Blaine Billings
Julian Smith
Tyler Newman

System Sequence Diagrams and Operation Contracts

Operation Contracts

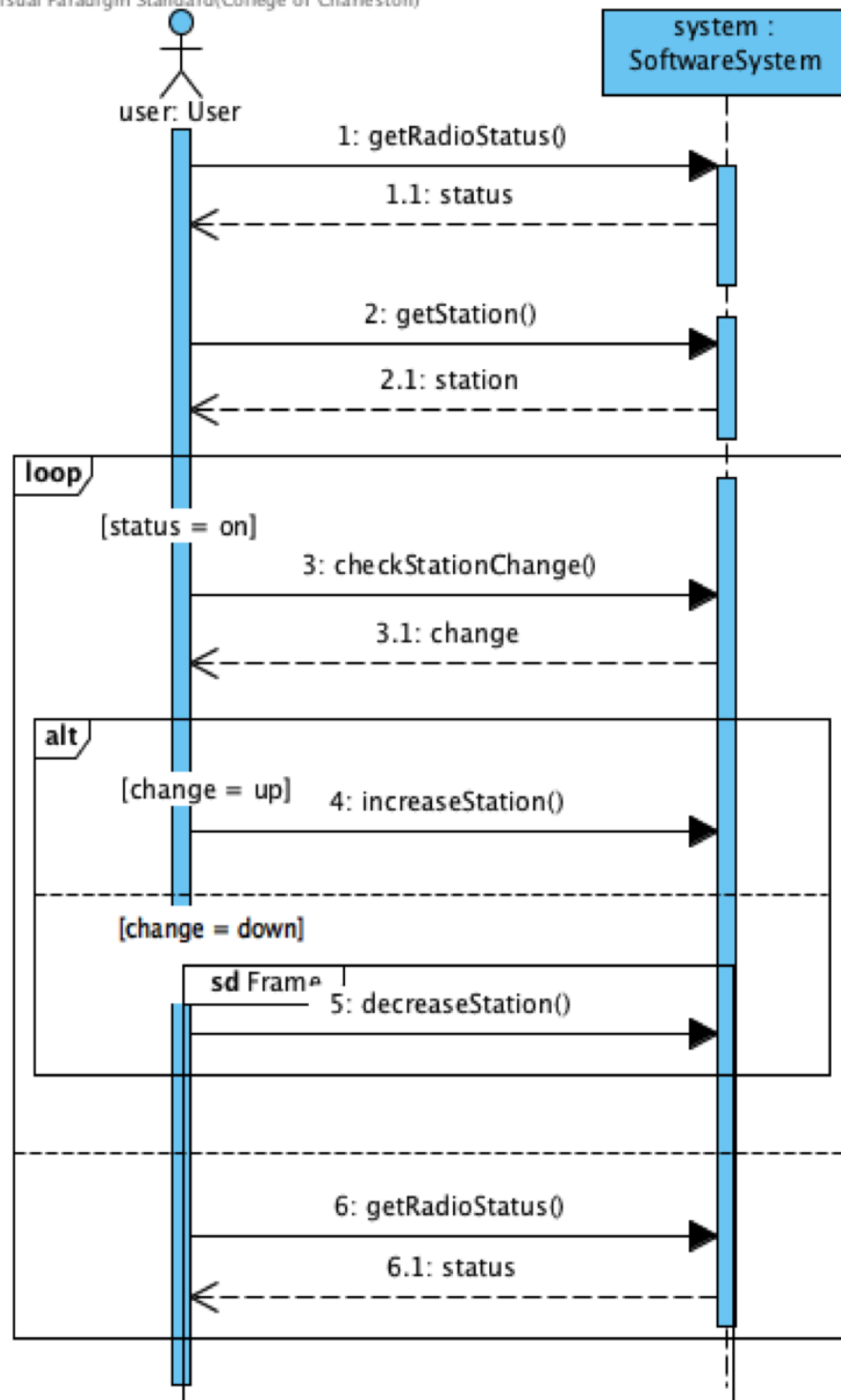I. Contract CO1
    A. Operation: getRadioStatus()
    B. Cross References: SSDs - Tune Radio
    C. Preconditions: None
    D. Postconditions:
         – Returns the current status of the radio (on/off)

II. Contract CO2
    A. Operation: getStation()
    B. Cross References: SSDs - Tune Radio
    C. Preconditions:
         – Radio must be set to a certain station (even if turned off)
    D. Postconditions:
         – Returns the station that the radio is currently set to

III. Contract CO3
    A. Operation: checkStationChange()
    B. Cross References: SSDs - Tune Radio
    C. Preconditions:
         – The radio must be turned on
    D. Postconditions:
         – The user has either chosen up, down, or nothing at all
         – This choice is returned

IV. Contract CO4
    A. Operation: increaseStation()
    B. Cross References: SSDs - Tune Radio
    C. Preconditions:
         – The radio must be turned on
    D. Postconditions:
         – The current station is increased by one interval

V. Contract CO5
    A. Operation: decreaseStation()
    B. Cross References: SSDs - Tune Radio
    C. Preconditions:
         – The radio must be turned on
    D. Postconditions:
         – The current station is decreased by one interval

VI.   Contract CO6
- A. Operation: switchTimeFormat()
- B. Cross References: SSDs - Change Time Format
- C. Preconditions:
  - The time is set to either 12 hour or 24 hour time
- D. Postconditions:
  - The time is switched to the opposite time format than in the precondition

VII.   Contract CO7
- A. Operation: enterAlarmSetMode(alarmNumber: integer)
- B. Cross References: SSDs - Set Alarm Time
- C. Preconditions: None
- D. PostConditions:
  - The alarm clock is in alarm-setting mode for the specified alarmNumber

VIII.   Contract CO8
- A. Operation: setAlarmTime(hours: integer, minutes: integer)
- B. Cross References: SSDs - Set Alarm Time
- C. Preconditions:
  - Alarm clock is in alarm-setting mode
- D. PostConditions:
  - The time of the previously specified alarm is changed to the inputted time

IX.   Contract CO9
- A. Operation: exitAlarmSetMode()
- B. Cross References: SSDs - Set Alarm Time
- C. Preconditions:
  - The alarm clock is in alarm-setting mode
- D. PostConditions:
  - The alarm clock is no longer in alarm-setting mode

X.   Contract CO10
- A. Operation: enterTimeSetMode()
- B. Cross References: SSDs - Set Time
- C. Preconditions: None
- D. PostConditions:
  - The alarm clock is in time-setting mode

XI.   Contract CO11
- A. Operation: setClockTime(hours: integer, minutes: integer)
- B. Cross References: SSDs - Set Time
- C. Preconditions:
  - The alarm clock is in time-setting mode
- D. Postconditions:
  - The time displayed on the alarm clock is changed to and saved as the inputted time
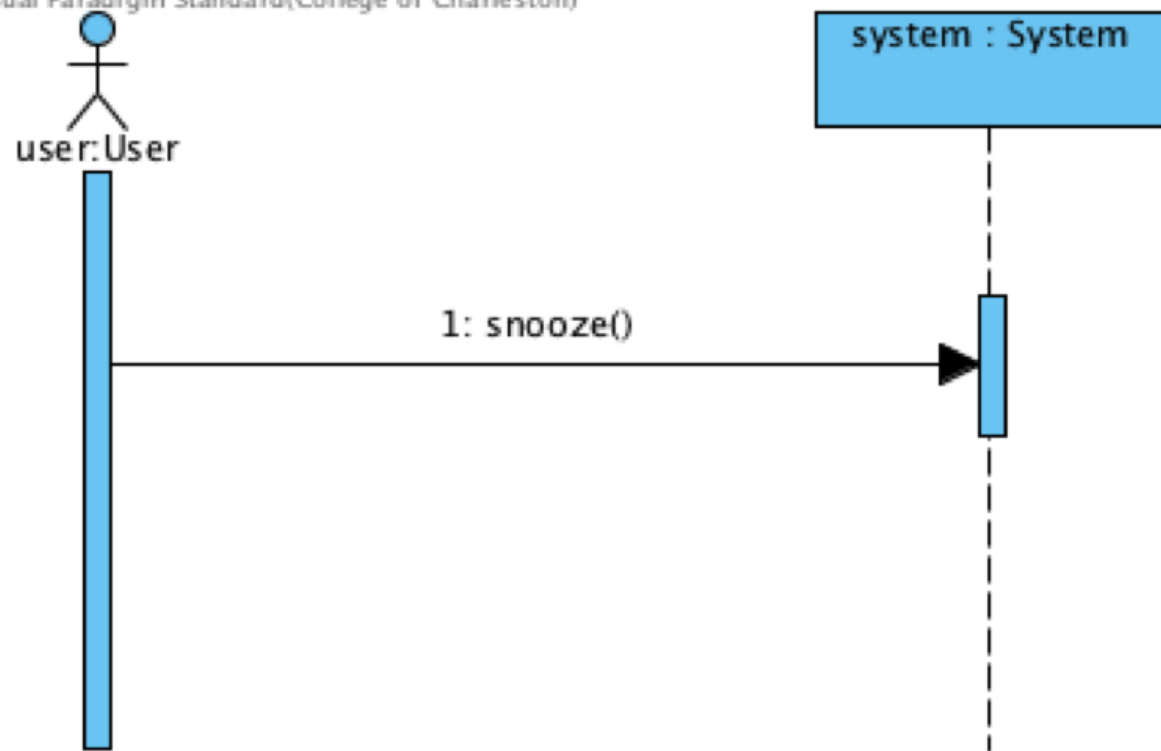
XII.   Contract CO12

A. Operation: exitTimeSetMode()
B. Cross References: SSDs - Set Time
C. Preconditions:
- The alarm clock is in time-setting mode
D. Postconditions:
- The alarm clock is no longer in time-setting mode

XIII. Contrat CO13
A. Opeartion: snoozeAlarm()
B. Cross References: SSDs - Snooze Alarm
C. Preconditions:
- There must be an alarm currently sounding
D. Postconditions:
- All alarms that were sounding at the time of activation are delayed to go off five minutes later

System Sequence Diagrams

user: User

system : SoftwareSystem

1: getRadioStatus()

1.1: status

2: getStation()

2.1: station

loop

[status = on]

3: checkStationChange()

3.1: change

alt

[change = up]   4: increaseStation()

[change = down]

sd Frame   5: decreaseStation()

6: getRadioStatus()

6.1: status

user:User

system : System

1: snooze()

user:User

system : System

1: switchTimeFormat()

user:User

system : System

1: enterTimeSetMode(alarmNumber : int)

2: setTime(hours : int, minutes : int)

3: exitTimeSetMode()

user:User

system : System

1: enterAlarmSetMode(alarmNumber : int)

2: setAlarmTime(hours : int, minutes : int)

3: exitAlarmSetMode()