ICS 372 Object Oriented Design and Implementation

Group Project #2, Iteration #1

Group #3 (Hanna Prussog, Laura Anderson, Kevin Turgeon, Tom Carney)

Fridge/Freezer behavior:

- The Fridge portion works to maintain a temperature between 37 and 41 degrees Fahrenheit.
- The Freezer portion works to maintain a temperature between -9 and 0 degrees Fahrenheit.
- Fridge and Freezer have independent physical mechanisms (compressor, thermostat, etc.) for maintaining their respective temperatures.
- A user can set the desired temperature for either fridge or freezer to a value within their respective ranges.
- The temperature outside the collective fridge-freezer combination may range between 50 and 110 degrees Fahrenheit.
- One unit (fridge or freezer) may be "actively cooling" independent of the other.
- As long as a unit has reached its' desired temperature it will not be actively cooling.
- When not being actively cooled, the higher outside temperature causes the temperature inside each unit to slowly rise.
- A temperature 1 degree above a units set desired temperature will cause it to start actively cooling.
- Each unit has an internal light which turns on when the doors are opened.
- When the door is open, a unit's temperature rises more quickly than when it is closed.

Assumptions going forward:

- The "light" in each unit goes on when the door is opened and off when closed. For the sake of space, omitting repeatedly restating that detail going forward but is still part of the design.
- The design that follows assumes separate objects for fridge and freezer is a simpler approach vs. one single object with two parts.
- For simplicity, the temperature will be handled in whole number increments.

States:

Each unit is either active or idle, its' door either open or closed leading to four possible states.

- ElectricCooler (door closed) and idle.
- ElectricCooler (door closed) and active.
- ElectricCooler (door open) and idle.
- ElectricCooler (door open) and active.

Events:

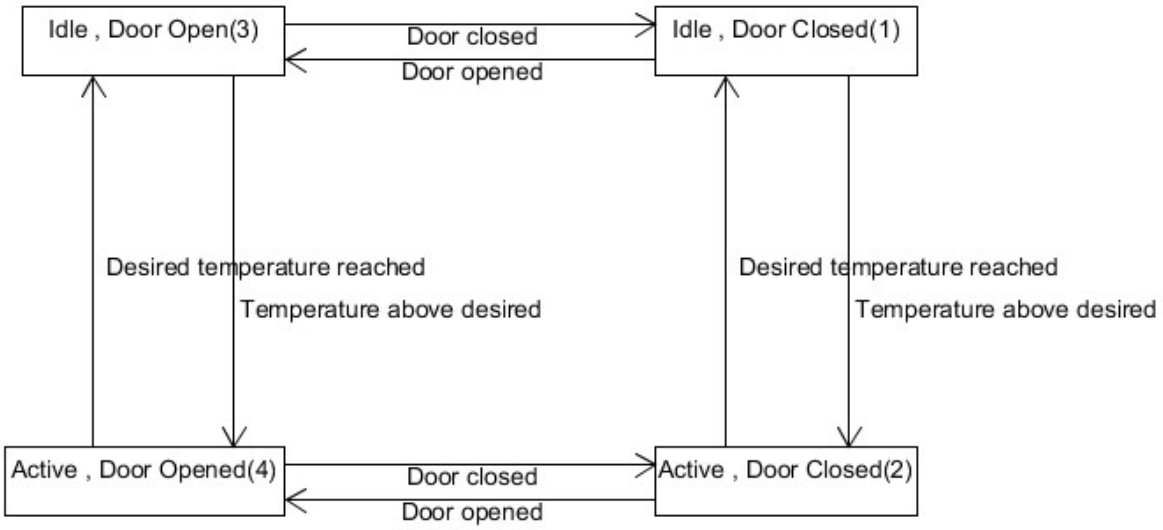
Things to keep in mind, change in door status, change in desired temp, temperature changes due to the passage of time (quickness varies based on door status)

- User opens the door.
- User closes the door.
- Actual temperature lowered to desired temperature (based on state and clock ticks).
- Actual temperature goes above desired temperature (based on state and clock ticks).

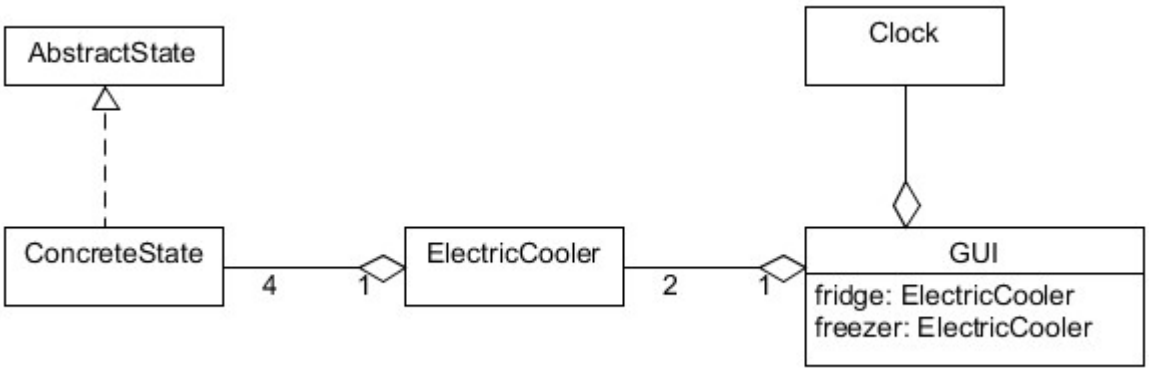
Transition Table: for an individual unit…

	Open door	Close door	Actual temp Lowers to desired	Actual Temp goes above desired
(1) Idle: closed	3	1	1	2
(2) Active: closed	4	2	1	2
(3) Idle: open	3	1	3	4
(4) Active: open	4	2	3	4

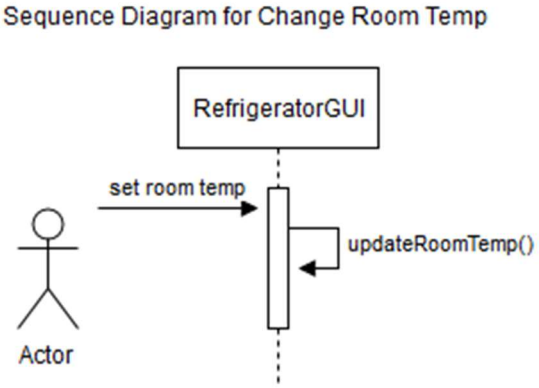
State Transition Diagram:



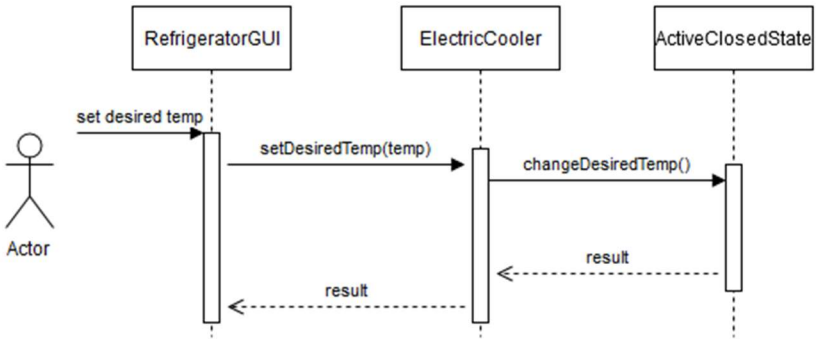
Rough Conceptual Diagram: Treating them as two separate units…



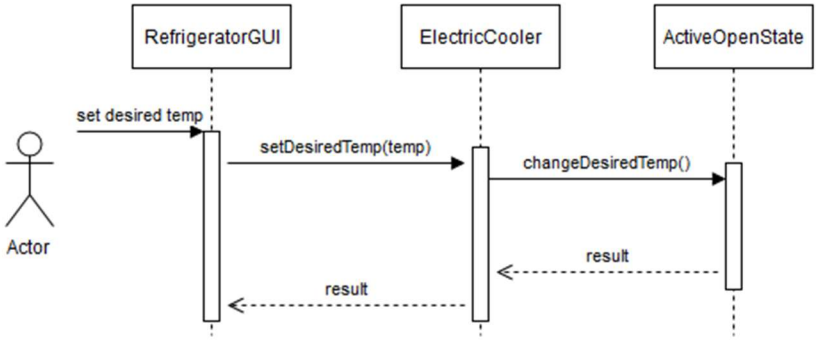
Sequence Diagrams:



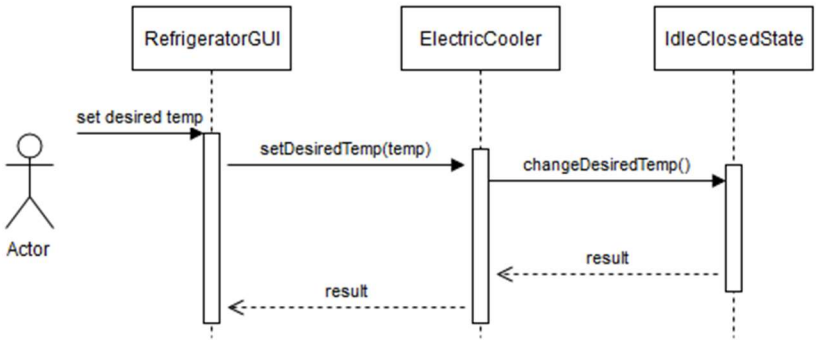
Sequence Diagram for Change Desired Temperature event in the ActiveClosedState



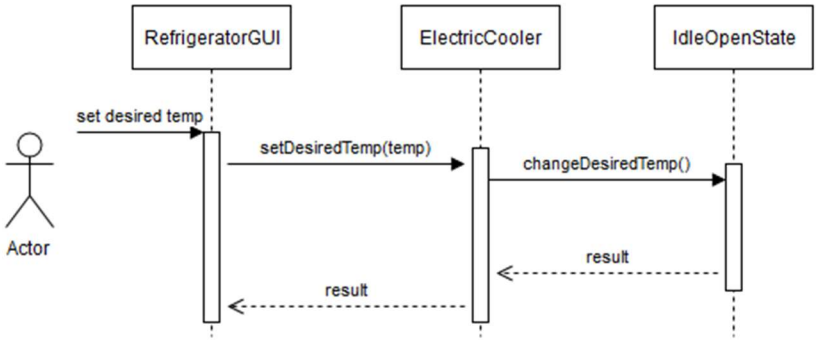
Sequence Diagram for Change Desired Temperature event in the ActiveOpenState



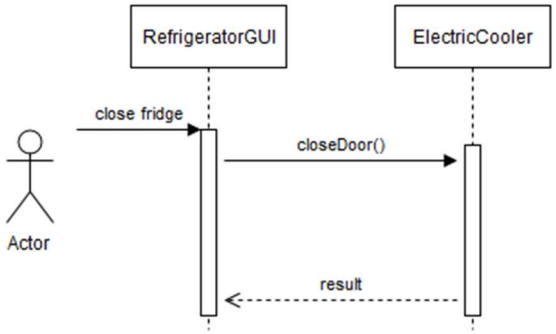
Sequence Diagram for Change Desired Temperature event in the IdleClosedState



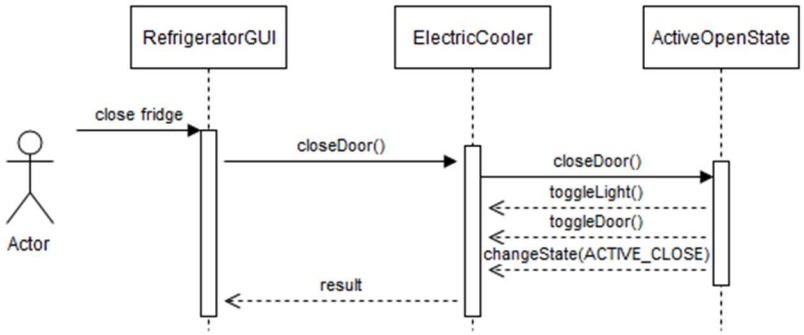
Sequence Diagram for Change Desired Temperature event in the IdleOpenState



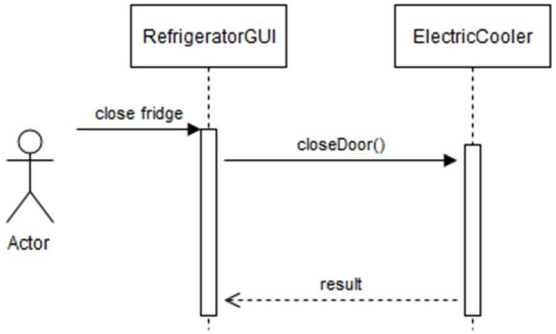
Sequence Diagram for Close Door event in the ActiveClosedState



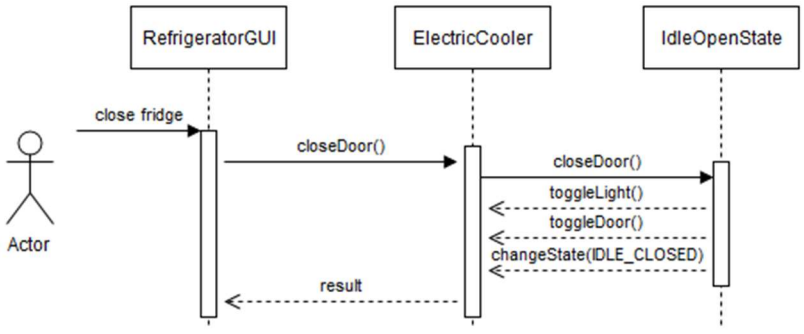
Sequence Diagram for Close Door event in the ActiveOpenState



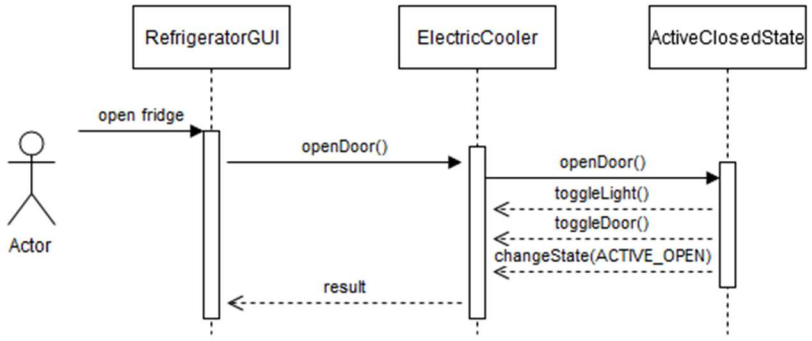
Sequence Diagram for Close Door event in the IdleClosedState



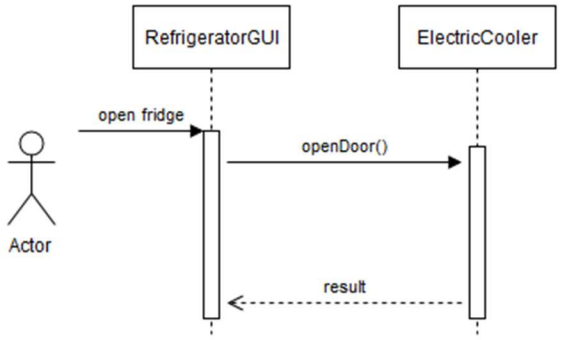
Sequence Diagram for Close Door event in the IdleOpenState



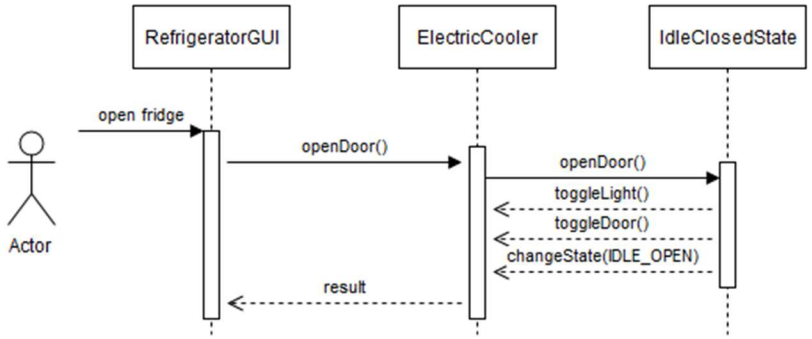
Sequence Diagram for Open Door event in the ActiveClosedState



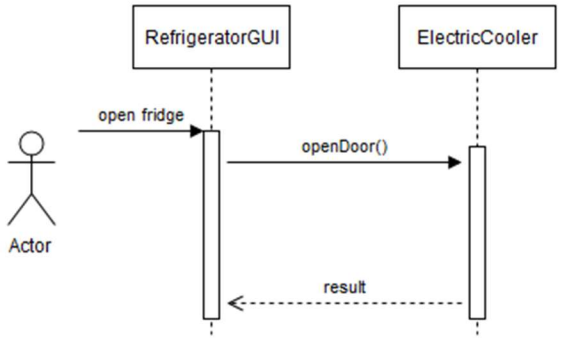
Sequence Diagram for Open Door event in the ActiveOpenState



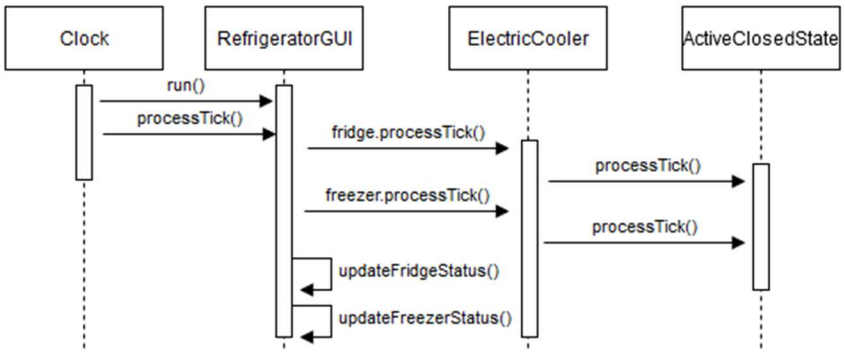
Sequence Diagram for Open Door event in the IdleClosedState



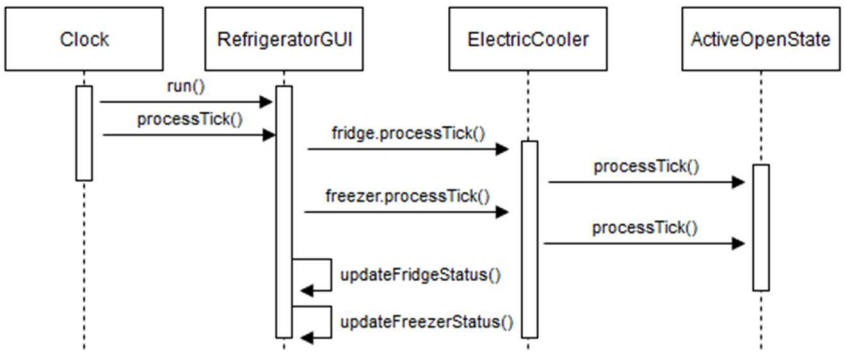
Sequence Diagram for Open Door event in the IdleOpenState



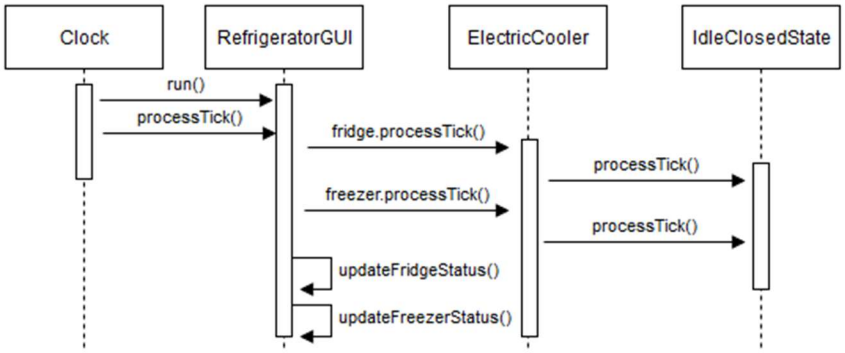
Sequence Diagram for Process Tick event in the ActiveClosedState



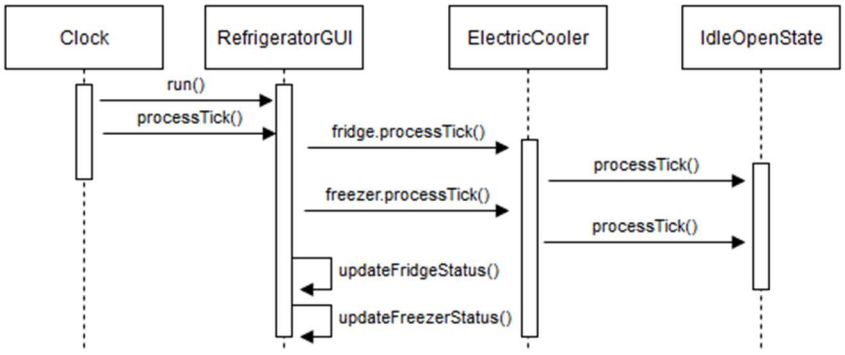
Sequence Diagram for Process Tick event in the ActiveOpenState

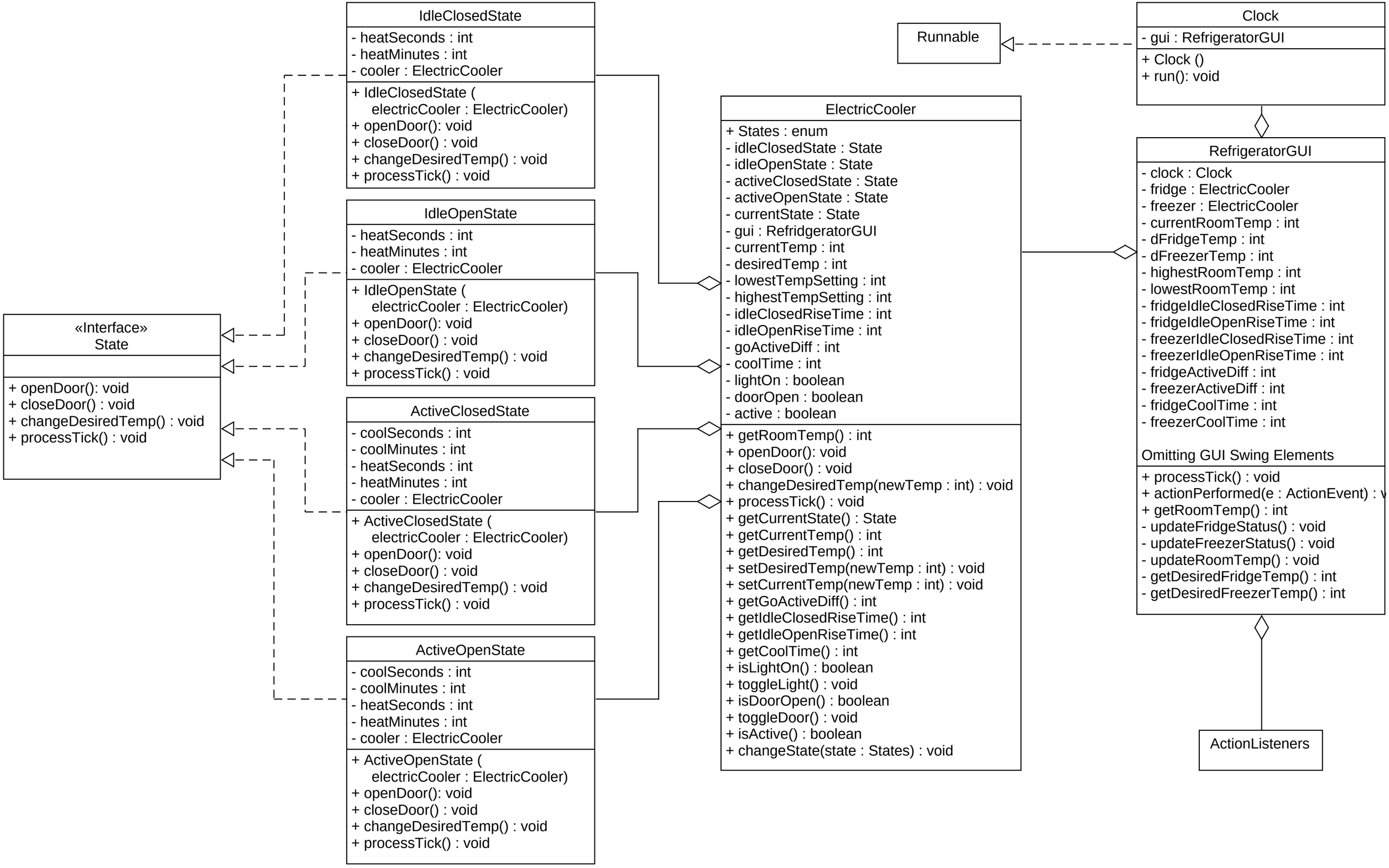


Sequence Diagram for Process Tick event in the IdleClosedState



Sequence Diagram for Process Tick event in the IdleOpenState




IdleClosedState
- heatSeconds : int
- heatMinutes : int
- cooler : ElectricCooler
+ IdleClosedState (
electricCooler : ElectricCooler)
+ openDoor(): void
+ closeDoor() : void
+ changeDesiredTemp() : void
+ processTick() : void
IdleOpenState
- heatSeconds : int
- heatMinutes : int
- cooler : ElectricCooler
+ IdleOpenState (
electricCooler : ElectricCooler)
+ openDoor(): void
+ closeDoor() : void
+ changeDesiredTemp() : void
+ processTick() : void
«Interface»
State
+ openDoor(): void
+ closeDoor() : void
+ changeDesiredTemp() : void
+ processTick() : void
ActiveClosedState
- coolSeconds : int
- coolMinutes : int
- heatSeconds : int
- heatMinutes : int
- cooler : ElectricCooler
+ ActiveClosedState (
electricCooler : ElectricCooler)
+ openDoor(): void
+ closeDoor() : void
+ changeDesiredTemp() : void
+ processTick() : void
ActiveOpenState
- coolSeconds : int
- coolMinutes : int
- heatSeconds : int
- heatMinutes : int
- cooler : ElectricCooler
+ ActiveOpenState (
electricCooler : ElectricCooler)
+ openDoor(): void
+ closeDoor() : void
+ changeDesiredTemp() : void
+ processTick() : void
ElectricCooler
+ States : enum
- idleClosedState : State
- idleOpenState : State
- activeClosedState : State
- activeOpenState : State
- currentState : State
- gui : RefridgeratorGUI
- currentTemp : int
- desiredTemp : int
- lowestTempSetting : int
- highestTempSetting : int
- idleClosedRiseTime : int
- idleOpenRiseTime : int
- goActiveDiff : int
- coolTime : int
- lightOn : boolean
- doorOpen : boolean
- active : boolean
+ getRoomTemp() : int
+ openDoor(): void
+ closeDoor() : void
+ changeDesiredTemp(newTemp : int) : void
+ processTick() : void
+ getCurrentState() : State
+ getCurrentTemp() : int
+ getDesiredTemp() : int
+ setDesiredTemp(newTemp : int) : void
+ setCurrentTemp(newTemp : int) : void
+ getGoActiveDiff() : int
+ getIdleClosedRiseTime() : int
+ getIdleOpenRiseTime() : int
+ getCoolTime() : int
+ isLightOn() : boolean
+ toggleLight() : void
+ isDoorOpen() : boolean
+ toggleDoor() : void
+ isActive() : boolean
+ changeState(state : States) : void
Runnable
Clock
- gui : RefrigeratorGUI
+ Clock ()
+ run(): void
RefrigeratorGUI
- clock : Clock
- fridge : ElectricCooler
- freezer : ElectricCooler
- currentRoomTemp : int
- dFridgeTemp : int
- dFreezerTemp : int
- highestRoomTemp : int
- lowestRoomTemp : int
- fridgeIdleClosedRiseTime : int
- fridgeIdleOpenRiseTime : int
- freezerIdleClosedRiseTime : int
- freezerIdleOpenRiseTime : int
- fridgeActiveDiff : int
- freezerActiveDiff : int
- fridgeCoolTime : int
- freezerCoolTime : int
Omitting GUI Swing Elements
+ processTick() : void
+ actionPerformed(e : ActionEvent) : v
+ getRoomTemp() : int
- updateFridgeStatus() : void
- updateFreezerStatus() : void
- updateRoomTemp() : void
- getDesiredFridgeTemp() : int
- getDesiredFreezerTemp() : int
ActionListeners