Kyle Zeller and Andrew Braunagel 2/29/2016

Priority Management Assistant

The Priority Management Assistant (PMA) will be a desktop application for windows and linux that will prioritize the everyday life of an individual. The user must be able to input their class assignments such as homeworks, projects, labs, upcoming exams and quizzes, meetings, classes, etc. The program needs to be well organized and have good structure for a graphical user interface to make it as simple as possible for the user to create, remove, or edit an upcoming assignment. This graphical user interface must provide clickable buttons for certain features such as adding, removing, and saving an assignment. As the program is running, there needs to be a list of all assignments with deadlines for some set interval of time. The program needs to be able to update itself every time the user adds, removes, or edits their assignments. There also needs to be a way to list the assignments in order of importance according to the user's specification when he/she submits the assignment. There will be a feature that the user can configure that will enable screen pop-ups as reminders for those assignments.

1. System Requirements

a. Enumerated Functional Requirements

ID	Points	Requirement
REQ-1	4	System shall allow user to input assignments w/ (title, due date, & priority)
REQ-2	4	System shall allow assignments to be saved for later use & modification
REQ-3	2	System shall allow saved file to be opened as a list file of the schedule & is updated with every change made to the assignment list
REQ-4	3	System shall allow assignments to be deleted by user
REQ-5	3	System shall allow assignments to be deleted by the program after due date for any given assignment has passed
REQ-6	3	System shall allow saved assignments to be edited by user
REQ-7	2	System shall allow user to choose the assignment priority level

		(green, yellow, or red)
REQ-8	5	System shall order assignments by due date, and priority
REQ-9	10	System shall run in background to allow pop-ups that can be used as reminders
REQ-10	2	System shall allow user to configure a checkbox that enables or disables the pop-ups
REQ-11	8	System shall be implemented with graphical user interface as a way to configure and manage assignments

We will be constantly working with the feedback from our customer in weekly meetings to assess any changes, issues, and improvements to be made with the product. As a group we will be constantly be meeting up to collaborate and innovate new solutions and approaches to design the product. These meetings will consist of discussing our progress with the product, brainstorming new ideas for the product, and effectively leave the meeting with a sense of assurance of what is expected by the next meeting. These meetings may or may not be discussions through skype, google hangouts, or facebook as a way to actively discuss the product. This management process will be very dynamic as the way we meet up throughout the process of developing the product, in regard to the way the meetings go. The process of how we are to complete the product depends largely on how we communicate with each other and the customer. Our product similar to the process of how we manage this product will adapt to any changes and accommodations that occur. Our management process for this project will allow for optimization of the product later on. Team-work is key.

This desktop application will allow the user to set their daily/weekly priorities from their computer and our software will immediately prioritize their activities according to date, time, topic, and set priority level (green, yellow, or red). Then our program will generate a list file that the user can open and in it is a detailed schedule of what activities should come first day-by-day. After the program is first run, the user will see options on the screen to add, store, and save their activities. The user interface will be designed to be “pretty” or appealing to the user with clickable GUI’s. Our desktop application later on will have check-boxes next to the save button for each newly added activity, to generate screen-size pop-ups for the user.

Identifier	**User Story**	**Points**
ST-1	As a user, I can add an upcoming assignment	4
ST-2	As a user, I can delete an assignment to show that I completed it	3
ST-3	As a user, I can edit an assignment to change some of its information	3
ST-4	As a user, I can set the importance of an assignment to know which ones are the most important	2
ST-5	As a user, I can allow the program to create a pop-up to remind me that I have an upcoming assignment due	10
ST-6	As a user, I can save my assignments to be able to check them again later	4
ST-7	As a user, I can choose to order assignments by due date, and by priority, to keep it organized	5

Estimation of Project Effort		
Identifier	Points	Time Estimation
ST-1	4	2 Days / 2 people
ST-2	3	2 Days / 2 people
ST-3	3	3 Days / 2 people
ST-4	2	1 Days / 2 people
ST-5	10	5 Days / 2 people
ST-6	4	2 Days / 2 people
ST-7	5	3 Days / 2 people

Actor	Actor’s Goal	Use Case Name
User	To type in assignments by title, due date, & priority.	Input (UC-1)
User	To store all changes made to assignments in a list file.	Save (UC-2)
User	To see upcoming due assignments & update the list with every change.	User-Update (UC-3)
User	To get rid of the assignments that no longer matter.	Delete (UC-4)
User	To get rid of the unnecessary overdue assignments.	Auto-Update (UC-5)
User	To edit assignments with the wrong information and/or information that needs to be updated.	Edit (UC-6)
User	To assign the importance level of an assignment.	Set Priority (UC-7)
User	To look at the list file at any time with everything prioritized in a neat format.	Sort List File (UC-8)
Program	To be alerted about upcoming assignments if the due date is close to the current day & time.	Notifications (UC-9)
User	To be able to make pop-up alerts optional by clicking a checkbox.	Check Box (UC-10)
Program	To easily access all the features of the program through text boxes, drop-down select buttons, check boxes, & clickable items.	GUI (UC-11)

Traceability Matrix

Req’t	UC-1	UC-2	UC-3	UC-4	UC-5	UC-6	UC-7	UC-8	UC-9	UC-10	UC-11
REQ-1	X						X				X
REQ-2		X									X
REQ-3			X								X
REQ-4				X							X
REQ-5					X						

REQ-6						X					X
REQ-7							X				X
REQ-8								X			X
REQ-9									X	X	X
REQ-10										X	X
REQ-11											X

Use Case UC-1:	**Input**
Related Requirements:	REQ-1
Initiating Actor:	User
Actor's Goal:	To type in assignments by title, due date, & priority level.
Participating Actors:	Program
Preconditions:	● The user must say "yes" to the program prompt if they want to make an assignment.
Postconditions:	● The user must say "yes" to the program prompt if they are done making assignments.
Flow of Events for Main Success Scenario:	
→ 1. The user runs the program, so they can create, edit, and / or delete assignments. ← 2. The user is prompted to enter "yes" or "no" if they want to make an assignment. → 3. The user provides one of these two choices. ← 4. The user is prompted to enter a title, due date, & priority level for their assignment. → 5. The user provides this information accurately. ← 6. The user is prompted to enter "yes" or "no" if they are done making assignments. → 7. The user provides one of these two choices.	
Flow of Events for Extensions:	
2a. User enters invalid information, so they are prompted continuously until correct information has been entered.	

Use Case UC-2:	**Save**
Related Requirements:	REQ-2
Initiating Actor:	User
Actor’s Goal:	To store all changes made to assignments in a list file.
Participating Actors:	Program
Preconditions:	● The user must be finished creating, deleting, and / or editing assignments.
Postconditions:	● None
Flow of Events for Main Success Scenario:	
1a. The program will automatically generate a list file for all the assignments: ← 1. The program will automatically generate a list file at the end of the program.	

Use Case UC-3:	**User-Update**
Related Requirements:	REQ-3
Initiating Actor:	User
Actor’s Goal:	To see upcoming due assignments & update the list with every change.
Participating Actors:	Program
Preconditions:	● The user must be finished creating, deleting, and / or editing assignments. ● The user must double-click on the list file to see the contents of their assignments.
Postconditions:	● The user must exit out of the list file to close out of the file.
Flow of Events for Main Success Scenario:	
1a. The program will update the contents of the list upon startup & finishing: → 1. The user runs the program, so they can create, edit, and / or delete assignments. ← The program will automatically update the contents of the list after it loads the list from the list file.	

1b. The user will attempt to view the contents of their assignments:
→ 1. The user double-clicks on the generated list file, so they can view the contents of their assignments.
→ 2. The user exits out of the list file when they are done, which closes the file.

Use Case UC-4:	**Delete**
Related Requirements:	REQ-4
Initiating Actor:	User
Actor's Goal:	To get rid of the assignments that no longer matter.
Participating Actors:	Program
Preconditions:	● The user must say "yes" to the program prompt if they want to delete an assignment.
Postconditions:	● The user must say "yes" to the program prompt if they are done deleting assignments.

Flow of Events for Main Success Scenario:

→ 1. The user runs the program, so they can create, edit, and / or delete assignments.
← 2. The user is prompted to enter "yes" or "no" if they want to delete an assignment.
→ 3. The user provides one of these two choices.
← 4. The user is given a list of all their assignments & are prompted to enter "yes" or "no" if they want more information about an assignment.
→ 5. The user provides this information accurately.
1a. The user wants to get more details about an assignment before deleting an assignment:
← 6. The user is prompted to enter the title of an assignment.
→ 7. The user provides this information accurately.
← 8. The user is given a list of all their assignments & are prompted to enter "yes" or "no" if they want more information about an assignment.
1b. The user doesn't want to get more details about an assignment before deleting an assignment:
← 6. The user is prompted to enter the title of an assignment to delete.
→ 7. The user provides this information accurately.
← 8. The user is prompted to enter "yes" or "no" if they are done deleting assignments.
→ 9. The user provides one of these two choices.

Flow of Events for Extensions:

2a. User enters invalid information, so they are prompted continuously until correct information has been entered.

Use Case UC-5:	Auto-Update
Related Requirements:	REQ-5
Initiating Actor:	User
Actor's Goal:	To get rid of the unnecessary overdue assignments.
Participating Actors:	Program
Preconditions:	● None
Postconditions:	● None
Flow of Events for Main Success Scenario:	
1a. The program will update the contents of the list upon startup: → 1. The user runs the program, so they can create, edit, and / or delete assignments. ← The program will automatically update the contents of the list after it loads the list from the list file, by checking for overdue assignments to delete.	

Use Case UC-6:	Edit
Related Requirements:	REQ-6
Initiating Actor:	User
Actor's Goal:	To edit assignments with the wrong information and/or information that needs to be updated.
Participating Actors:	Program
Preconditions:	● The user must say "yes" to the program prompt if they want to delete an assignment.
Postconditions:	● The user must say "yes" to the program prompt if they are done deleting assignments.
Flow of Events for Main Success Scenario:	
→ 1. The user runs the program, so they can create, edit, and / or delete assignments. ← 2. The user is prompted to enter "yes" or "no" if they want to edit an assignment. → 3. The user provides one of these two choices. ← 4. The user is given a list of all their assignments & are prompted to enter "yes" or "no" if they want more information about an assignment.	

→ 5. The user provides this information accurately. 1a. The user wants to get more details about an assignment before deleting an assignment: ← 6. The user is prompted to enter the title of an assignment. → 7. The user provides this information accurately. ← 8. The user is given a list of all their assignments & are prompted to enter “yes” or “no” if they want more information about an assignment. 1b. The user doesn’t want to get more details about an assignment before deleting an assignment: ← 6. The user is prompted to enter the title of an assignment to edit. → 7. The user provides this information accurately. ← 8. The user is prompted to enter “N” or “T” or “P” if they want to edit an assignment’s name, due date, or priority. → 9. The user provides one of these three choices. 1c. The user wants to edit the name (“N”) ← 10. The user is prompted to enter a title for their assignment. → 11. The user provides this information accurately. 1d. The user wants to edit the name (“T”) ← 10. The user is prompted to enter a due date for their assignment. → 11. The user provides this information accurately. 1e. The user wants to edit the name (“P”) ← 10. The user is prompted to enter a priority level for their assignment. → 11. The user provides this information accurately. ← 12. The user is prompted to enter “yes” or “no” if they are done editing this assignment. → 13. The user provides one of these two choices. ← 14. The user is prompted to enter “yes” or “no” if they are done editing assignments. → 15. The user provides one of these two choices.
Flow of Events for Extensions:
2a. User enters invalid information, so they are prompted continuously until correct information has been entered.

Use Case UC-7:	**Set Priority**
Related Requirements:	REQ-7 & REQ-1
Initiating Actor:	User
Actor’s Goal:	To assign the importance level of an assignment.
Participating Actors:	Program
Preconditions:	● The user must say “yes” to the program prompt if they want to make an assignment.

Postconditions:	● The user must say “yes” to the program prompt if they are done making assignments.
Flow of Events for Main Success Scenario:	
→ 1. The user runs the program, so they can create, edit, and / or delete assignments. ← 2. The user is prompted to enter “yes” or “no” if they want to make an assignment. → 3. The user provides one of these two choices. ← 4. The user is prompted to enter a title, due date, & priority level for their assignment. → 5. The user provides this information accurately. ← 6. The user is prompted to enter “yes” or “no” if they are done making assignments. → 7. The user provides one of these two choices.	
Flow of Events for Extensions:	
2a. User enters invalid information, so they are prompted continuously until correct information has been entered.	

Use Case UC-8:	**Sort List File**
Related Requirements:	REQ-8
Initiating Actor:	User
Actor’s Goal:	To look at the list file at any time with everything prioritized in a neat format.
Participating Actors:	Program
Preconditions:	● None
Postconditions:	● None
Flow of Events for Main Success Scenario:	
→ 1. The user runs the program, so they can create, edit, and / or delete assignments. ← 2. The program will sort the contents of the list upon finishing in an orderly manner.	

Use Case UC-9:	**Notifications**
Related Requirements:	REQ-9
Initiating Actor:	Program

Actor's Goal:	To be alerted about upcoming assignments if the due date is close to the current day & time.
Participating Actors:	None
Preconditions:	● None
Postconditions:	● None
Flow of Events for Main Success Scenario:	
← The program will look at the contents of the list file and will generate alert notification times at 24,12,6,3,2, & 1 hr times. The size of the pop-up depends on the time before the deadline & the color of the pop-up depends on the priority level.	

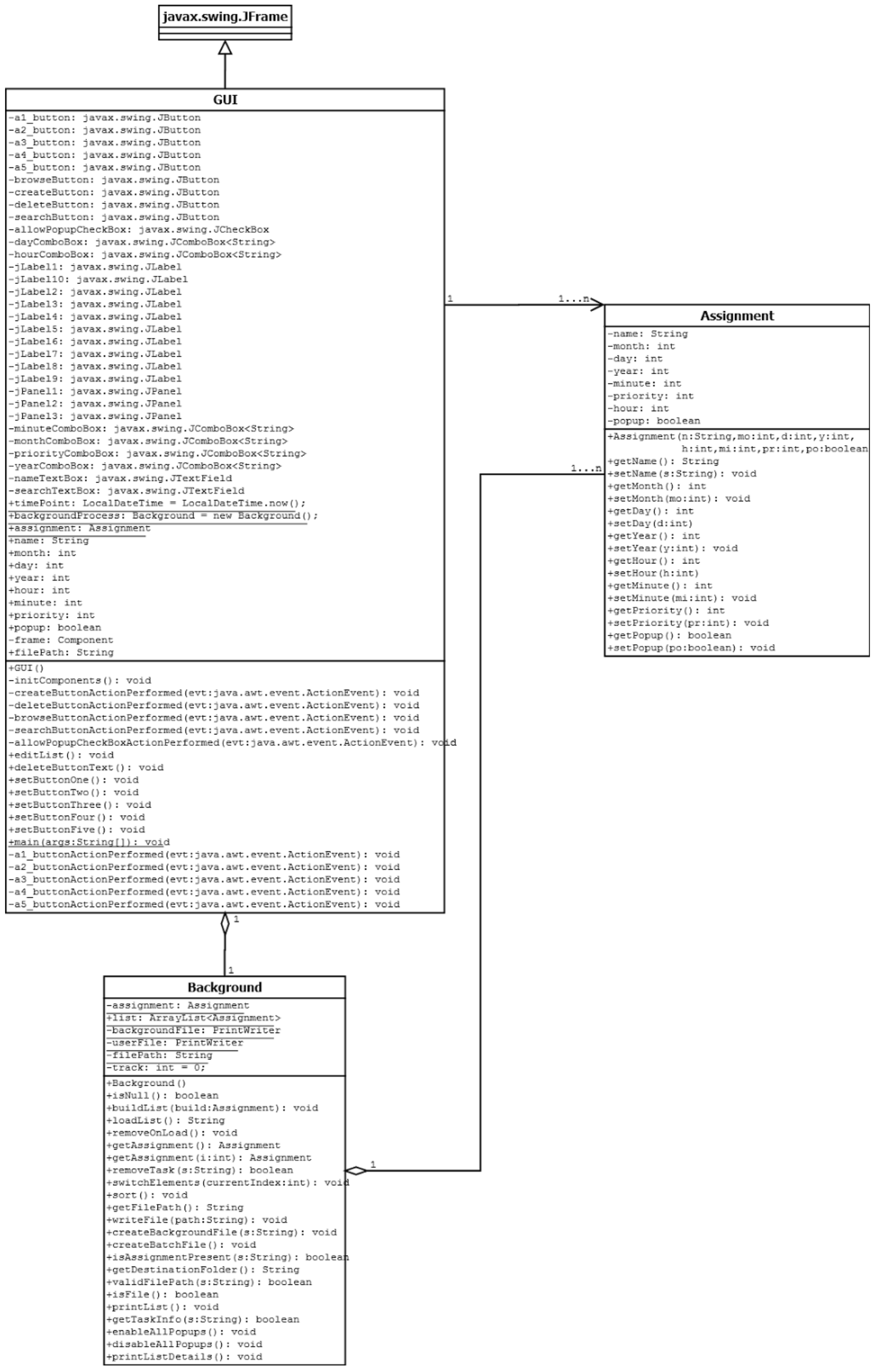
Use Case UC-10:	**Check Box**
Related Requirements:	REQ10 & REQ9
Initiating Actor:	User
Actor's Goal:	To be able to make pop-up alerts optional by clicking a checkbox.
Participating Actors:	Program
Preconditions:	● None
Postconditions:	● None
Flow of Events for Main Success Scenario:	
→ 1. The user runs the program, so they can create, edit, and / or delete assignments. ← 2. The user is prompted to enter "yes" or "no" if they want to make an assignment. → 3. The user provides one of these two choices. ← 4. The user is prompted to enter a title, due date, & priority level for their assignment. → 5. The user provides this information accurately. ← 6. The user is prompted to enter "yes" or "no" if they are done making assignments. → 7. The user provides one of these two choices. ← 8. The user is prompted to click on a "check box" if they want pop-ups for their assignments. → 9. The user provides one of these two choices. ← 10. The program sets status of enabling or disabling the pop-ups.	

Use Case UC-11:	**GUI**

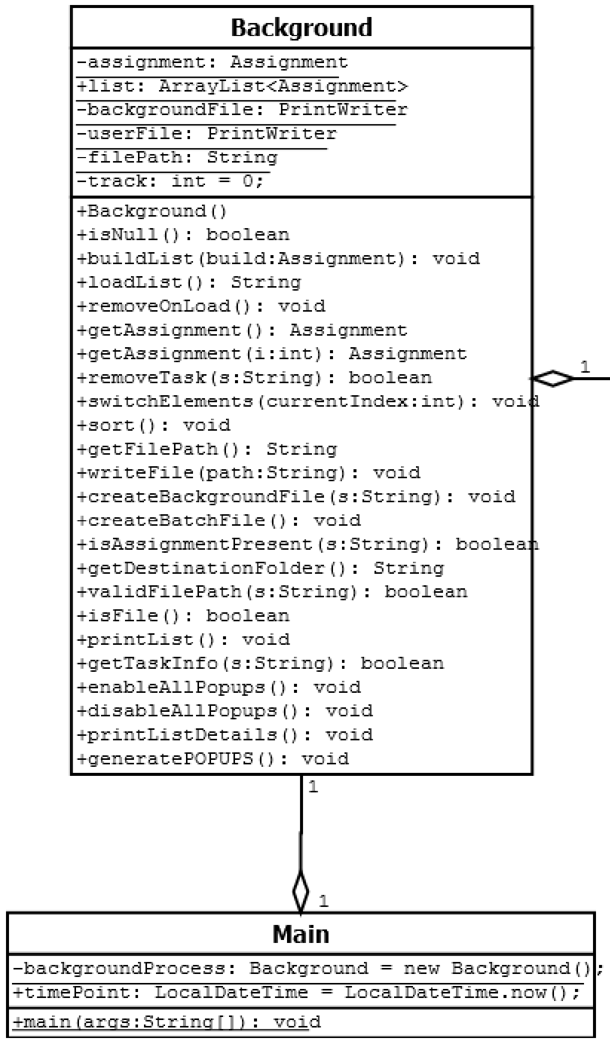
Related Requirements:	REQ-11, REQ-10, REQ-9, REQ-8, REQ-7, REQ-6, REQ-4, REQ-3, REQ-2, & REQ-1
Initiating Actor:	Program
Actor's Goal:	To easily access all the features of the program through text boxes, drop-down select buttons, check boxes, & clickable items.
Participating Actors:	None
Preconditions:	● None
Postconditions:	● None
Flow of Events for Main Success Scenario:	
← The program will feature check boxes for REQ-10, pop-ups with different sizes & colors for REQ-9, a neat format for displaying all the assignments organized for REQ-8, drop down menus for selecting the priority level by the colors: ("GREEN", "YELLOW", "RED") for REQ-7, assignments will be clickable so they can be edited in pop-up boxes for REQ-6, assignments will have checkbox pop-ups to delete the assignments for REQ-4, a neatly formatted list file for the user to do with as they please for REQ-3, checkbox popups to save new assignments or any changes made to assignments or the list of assignments for REQ-2, & pop-up checkboxes & dropdown menus for the the user to select from to create & edit their assignments for REQ-1.	

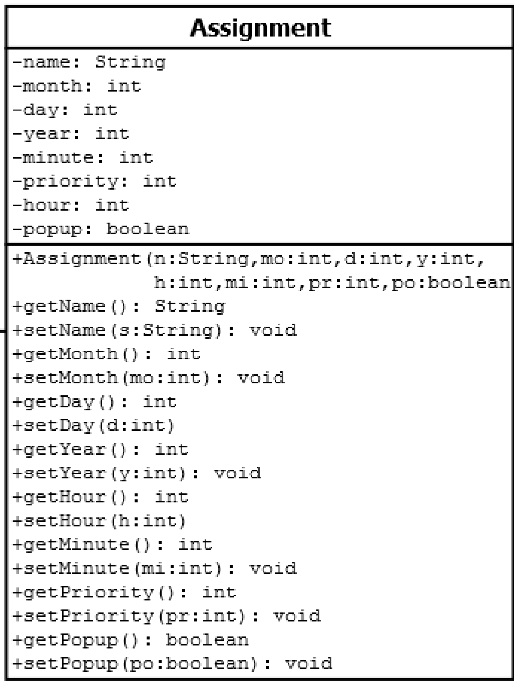
Class Diagrams

Actor 1 : User


javax.swing.JFrame
GUI
-a1_button: javax.swing.JButton
-a2_button: javax.swing.JButton
-a3_button: javax.swing.JButton
-a4_button: javax.swing.JButton
-a5_button: javax.swing.JButton
-browseButton: javax.swing.JButton
-createButton: javax.swing.JButton
-deleteButton: javax.swing.JButton
-searchButton: javax.swing.JButton
-allowPopupCheckBox: javax.swing.JCheckBox
-dayComboBox: javax.swing.JComboBox<String>
-hourComboBox: javax.swing.JComboBox<String>
-jLabel1: javax.swing.JLabel
-jLabel10: javax.swing.JLabel
-jLabel2: javax.swing.JLabel
-jLabel3: javax.swing.JLabel
-jLabel4: javax.swing.JLabel
-jLabel5: javax.swing.JLabel
-jLabel6: javax.swing.JLabel
-jLabel7: javax.swing.JLabel
-jLabel8: javax.swing.JLabel
-jLabel9: javax.swing.JLabel
-jPanel1: javax.swing.JPanel
-jPanel2: javax.swing.JPanel
-jPanel3: javax.swing.JPanel
-minuteComboBox: javax.swing.JComboBox<String>
-monthComboBox: javax.swing.JComboBox<String>
-priorityComboBox: javax.swing.JComboBox<String>
-yearComboBox: javax.swing.JComboBox<String>
-nameTextBox: javax.swing.JTextField
-searchTextBox: javax.swing.JTextField
+timePoint: LocalDateTime = LocalDateTime.now();
+backgroundProcess: Background = new Background();
+assignment: Assignment
+name: String
+month: int
+day: int
+year: int
+hour: int
+minute: int
+priority: int
+popup: boolean
-frame: Component
+filePath: String
+GUI()
-initComponents(): void
-createButtonActionPerformed(evt:java.awt.event.ActionEvent): void
-deleteButtonActionPerformed(evt:java.awt.event.ActionEvent): void
-browseButtonActionPerformed(evt:java.awt.event.ActionEvent): void
-searchButtonActionPerformed(evt:java.awt.event.ActionEvent): void
-allowPopupCheckBoxActionPerformed(evt:java.awt.event.ActionEvent): void
+editList(): void
+deleteButtonText(): void
+setButtonOne(): void
+setButtonTwo(): void
+setButtonThree(): void
+setButtonFour(): void
+setButtonFive(): void
+main(args:String[]): void
-a1_buttonActionPerformed(evt:java.awt.event.ActionEvent): void
-a2_buttonActionPerformed(evt:java.awt.event.ActionEvent): void
-a3_buttonActionPerformed(evt:java.awt.event.ActionEvent): void
-a4_buttonActionPerformed(evt:java.awt.event.ActionEvent): void
-a5_buttonActionPerformed(evt:java.awt.event.ActionEvent): void
1
1...n
Assignment
-name: String
-month: int
-day: int
-year: int
-minute: int
-priority: int
-hour: int
-popup: boolean
+Assignment(n:String,mo:int,d:int,y:int,
h:int,mi:int,pr:int,po:boolean
+getName(): String
+setName(s:String): void
+getMonth(): int
+setMonth(mo:int): void
+getDay(): int
+setDay(d:int)
+getYear(): int
+setYear(y:int): void
+getHour(): int
+setHour(h:int)
+getMinute(): int
+setMinute(mi:int): void
+getPriority(): int
+setPriority(pr:int): void
+getPopup(): boolean
+setPopup(po:boolean): void
1...n
1
1
Background
-assignment: Assignment
+list: ArrayList<Assignment>
-backgroundFile: PrintWriter
-userFile: PrintWriter
-filePath: String
-track: int = 0;
+Background()
+isNull(): boolean
+buildList(build:Assignment): void
+loadList(): String
+removeOnLoad(): void
+getAssignment(): Assignment
+getAssignment(i:int): Assignment
+removeTask(s:String): boolean
+switchElements(currentIndex:int): void
+sort(): void
+getFilePath(): String
+writeFile(path:String): void
+createBackgroundFile(s:String): void
+createBatchFile(): void
+isAssignmentPresent(s:String): boolean
+getDestinationFolder(): String
+validFilePath(s:String): boolean
+isFile(): boolean
+printList(): void
+getTaskInfo(s:String): boolean
+enableAllPopups(): void
+disableAllPopups(): void
+printListDetails(): void
1

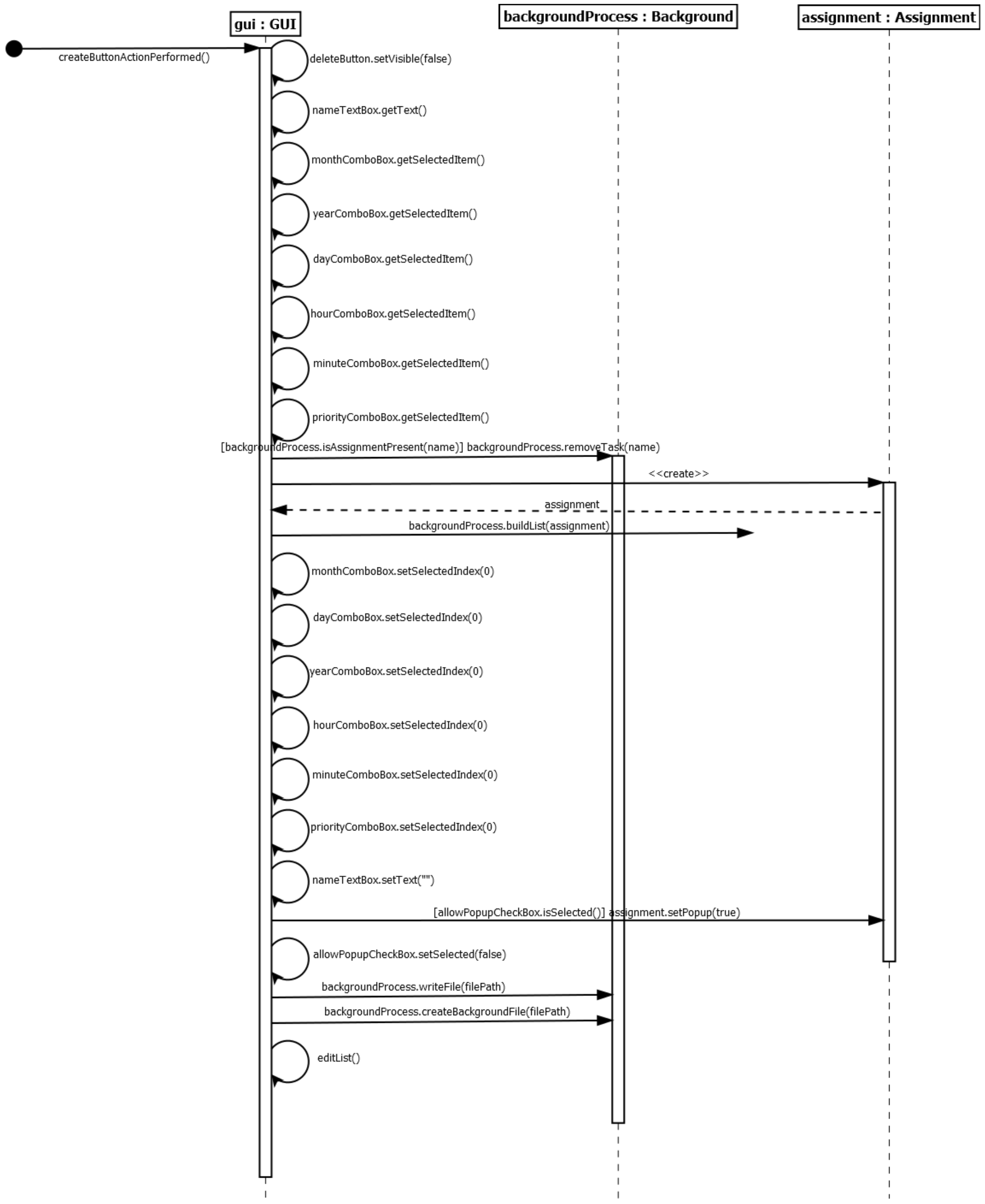
Actor 2 : System


Background
-assignment: Assignment
+list: ArrayList<Assignment>
-backgroundFile: PrintWriter
-userFile: PrintWriter
-filePath: String
-track: int = 0;
+Background()
+isNull(): boolean
+buildList(build:Assignment): void
+loadList(): String
+removeOnLoad(): void
+getAssignment(): Assignment
+getAssignment(i:int): Assignment
+removeTask(s:String): boolean
+switchElements(currentIndex:int): void
+sort(): void
+getFilePath(): String
+writeFile(path:String): void
+createBackgroundFile(s:String): void
+createBatchFile(): void
+isAssignmentPresent(s:String): boolean
+getDestinationFolder(): String
+validFilePath(s:String): boolean
+isFile(): boolean
+printList(): void
+getTaskInfo(s:String): boolean
+enableAllPopups(): void
+disableAllPopups(): void
+printListDetails(): void
+generatePOPUPS(): void
1
1...n
1
1
Assignment
-name: String
-month: int
-day: int
-year: int
-minute: int
-priority: int
-hour: int
-popup: boolean
+Assignment(n:String,mo:int,d:int,y:int, h:int,mi:int,pr:int,po:boolean
+getName(): String
+setName(s:String): void
+getMonth(): int
+setMonth(mo:int): void
+getDay(): int
+setDay(d:int)
+getYear(): int
+setYear(y:int): void
+getHour(): int
+setHour(h:int)
+getMinute(): int
+setMinute(mi:int): void
+getPriority(): int
+setPriority(pr:int): void
+getPopup(): boolean
+setPopup(po:boolean): void
Main
-backgroundProcess: Background = new Background();
+timePoint: LocalDateTime = LocalDateTime.now();
+main(args:String[]): void

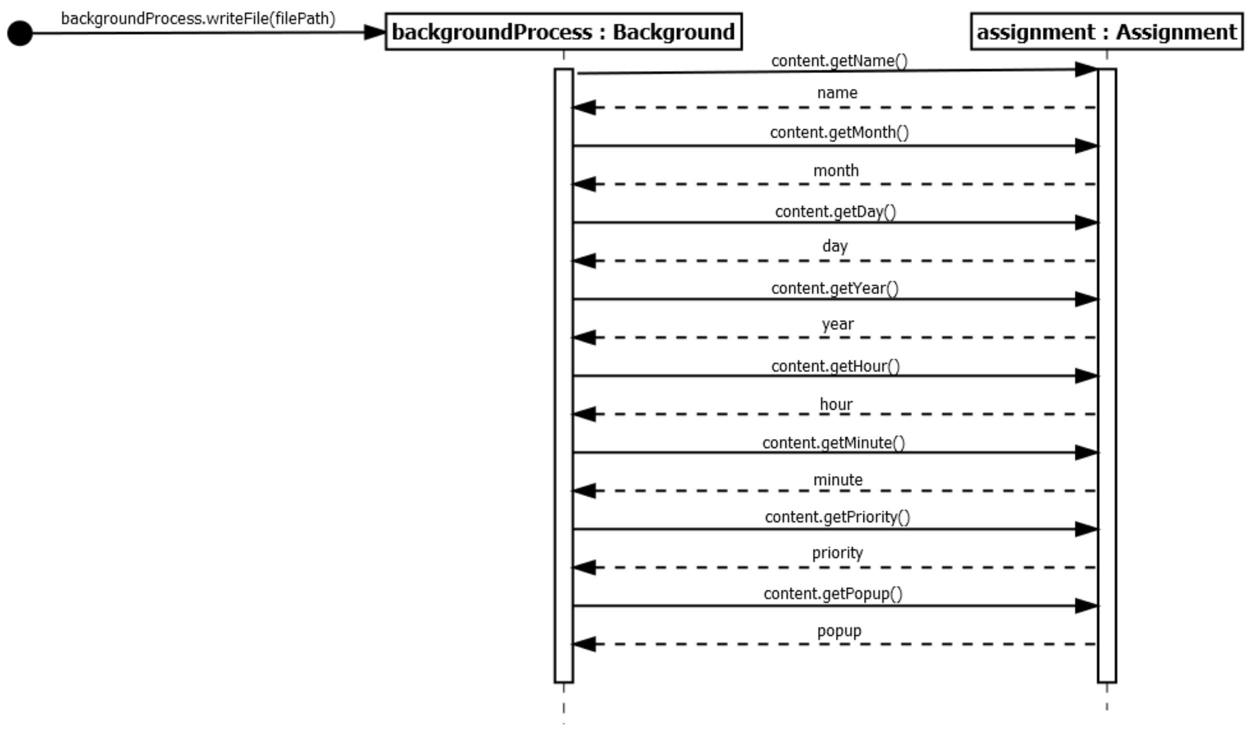


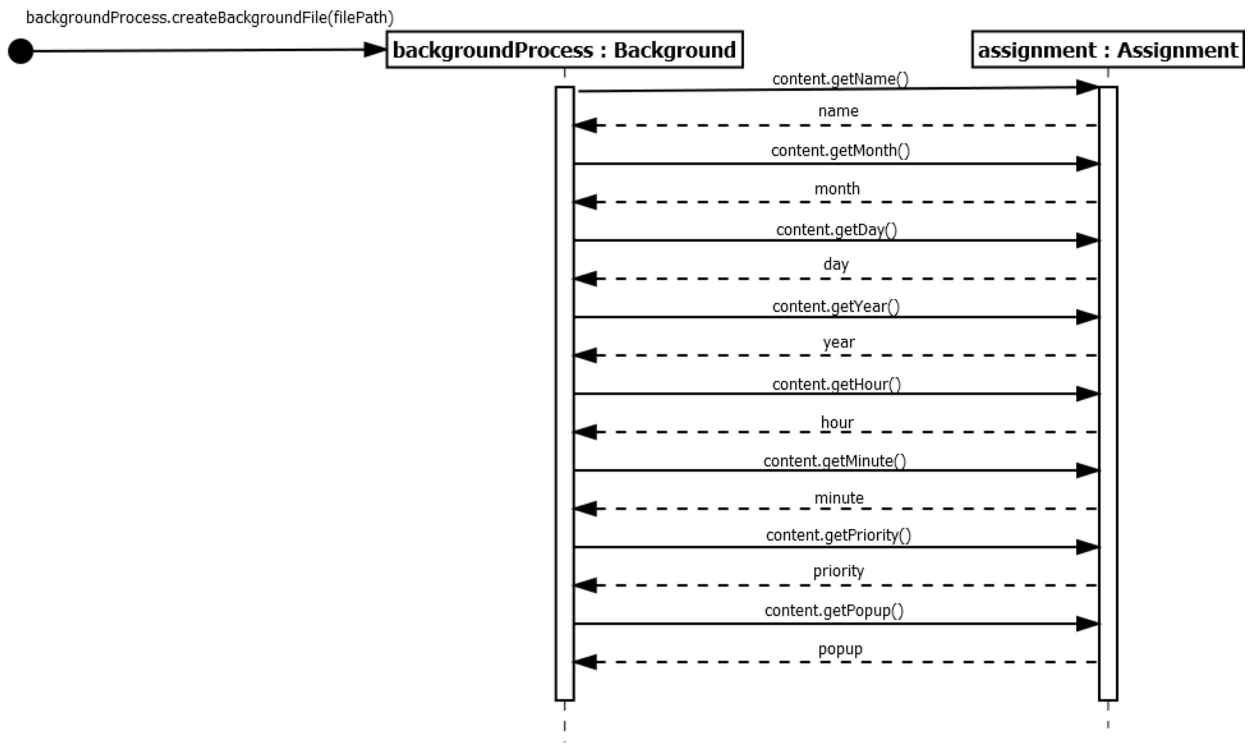
Sequence Diagrams

Use Case 1 : Input

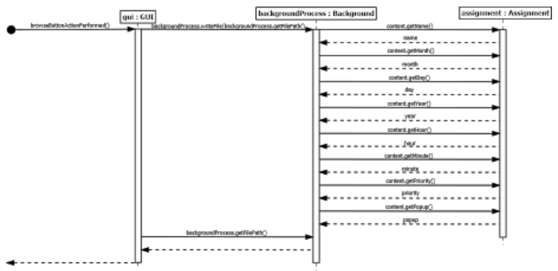

gui : GUI
backgroundProcess : Background
assignment : Assignment
createButtonActionPerformed()
deleteButton.setVisible(false)
nameTextBox.getText()
monthComboBox.getSelectedItem()
yearComboBox.getSelectedItem()
dayComboBox.getSelectedItem()
hourComboBox.getSelectedItem()
minuteComboBox.getSelectedItem()
priorityComboBox.getSelectedItem()
[backgroundProcess.isAssignmentPresent(name)] backgroundProcess.removeTask(name)
<<create>>
assignment
backgroundProcess.buildList(assignment)
monthComboBox.setSelectedIndex(0)
dayComboBox.setSelectedIndex(0)
yearComboBox.setSelectedIndex(0)
hourComboBox.setSelectedIndex(0)
minuteComboBox.setSelectedIndex(0)
priorityComboBox.setSelectedIndex(0)
nameTextBox.setText("")
[allowPopupCheckBox.isSelected()] assignment.setPopup(true)
allowPopupCheckBox.setSelected(false)
backgroundProcess.writeFile(filePath)
backgroundProcess.createBackgroundFile(filePath)
editList()

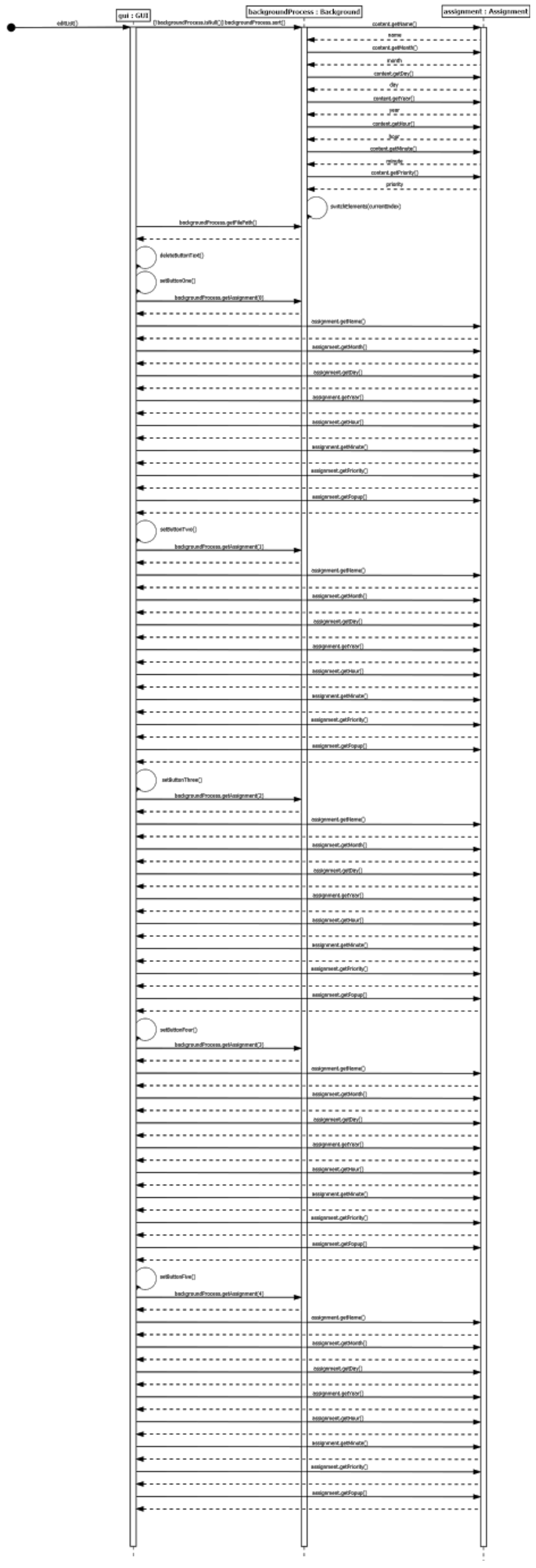
Use Case 2 : Save



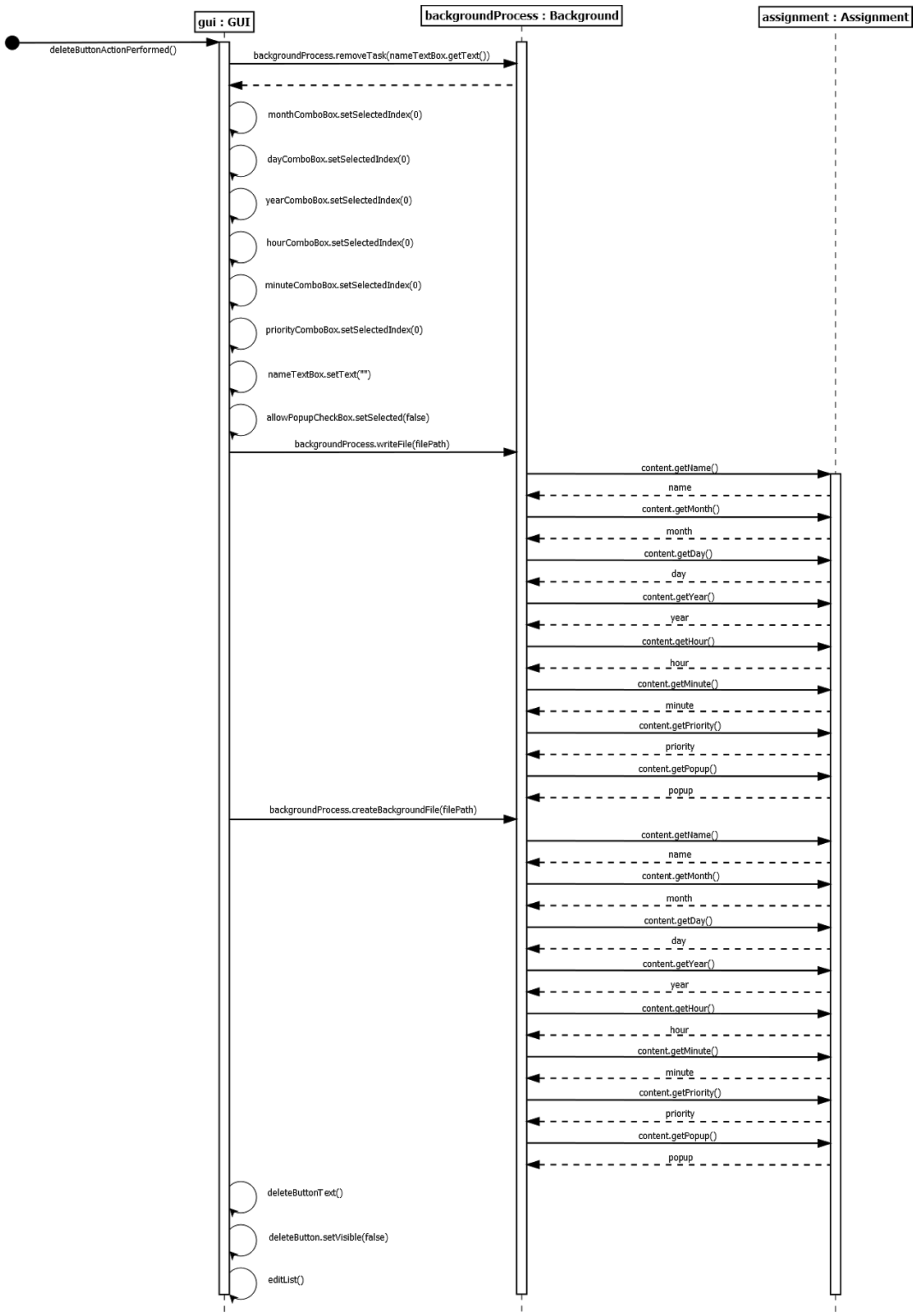


Use Case 3 : User Update

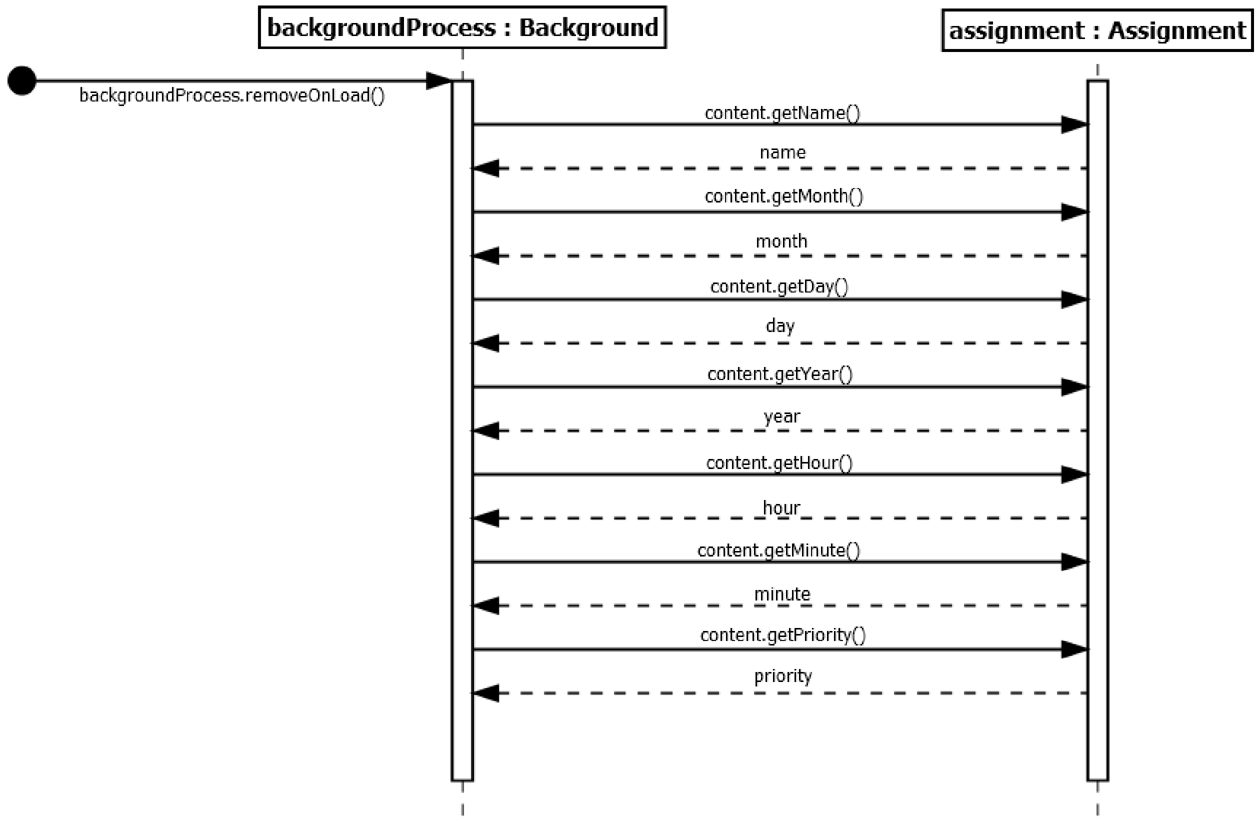

gui : GUI
backgroundProcess : Background
assignment : Assignment
browseButtonActionPerformed()
backgroundProcess.writeFile(backgroundProcess.getFilePath())
content.getName()
name
content.getMonth()
month
content.getDay()
day
content.getYear()
year
content.getHour()
hour
content.getMinute()
minute
content.getPriority()
priority
content.getPopup()
popup
backgroundProcess.getFilePath()


gui : GUI
backgroundProcess : Background
assignment : Assignment
editList()
[!backgroundProcess.isNull()] backgroundProcess.sort()
content.getName()
name
content.getMonth()
month
content.getDay()
day
content.getYear()
year
content.getHour()
hour
content.getMinute()
minute
content.getPriority()
priority
switchElements(currentIndex)
backgroundProcess.getFilePath()
deleteButtonText()
setButtonOne()
backgroundProcess.getAssignment(0)
assignment.getName()
assignment.getMonth()
assignment.getDay()
assignment.getYear()
assignment.getHour()
assignment.getMinute()
assignment.getPriority()
assignment.getPopup()
setButtonTwo()
backgroundProcess.getAssignment(1)
assignment.getName()
assignment.getMonth()
assignment.getDay()
assignment.getYear()
assignment.getHour()
assignment.getMinute()
assignment.getPriority()
assignment.getPopup()
setButtonThree()
backgroundProcess.getAssignment(2)
assignment.getName()
assignment.getMonth()
assignment.getDay()
assignment.getYear()
assignment.getHour()
assignment.getMinute()
assignment.getPriority()
assignment.getPopup()
setButtonFour()
backgroundProcess.getAssignment(3)
assignment.getName()
assignment.getMonth()
assignment.getDay()
assignment.getYear()
assignment.getHour()
assignment.getMinute()
assignment.getPriority()
assignment.getPopup()
setButtonFive()
backgroundProcess.getAssignment(4)
assignment.getName()
assignment.getMonth()
assignment.getDay()
assignment.getYear()
assignment.getHour()
assignment.getMinute()
assignment.getPriority()
assignment.getPopup()

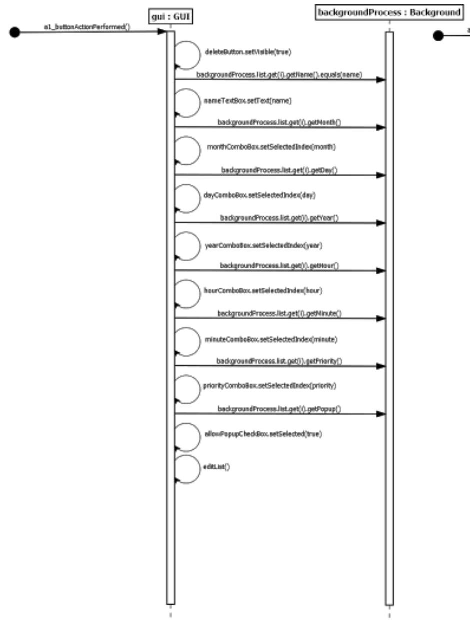
Use Case 4 : Delete

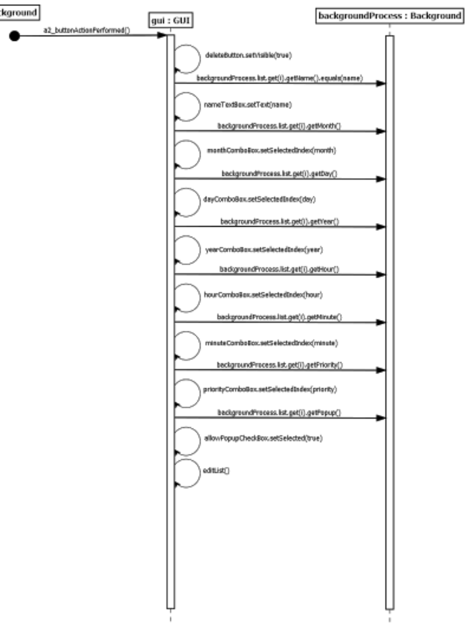

gui : GUI
backgroundProcess : Background
assignment : Assignment
deleteButtonActionPerformed()
backgroundProcess.removeTask(nameTextBox.getText())
monthComboBox.setSelectedIndex(0)
dayComboBox.setSelectedIndex(0)
yearComboBox.setSelectedIndex(0)
hourComboBox.setSelectedIndex(0)
minuteComboBox.setSelectedIndex(0)
priorityComboBox.setSelectedIndex(0)
nameTextBox.setText("")
allowPopupCheckBox.setSelected(false)
backgroundProcess.writeFile(filePath)
content.getName()
name
content.getMonth()
month
content.getDay()
day
content.getYear()
year
content.getHour()
hour
content.getMinute()
minute
content.getPriority()
priority
content.getPopup()
popup
backgroundProcess.createBackgroundFile(filePath)
content.getName()
name
content.getMonth()
month
content.getDay()
day
content.getYear()
year
content.getHour()
hour
content.getMinute()
minute
content.getPriority()
priority
content.getPopup()
popup
deleteButtonText()
deleteButton.setVisible(false)
editList()

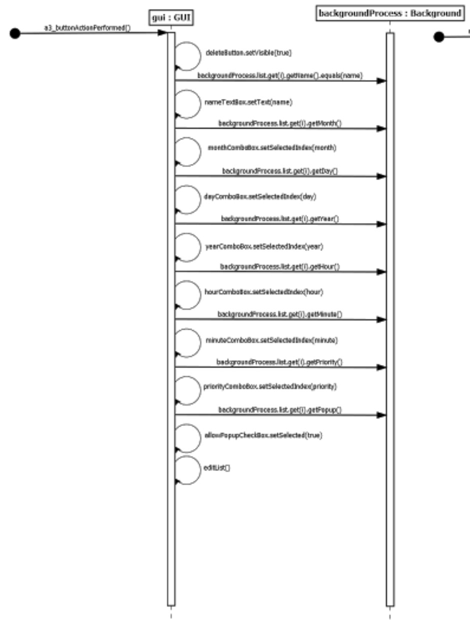
Use Case 5 : Auto Update

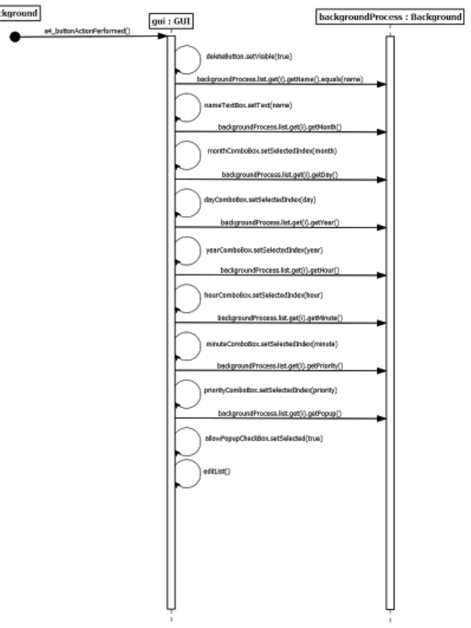


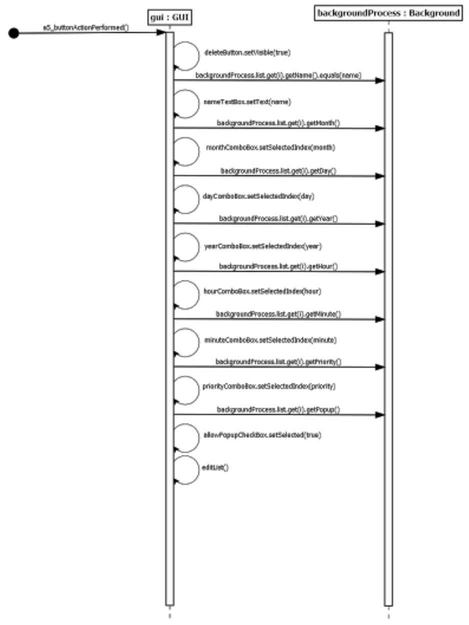
Use Case 6 : Edit


gui : GUI
backgroundProcess : Background
a1_buttonActionPerformed()
deleteButton.setVisible(true)
backgroundProcess.list.get(i).getName().equals(name)
nameTextBox.setText(name)
backgroundProcess.list.get(i).getMonth()
monthComboBox.setSelectedIndex(month)
backgroundProcess.list.get(i).getDay()
dayComboBox.setSelectedIndex(day)
backgroundProcess.list.get(i).getYear()
yearComboBox.setSelectedIndex(year)
backgroundProcess.list.get(i).getHour()
hourComboBox.setSelectedIndex(hour)
backgroundProcess.list.get(i).getMinute()
minuteComboBox.setSelectedIndex(minute)
backgroundProcess.list.get(i).getPriority()
priorityComboBox.setSelectedIndex(priority)
backgroundProcess.list.get(i).getPopup()
allowPopupCheckBox.setSelected(true)
editList()

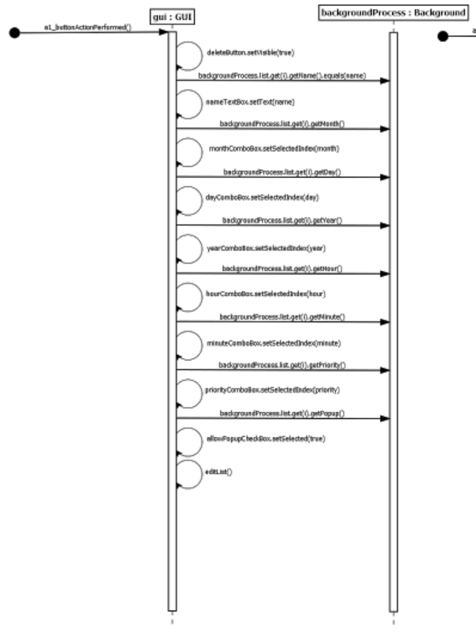

gui : GUI
backgroundProcess : Background
a2_buttonActionPerformed()
deleteButton.setVisible(true)
backgroundProcess.list.get(i).getName().equals(name)
nameTextBox.setText(name)
backgroundProcess.list.get(i).getMonth()
monthComboBox.setSelectedIndex(month)
backgroundProcess.list.get(i).getDay()
dayComboBox.setSelectedIndex(day)
backgroundProcess.list.get(i).getYear()
yearComboBox.setSelectedIndex(year)
backgroundProcess.list.get(i).getHour()
hourComboBox.setSelectedIndex(hour)
backgroundProcess.list.get(i).getMinute()
minuteComboBox.setSelectedIndex(minute)
backgroundProcess.list.get(i).getPriority()
priorityComboBox.setSelectedIndex(priority)
backgroundProcess.list.get(i).getPopup()
allowPopupCheckBox.setSelected(true)
editList()

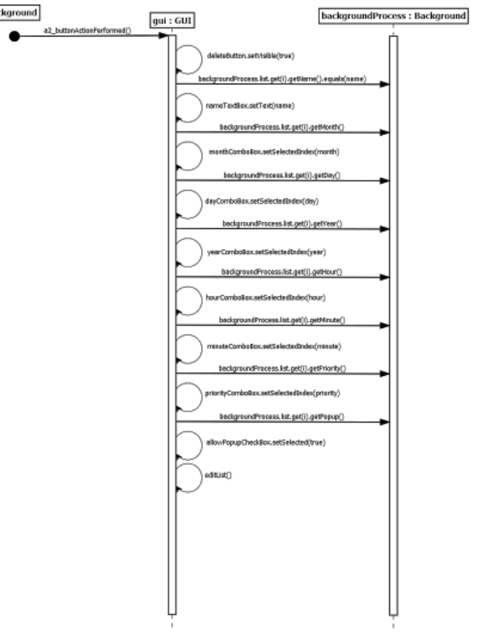

gui : GUI
backgroundProcess : Background
a3_buttonActionPerformed()
deleteButton.setVisible(true)
backgroundProcess.list.get(i).getName().equals(name)
nameTextBox.setText(name)
backgroundProcess.list.get(i).getMonth()
monthComboBox.setSelectedIndex(month)
backgroundProcess.list.get(i).getDay()
dayComboBox.setSelectedIndex(day)
backgroundProcess.list.get(i).getYear()
yearComboBox.setSelectedIndex(year)
backgroundProcess.list.get(i).getHour()
hourComboBox.setSelectedIndex(hour)
backgroundProcess.list.get(i).getMinute()
minuteComboBox.setSelectedIndex(minute)
backgroundProcess.list.get(i).getPriority()
priorityComboBox.setSelectedIndex(priority)
backgroundProcess.list.get(i).getPopup()
allowPopupCheckBox.setSelected(true)
editList()

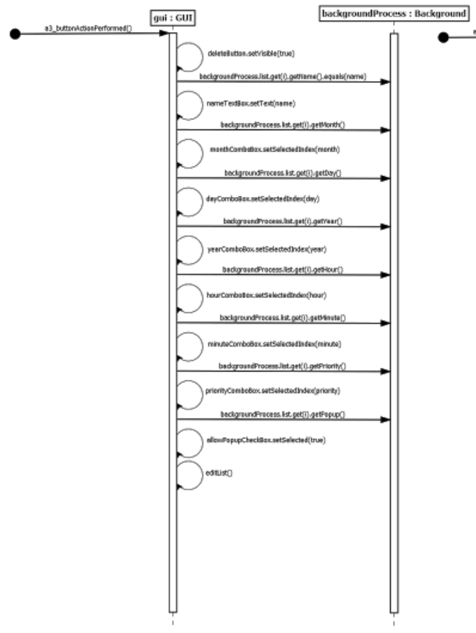

gui : GUI
backgroundProcess : Background
a4_buttonActionPerformed()
deleteButton.setVisible(true)
backgroundProcess.list.get(i).getName().equals(name)
nameTextBox.setText(name)
backgroundProcess.list.get(i).getMonth()
monthComboBox.setSelectedIndex(month)
backgroundProcess.list.get(i).getDay()
dayComboBox.setSelectedIndex(day)
backgroundProcess.list.get(i).getYear()
yearComboBox.setSelectedIndex(year)
backgroundProcess.list.get(i).getHour()
hourComboBox.setSelectedIndex(hour)
backgroundProcess.list.get(i).getMinute()
minuteComboBox.setSelectedIndex(minute)
backgroundProcess.list.get(i).getPriority()
priorityComboBox.setSelectedIndex(priority)
backgroundProcess.list.get(i).getPopup()
allowPopupCheckBox.setSelected(true)
editList()

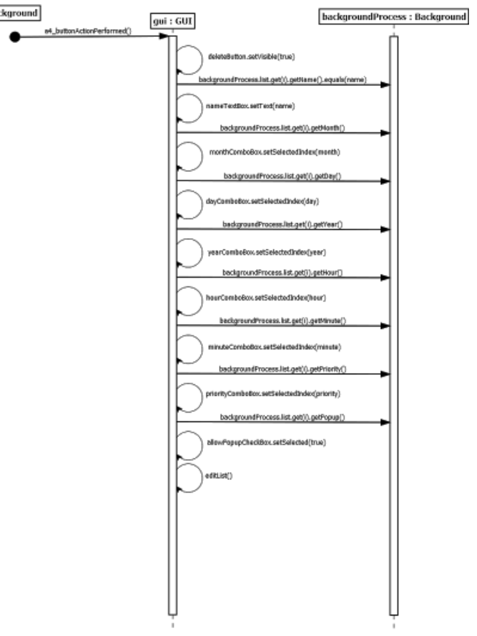

gui : GUI
backgroundProcess : Background
a5_buttonActionPerformed()
deleteButton.setVisible(true)
backgroundProcess.list.get(i).getName().equals(name)
nameTextBox.setText(name)
backgroundProcess.list.get(i).getMonth()
monthComboBox.setSelectedIndex(month)
backgroundProcess.list.get(i).getDay()
dayComboBox.setSelectedIndex(day)
backgroundProcess.list.get(i).getYear()
yearComboBox.setSelectedIndex(year)
backgroundProcess.list.get(i).getHour()
hourComboBox.setSelectedIndex(hour)
backgroundProcess.list.get(i).getMinute()
minuteComboBox.setSelectedIndex(minute)
backgroundProcess.list.get(i).getPriority()
priorityComboBox.setSelectedIndex(priority)
backgroundProcess.list.get(i).getPopup()
allowPopupCheckBox.setSelected(true)
editList()

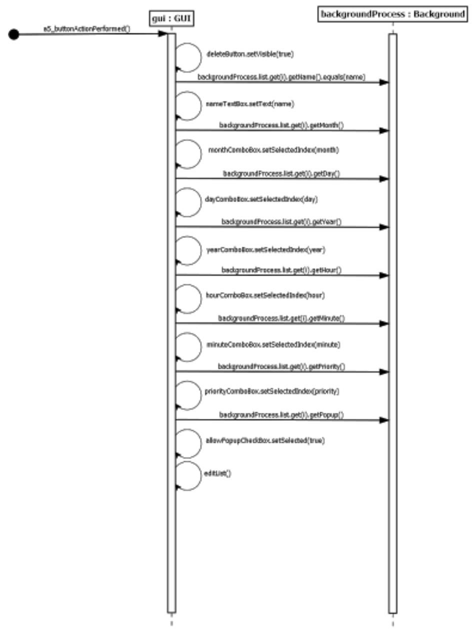
Use Case 7 : Set Priority


gui : GUI
backgroundProcess : Background
a1_buttonActionPerformed()
deleteButton.setVisible(true)
backgroundProcess.list.get(i).getName().equals(name)
nameTextBox.setText(name)
backgroundProcess.list.get(i).getMonth()
monthComboBox.setSelectedIndex(month)
backgroundProcess.list.get(i).getDay()
dayComboBox.setSelectedIndex(day)
backgroundProcess.list.get(i).getYear()
yearComboBox.setSelectedIndex(year)
backgroundProcess.list.get(i).getHour()
hourComboBox.setSelectedIndex(hour)
backgroundProcess.list.get(i).getMinute()
minuteComboBox.setSelectedIndex(minute)
backgroundProcess.list.get(i).getPriority()
priorityComboBox.setSelectedIndex(priority)
backgroundProcess.list.get(i).getPopup()
allowPopupCheckBox.setSelected(true)
editList()

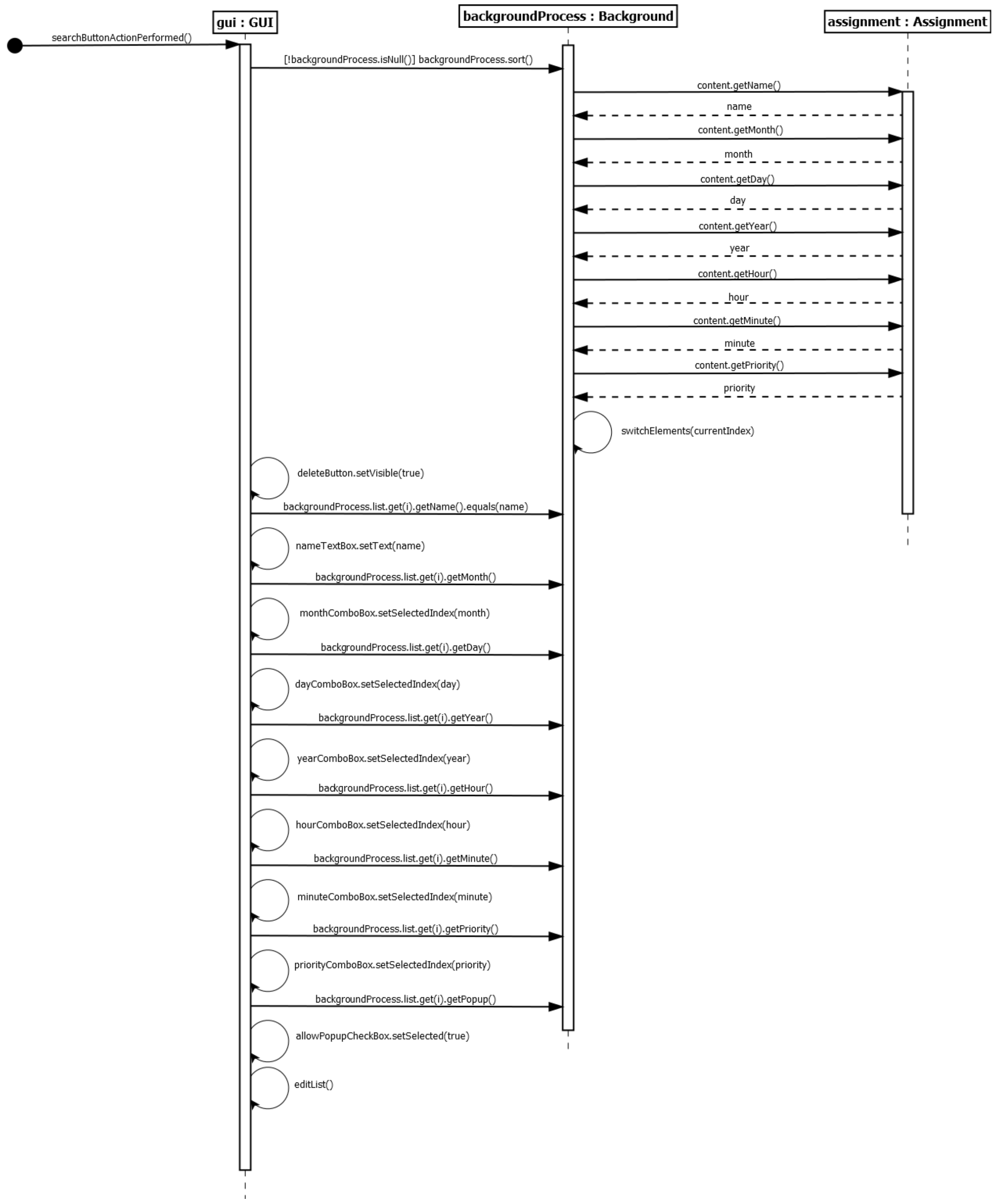

gui : GUI
backgroundProcess : Background
a2_buttonActionPerformed()
deleteButton.setVisible(true)
backgroundProcess.list.get(i).getName().equals(name)
nameTextBox.setText(name)
backgroundProcess.list.get(i).getMonth()
monthComboBox.setSelectedIndex(month)
backgroundProcess.list.get(i).getDay()
dayComboBox.setSelectedIndex(day)
backgroundProcess.list.get(i).getYear()
yearComboBox.setSelectedIndex(year)
backgroundProcess.list.get(i).getHour()
hourComboBox.setSelectedIndex(hour)
backgroundProcess.list.get(i).getMinute()
minuteComboBox.setSelectedIndex(minute)
backgroundProcess.list.get(i).getPriority()
priorityComboBox.setSelectedIndex(priority)
backgroundProcess.list.get(i).getPopup()
allowPopupCheckBox.setSelected(true)
editList()


gui : GUI
backgroundProcess : Background
a3_buttonActionPerformed()
deleteButton.setVisible(true)
backgroundProcess.list.get(i).getName().equals(name)
nameTextBox.setText(name)
backgroundProcess.list.get(i).getMonth()
monthComboBox.setSelectedIndex(month)
backgroundProcess.list.get(i).getDay()
dayComboBox.setSelectedIndex(day)
backgroundProcess.list.get(i).getYear()
yearComboBox.setSelectedIndex(year)
backgroundProcess.list.get(i).getHour()
hourComboBox.setSelectedIndex(hour)
backgroundProcess.list.get(i).getMinute()
minuteComboBox.setSelectedIndex(minute)
backgroundProcess.list.get(i).getPriority()
priorityComboBox.setSelectedIndex(priority)
backgroundProcess.list.get(i).getPopup()
allowPopupCheckBox.setSelected(true)
editList()

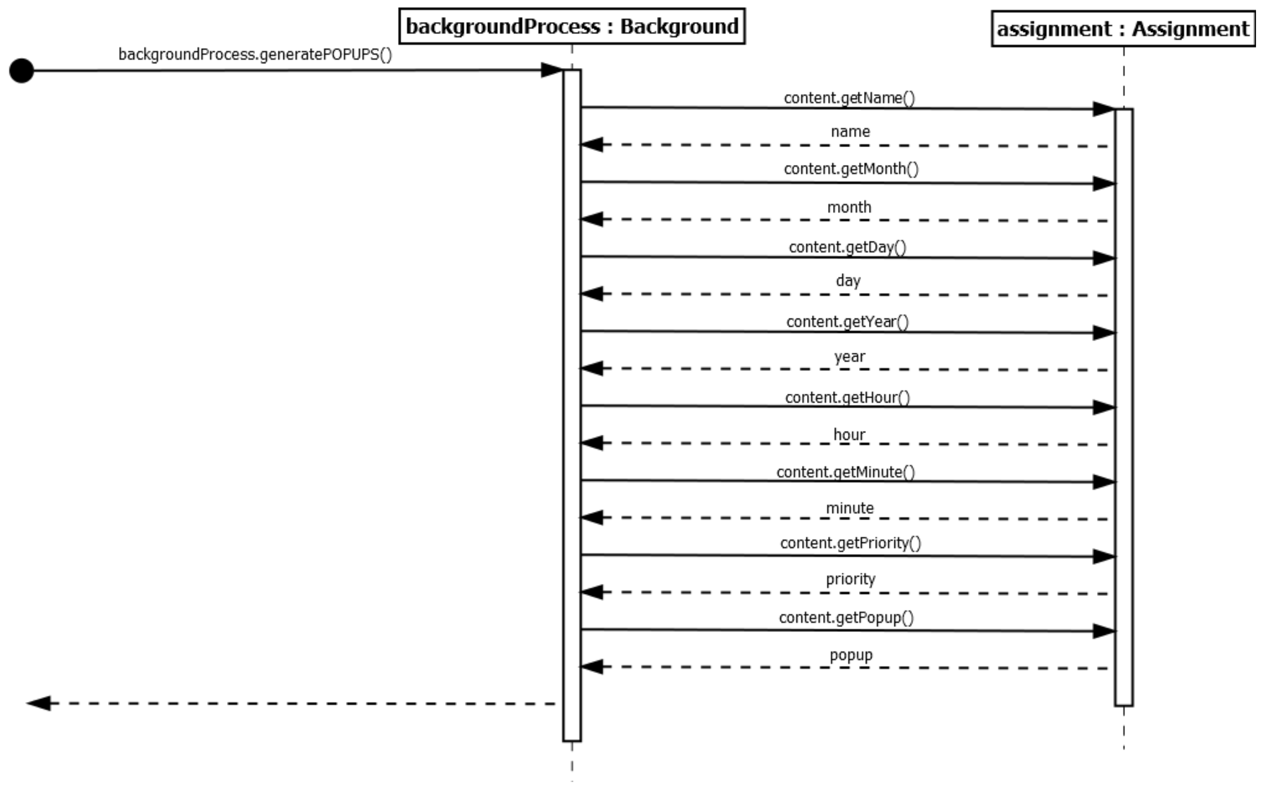

gui : GUI
backgroundProcess : Background
a4_buttonActionPerformed()
deleteButton.setVisible(true)
backgroundProcess.list.get(i).getName().equals(name)
nameTextBox.setText(name)
backgroundProcess.list.get(i).getMonth()
monthComboBox.setSelectedIndex(month)
backgroundProcess.list.get(i).getDay()
dayComboBox.setSelectedIndex(day)
backgroundProcess.list.get(i).getYear()
yearComboBox.setSelectedIndex(year)
backgroundProcess.list.get(i).getHour()
hourComboBox.setSelectedIndex(hour)
backgroundProcess.list.get(i).getMinute()
minuteComboBox.setSelectedIndex(minute)
backgroundProcess.list.get(i).getPriority()
priorityComboBox.setSelectedIndex(priority)
backgroundProcess.list.get(i).getPopup()
allowPopupCheckBox.setSelected(true)
editList()


gui : GUI
backgroundProcess : Background
a5_buttonActionPerformed()
deleteButton.setVisible(true)
backgroundProcess.list.get(i).getName().equals(name)
nameTextBox.setText(name)
backgroundProcess.list.get(i).getMonth()
monthComboBox.setSelectedIndex(month)
backgroundProcess.list.get(i).getDay()
dayComboBox.setSelectedIndex(day)
backgroundProcess.list.get(i).getYear()
yearComboBox.setSelectedIndex(year)
backgroundProcess.list.get(i).getHour()
hourComboBox.setSelectedIndex(hour)
backgroundProcess.list.get(i).getMinute()
minuteComboBox.setSelectedIndex(minute)
backgroundProcess.list.get(i).getPriority()
priorityComboBox.setSelectedIndex(priority)
backgroundProcess.list.get(i).getPopup()
allowPopupCheckBox.setSelected(true)
editList()

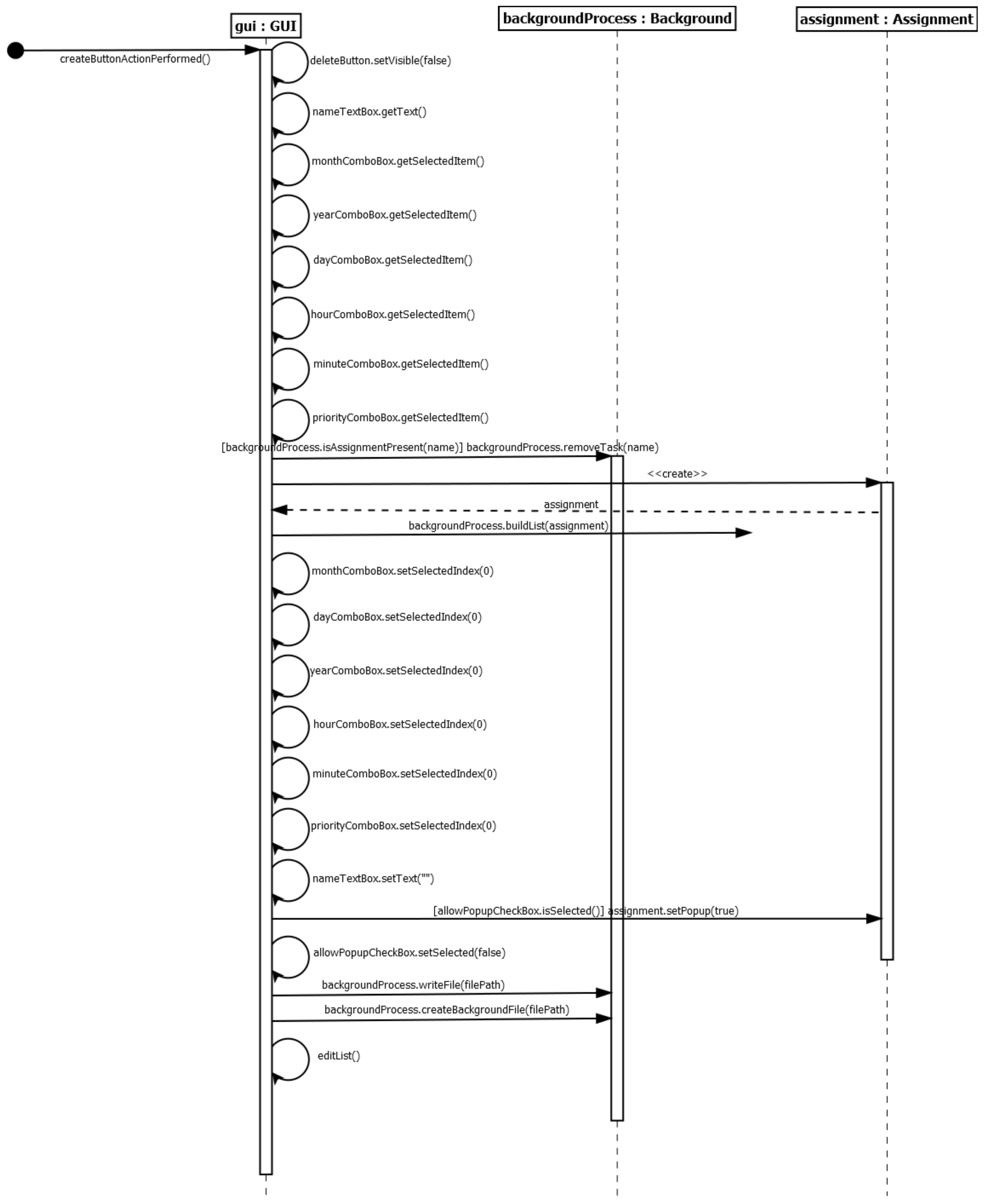
Use Case 8 : Sort List File


gui : GUI
backgroundProcess : Background
assignment : Assignment
searchButtonActionPerformed()
[!backgroundProcess.isNull()] backgroundProcess.sort()
content.getName()
name
content.getMonth()
month
content.getDay()
day
content.getYear()
year
content.getHour()
hour
content.getMinute()
minute
content.getPriority()
priority
switchElements(currentIndex)
deleteButton.setVisible(true)
backgroundProcess.list.get(i).getName().equals(name)
nameTextBox.setText(name)
backgroundProcess.list.get(i).getMonth()
monthComboBox.setSelectedIndex(month)
backgroundProcess.list.get(i).getDay()
dayComboBox.setSelectedIndex(day)
backgroundProcess.list.get(i).getYear()
yearComboBox.setSelectedIndex(year)
backgroundProcess.list.get(i).getHour()
hourComboBox.setSelectedIndex(hour)
backgroundProcess.list.get(i).getMinute()
minuteComboBox.setSelectedIndex(minute)
backgroundProcess.list.get(i).getPriority()
priorityComboBox.setSelectedIndex(priority)
backgroundProcess.list.get(i).getPopup()
allowPopupCheckBox.setSelected(true)
editList()

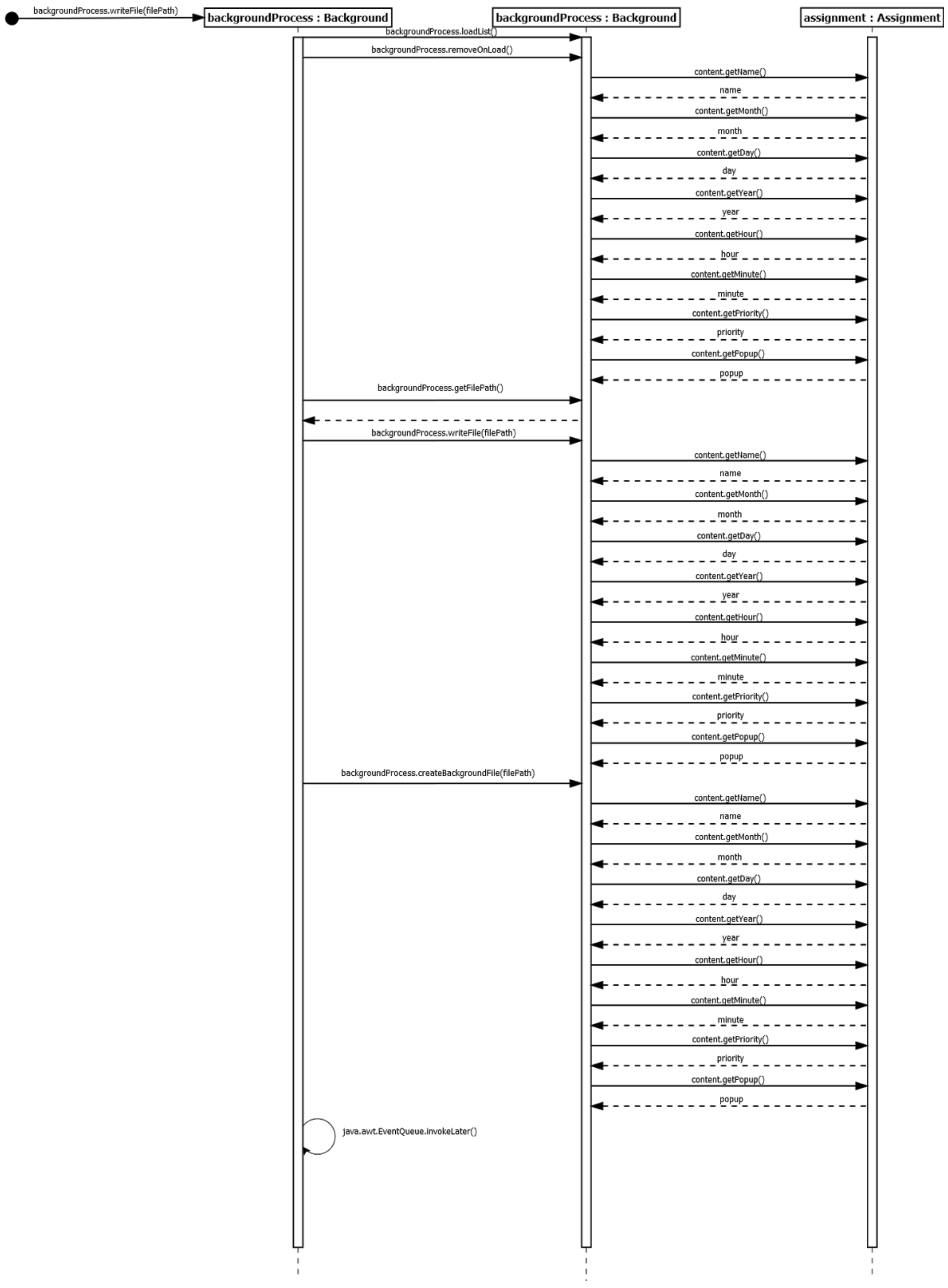
Use Case 9 : Notifications



Use Case 10 : Check-Box


gui : GUI
backgroundProcess : Background
assignment : Assignment
createButtonActionPerformed()
deleteButton.setVisible(false)
nameTextBox.getText()
monthComboBox.getSelectedItem()
yearComboBox.getSelectedItem()
dayComboBox.getSelectedItem()
hourComboBox.getSelectedItem()
minuteComboBox.getSelectedItem()
priorityComboBox.getSelectedItem()
[backgroundProcess.isAssignmentPresent(name)] backgroundProcess.removeTask(name)
<<create>>
assignment
backgroundProcess.buildList(assignment)
monthComboBox.setSelectedIndex(0)
dayComboBox.setSelectedIndex(0)
yearComboBox.setSelectedIndex(0)
hourComboBox.setSelectedIndex(0)
minuteComboBox.setSelectedIndex(0)
priorityComboBox.setSelectedIndex(0)
nameTextBox.setText("")
[allowPopupCheckBox.isSelected()] assignment.setPopup(true)
allowPopupCheckBox.setSelected(false)
backgroundProcess.writeFile(filePath)
backgroundProcess.createBackgroundFile(filePath)
editList()

Use Case 11 : GUI


backgroundProcess.writeFile(filePath)
backgroundProcess : Background
backgroundProcess : Background
assignment : Assignment
backgroundProcess.loadList()
backgroundProcess.removeOnLoad()
content.getName()
name
content.getMonth()
month
content.getDay()
day
content.getYear()
year
content.getHour()
hour
content.getMinute()
minute
content.getPriority()
priority
content.getPopup()
popup
backgroundProcess.getFilePath()
backgroundProcess.writeFile(filePath)
content.getName()
name
content.getMonth()
month
content.getDay()
day
content.getYear()
year
content.getHour()
hour
content.getMinute()
minute
content.getPriority()
priority
content.getPopup()
popup
backgroundProcess.createBackgroundFile(filePath)
content.getName()
name
content.getMonth()
month
content.getDay()
day
content.getYear()
year
content.getHour()
hour
content.getMinute()
minute
content.getPriority()
priority
content.getPopup()
popup
java.awt.EventQueue.invokeLater()