Tomislav S. Mitic
DePaul University Winter 2013
SE 430
Assignment 2

Use Case UC1: Add Alarm

Primary Actor: End user of the alarm system

Stakeholders and Interests:

- End user of the alarm system: wants a straightforward, accurate and dependable way to add alarms to the alarm system which can alert her/him to important events which he/she must address

Preconditions: None

Success Guarantee (Post conditions): Alarm is added to the alarm system list and will sound at the appropriate time

Main Success Scenario (Basic Flow):

1. User submits data for new alarm
2. Java clock validates the date and time is in the future
3. Java clock validates the name of the alarm is unique
4. Java clock uses default sound file for alarm
5. Java clock uses specified sound recurrence
6. Java clock uses specified alarm recurrence
7. Java clock updates alarm clock system xml file with new alarm data
8. An updated list of alarms is displayed to the user

Extensions (or Alternate Flows):

2a. Invalid date and/or time
  1. Java clock displays an error popup to the user
  2. User must re-enter a valid date and/or time

3a. Invalid name (isn't unique)
  1. Java clock displays an error popup to the user
  2. User must re-enter a unique name

4a. User selects different sound file for alarm
  1. Java clock checks whether the sound file exists
    1a. Java clock unable to detect the existence of the specified sound file
      1. Java clock displays an error popup to the user
      2. User must re-enter location of sound file or choose a new sound file

2. Java clock validates sound file is appropriate format
   2a. Invalid sound file format
      1. Java clock displays an error popup to the user
      2. User must choose a new sound file with the appropriate format

7a. Java clock is unable to update alarm clock system xml file
1. Java clock displays an error popup with the appropriate information to the user of why xml can't be updated
2. User take necessary steps to rectify the problem

Special Requirements:
- Time can be customized to 12/24 hour time format
- Time display font can be customized based on font type and font size
- Time display font color can be customized
- Clock face color can be customized
- Alarm Button color can be customized
- Alarm clock can be minimized to system tray

Technology and Data Variations List:
- N/A

Frequency of Occurrence:
- Depends on the use of the Java clock alarm clock by the end user

Open Issues:
- N/A

Use Case UC2: Edit Alarm

Primary Actor: End user of the alarm system

Stakeholders and Interests:
- End user of the alarm system: wants a straightforward, accurate and dependable way to edit alarms within the alarm system so that they can alert her/him to important events which he/she must address

Preconditions: Alarm must already exist within the alarm clock system

Success Guarantee (Post Conditions): Alarm is changed and re-added to the alarm system list and will sound at the appropriate time

Main Success Scenario (Basic Flow):

1. User submits data for alarm
2. Java clock validates the date and time is in the future
3. Java clock uses default sound file for alarm
4. Java clock uses specified sound recurrence
5. Java clock uses specified alarm recurrence
6. Java clock updates alarm clock system xml file with new alarm data
7. An updated list of alarms is displayed to the user

Extensions (or Alternate Flows):

2a. Invalid date and/or time
  1. Java clock displays an error popup to the user
  2. User must re-enter a valid date and/or time

3a. User selects different sound file for alarm
  1. Java clock checks whether the sound file exists
    1a. Java clock unable to detect the existence of the specified sound file
      1. Java clock displays an error popup to the user
      2. User must re-enter location of sound file or choose a new sound file
  2. Java clock validates sound file is appropriate format
    2a. Invalid sound file format
      1. Java clock displays an error popup to the user
      2. User must choose a new sound file with the appropriate format

6a. Java clock is unable to update alarm clock system xml file
  1. Java clock displays an error popup with the appropriate information to the user of why xml can't be updated
  2. User take necessary steps to rectify the problem

Special Requirements:

- Time can be customized to 12/24 hour time format
- Time display font can be customized based on font type and font size
- Time display font color can be customized
- Clock face color can be customized
- Alarm Button color can be customized
- Alarm clock can be minimized to system tray

Technology and Data Variations List:

- N/A

Frequency of Occurrence:

- Depends on the use of the Java clock alarm clock by the end user

Open Issues:

- N/A

Use Case UC3: Delete an Alarm

Primary Actor: End user of the alarm system

Stakeholders and Interests:

- End user of the alarm system: wants a straightforward, accurate and dependable way to delete an alarm within the alarm system that is no longer valid

Preconditions: Alarm must already exist within the alarm clock system

Success Guarantee (Post Conditions): Alarm is deleted from the alarm system list

Main Success Scenario (Basic Flow):

1. User submits alarm for deletion
2. Java clock deletes the alarm from the alarm clock system
3. Java clock updates alarm clock system xml file with new alarm data
4. An updated list of alarms is displayed to the user

Extensions (or Alternate Flows):

4a. Java clock is unable to update alarm clock system xml file

1. Java clock displays an error popup with the appropriate information to the user of why xml can't be updated
2. User take necessary steps to rectify the problem

Special Requirements:

- Time can be customized to 12/24 hour time format
- Time display font can be customized based on font type and font size
- Time display font color can be customized
- Clock face color can be customized
- Alarm Button color can be customized
- Alarm clock can be minimized to system tray

Technology and Data Variations List:

- N/A

Frequency of Occurrence:

- Depends on the use of the Java clock alarm clock by the end user

Open Issues:
- N/A

Use Case UC4: Delete All Alarms

Primary Actor: End user of the alarm system

Stakeholders and Interests:
- End user of the alarm system: wants a straightforward, accurate and dependable way to delete all alarms within the alarm system that are no longer valid

Preconditions: Alarms must already exist within the alarm clock system

Success Guarantee (Post Conditions): Alarms are deleted from the alarm system list

Main Success Scenario (Basic Flow):
1. User submits all alarms for deletion
2. Java clock deletes all alarms from the alarm clock system
3. Java clock updates alarm clock system xml file with new alarm data
4. An updated list of no alarms is displayed to the user

Extensions (or Alternate Flows):
4a. Java clock is unable to update alarm clock system xml file
  1. Java clock displays an error popup with the appropriate information to the user of why xml can't be updated
  2. User take necessary steps to rectify the problem

Special Requirements:
- Time can be customized to 12/24 hour time format
- Time display font can be customized based on font type and font size
- Time display font color can be customized
- Clock face color can be customized
- Alarm Button color can be customized
- Alarm clock can be minimized to system tray

Technology and Data Variations List:
- N/A

Frequency of Occurrence:
- Depends on the use of the Java clock alarm clock by the end user

Open Issues:
- N/A

Use Case UC5: Alarm Executes Notifying End User of an Event

Primary Actor: End user of the alarm system

Stakeholders and Interests:
- End user of the alarm system: wants the Java clock to accurately notify him/her of when an upcoming event that they must address

Preconditions: Alarm must already exist within the alarm clock system

Success Guarantee (Post Conditions): Alarm sounds off notifying end user of the event

Main Success Scenario (Basic Flow):
1. The time arrives for the alarm to sound
2. Alarm sounds off with predetermined sound recurrance
3. Alarm pop up presented to the user
4. User shuts off alarm by closing alarm pop up
5. Alarm has no recurrence and is permanently removed from the alarm clock system
6. Java clock updates alarm clock system xml file with new alarm data
7. An updated list of alarms is displayed to the user

Extensions (or Alternate Flows):
4a. User snoozes alarm for a specified interval
   1. Java clock updates system xml file with new info for clock
      1a. Java clock unable to update XML file
         1. An error message is displayed to the user with information about why xml can't be updated
         2. User must take appropriate measures to rectify problem
5a. Alarm has recurrence
   1. Java clock updates system xml file with appropriate recurrence for the alarm
      1a. Java clock unable to update XML file
         1. An error message is displayed to the user with information about why xml can't be updated
         2. User must take appropriate measures to rectify problem
6a. Java clock is unable to update alarm clock system xml file
   1. Java clock displays an error popup with the appropriate information to the user of why xml can't be updated

2. User takes necessary steps to rectify the problem

Special Requirements:
- Time can be customized to 12/24 hour time format
- Time display font can be customized based on font type and font size
- Time display font color can be customized
- Clock face color can be customized
- Alarm Button color can be customized
- Alarm clock can be minimized to system tray

Technology and Data Variations List:
- N/A

Frequency of Occurence:
- Depends on the use of the Java clock alarm clock by the end user

Open Issues:
- N/A
