Team Malaga

● ● ●

Eric Hofesmann, Marge Marshall, Bryce Charydczak

Design Plan

- Iterative development.
- Github Deployment.
- Illustrations:
  - System Sequence Diagrams
  - Object Sequence Diagrams
  - State Machine Models
- Test Driven Developments
- Final Implementation

End User Product/Goals

Product: Alarm Clock featuring radio transpondence and dual alarm capabilities.
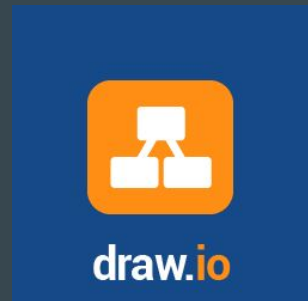
Goal:

- Be able to wake the user at user-specified times
- Give user the freedom to navigate the radio
- Operational and user friendly GUI to display the project

Instruments of Use

- NetBeans IDE 8.2
- JUnit 2.0
- Draw.IO
- JavaFx SceneBuilder
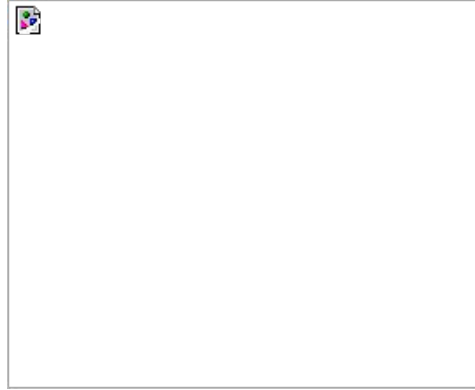- GitHub

Proprietary Aspects

- Increment/Decrement buttons for all input to dissuade any user error.
  - Increment By 10
- Color scheme
- Radio has seek functionality to find clear radio stations
- UITimingInterface and ControllerInterface
- This one goes up to 11
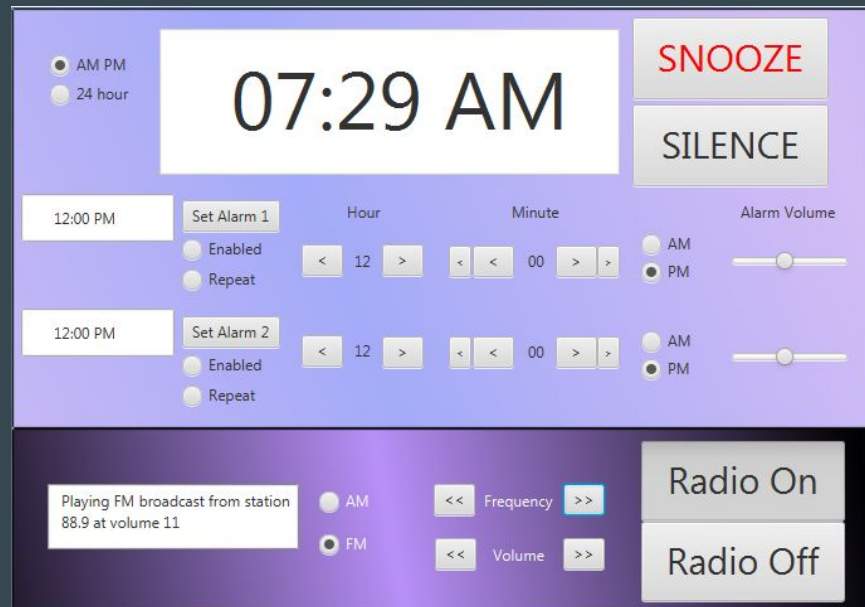

```
public interface UITimingInterface {

    public void updateMilitaryTime(int hours, int minutes);
    public void updateAMPMTime(int hours, int minutes, boolean isAM);
    public void soundAlarm(int alarmNum);

}
```

```
public interface ControllerInterface {

    public void soundAlarm(Alarm a);
    public void updateTime(Calendar cal);

}
```

System Sequence Diagrams
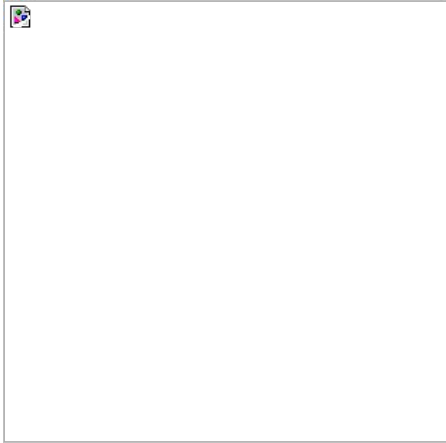
Original

Final

Object Sequence Diagram

Class Diagram


```
+ snoozeAlarm():
+ radioOn():
+ radioOff():
+ radioVolumeUp():
+ radioVolumeDown():
+ switchRadioReceiver():
+ incrementStation():
+ decrementStation():
+ playBroadcast():
+ updateTime(cal):
```
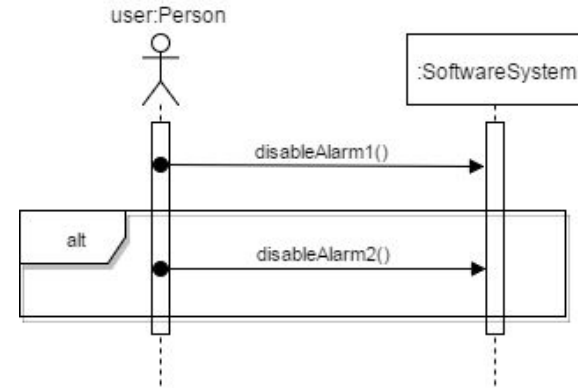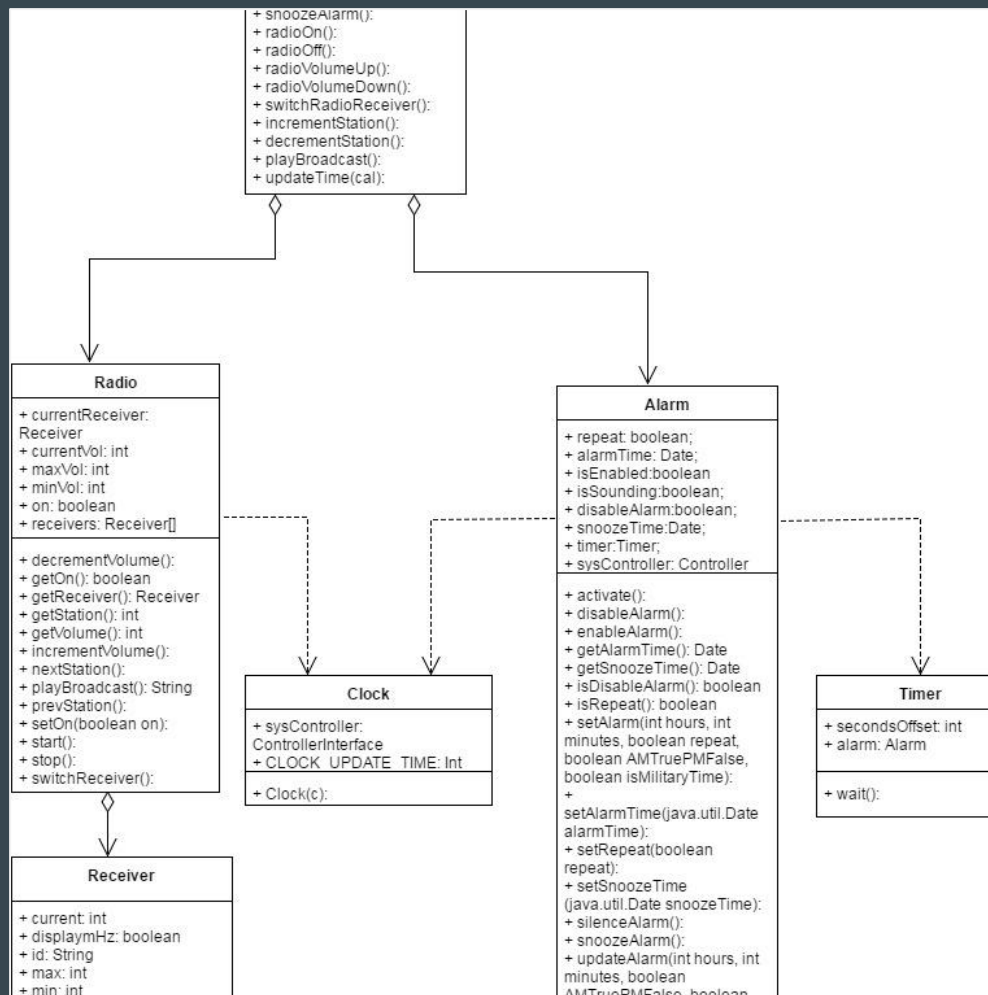
Radio

```
+ currentReceiver:
Receiver
+ currentVol: int
+ maxVol: int
+ minVol: int
+ on: boolean
+ receivers: Receiver[]
```

```
+ decrementVolume():
+ getOn(): boolean
+ getReceiver(): Receiver
+ getStation(): int
+ getVolume(): int
+ incrementVolume():
+ nextStation():
+ playBroadcast(): String
+ prevStation():
+ setOn(boolean on):
+ start():
+ stop():
+ switchReceiver():
```

Receiver

```
+ current: int
+ displaymHz: boolean
+ id: String
+ max: int
+ min: int
```

Clock

```
+ sysController:
ControllerInterface
+ CLOCK_UPDATE_TIME: Int
```

```
+ Clock(c):
```

Alarm

```
+ repeat: boolean;
+ alarmTime: Date;
+ isEnabled:boolean
+ isSounding:boolean;
+ disableAlarm:boolean;
+ snoozeTime:Date;
+ timer:Timer;
+ sysController: Controller
```

```
+ activate():
+ disableAlarm():
+ enableAlarm():
+ getAlarmTime(): Date
+ getSnoozeTime(): Date
+ isDisableAlarm(): boolean
+ isRepeat(): boolean
+ setAlarm(int hours, int
minutes, boolean repeat,
boolean AMTruePMFalse,
boolean isMilitaryTime):
+
setAlarmTime(java.util.Date
alarmTime):
+ setRepeat(boolean
repeat):
+ setSnoozeTime
(java.util.Date snoozeTime):
+ silenceAlarm():
+ snoozeAlarm():
+ updateAlarm(int hours, int
minutes, boolean
AMTruePMFalse, boolean
```

Timer

```
+ secondsOffset: int
+ alarm: Alarm
```

```
+ wait():
```

Triumphs,

- Collaborative development and efficiency
- Punctuality
- Git

Pitfalls,

- No sound
- Java FXML
- Keeping up with document
- Test-Driven Development

Lessons Learned

- Test-Driven Development
- Not getting stuck on details
- GRASP
- Iterative development