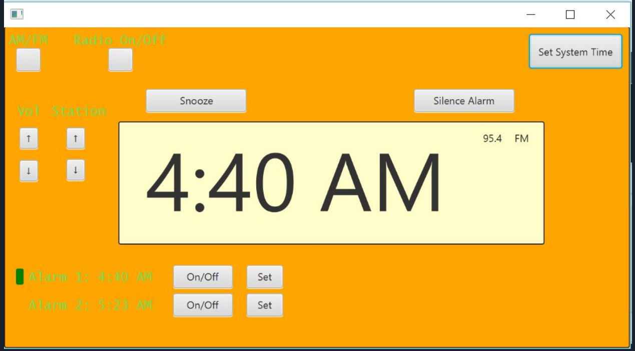
Set Alarm 1 Time

10

15

PM

Beep



95.4 FM

4:40 AM

Alarm 1: 4:40 AM

Alarm 2: 5:23 AM

Post-Process: Reflection on Difficulties

- Prior planning was inadequate for actual implementation

- Working with unique classes and their behaviours and limitations

Post Process: Lessons Learned

- Models are helpful when transitioning from ideas to code

- There will be deviations from the models

- Models should not take up a large amount of time