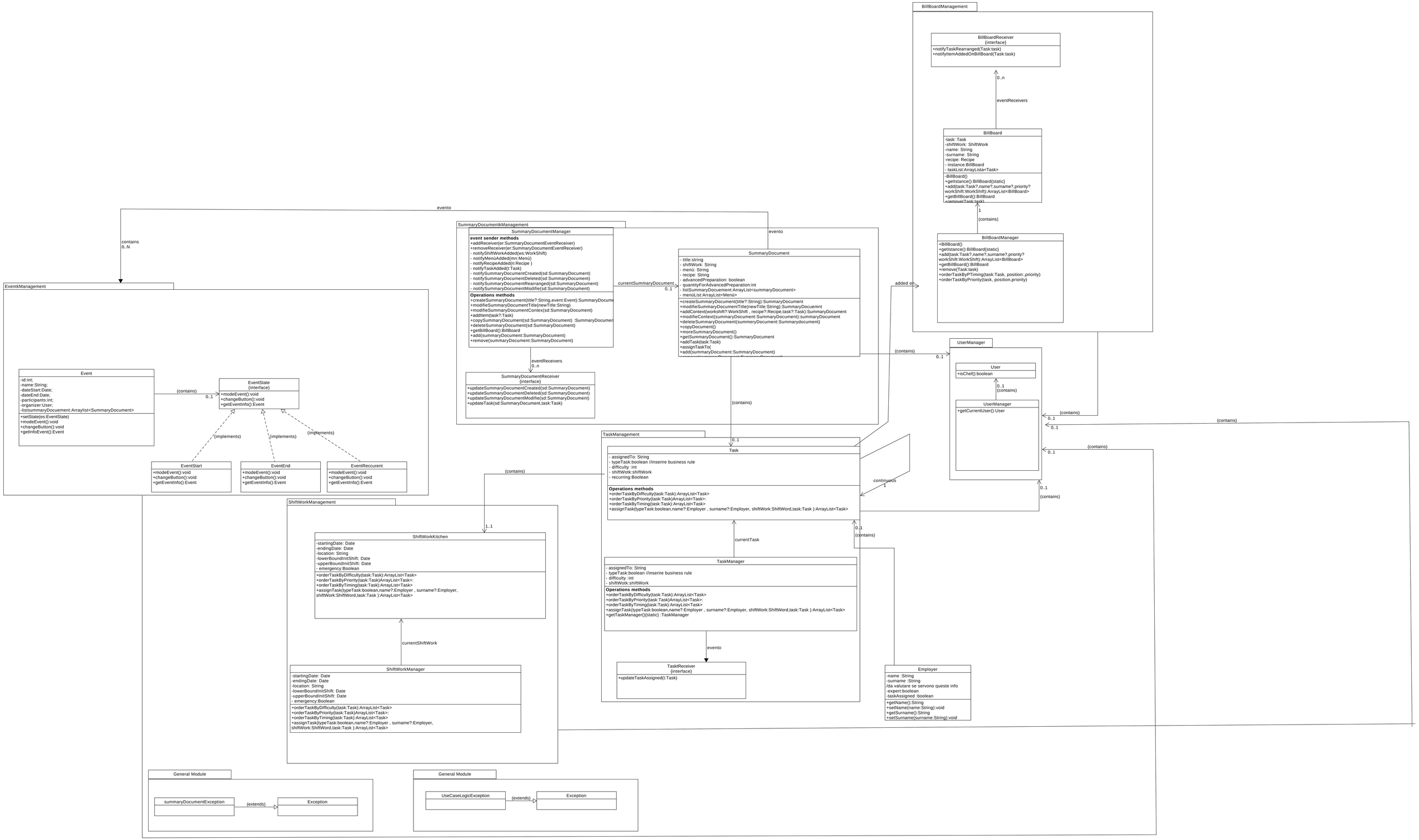
BillBoardManagement
BillBoardReceiver
{interface}
+notifyTaskRearranged(Task:task)
+notifyItemAddedOnBillBoard(Task:task)
0..n
eventReceivers
BillBoard
-task: Task
-shiftWork: ShiftWork
-name: String
-surname: String
-recipe: Recipe
- instance:BillBoard
- taskList:ArrayLista<Task>
-BillBoard()
+getIstance():BillBoard{static}
+add(task:Task?,name?,surname?,priority?
workShift:WorkShift):ArrayList<BillBoard>
+getBillBoard():BillBoard
+remove(Task:task)
1
(contains)
BillBoardManager
+BillBoard()
+getIstance():BillBoard{static}
+add(task:Task?,name?,surname?,priority?
workShift:WorkShift):ArrayList<BillBoard>
+getBillBoard():BillBoard
+remove(Task:task)
+orderTaskByPTiming(task:Task, position:,priority)
+orderTaskByPriority(task, position,priority)
evento
contains
0..N
EventkManagement
Event
-id:int;
-name:String;
-dateStart:Date;
-dateEnd:Date;
-participants:int;
-organizer:User;
-listsummaryDocuement:Arraylist<SummaryDocument>
+setState(es:EventState)
+modeEvent():void
+changeButton():void
+getInfoEvent():Event
(contains)
0..1
EventState
{interface}
+modeEvent():void
+changeButton():void
+getEventInfo():Event
(implements)
(implements)
(implements)
EventStart
+modeEvent():void
+changeButton():void
+getEventInfo():Event
EventEnd
+modeEvent():void
+changeButton():void
+getEventInfo():Event
EventReccurent
+modeEvent():void
+changeButton():void
+getEventInfo():Event
SummaryDocumentkManagement
SummaryDocumentManager
event sender methods
+addReceiver(er:SummaryDocumentEventReceiver)
+removeReceiver(er:SummaryDocumentEventReceiver)
- notifyShiftWorkAdded(ws:WorkShift)
- notifyMenùAdded(mn:Menù)
- notifyRecipeAdded(ri:Recipe)
- notifyTaskAdded(t:Task)
- notifySummaryDocumentCreated(sd:SummaryDocument)
- notifySummaryDocumentDeleted(sd:SummaryDocument)
- notifySummaryDocumentRearranged(sd:SummaryDocument)
- notifySummaryDocumentModifie(sd:SummaryDocument)
Operations methods
+createSummaryDocument(title?:String,event:Event):SummaryDocume
+modifieSummaryDocumentTitle(newTitle:String)
+modifieSummaryDocumentContex(sd:SummaryDocument)
+addItem(task?:Task)
+copySummaryDocument(sd:SummaryDocument) :SummaryDocumen
+deleteSummaryDocument(sd:SummaryDocument)
+getBillBoard():BillBoard
+add(summaryDocument:SummaryDocument)
+remove(summaryDocument:SummaryDocument)
currentSummaryDocument
0..1
SummaryDocument
- title:string
- shiftWork: String
- menù: String
- recipe: String
- advancedPreparation: boolean
- quantityForAdvancedPreparation:int
- listSummaryDocuement:ArrayList<summaryDocument>
- menùList:ArrayList<Menù>
+createSummaryDocument(title?:String):SummaryDocument
+modifieSummaryDocumentTitle(newTitle:String):SummaryDocuemnt
+addContext(workshift?:WorkShift , recipe?:Recipe,task?:Task):SummaryDocument
+modifieContext(summaryDocument:SummaryDocument):summaryDocument
+deleteSummaryDocument(summeryDocument:Summarydocument)
+copyDocument()
+moreSummaryDocument()
+getSummaryDocument():SummaryDocument
+addTask(task:Task)
+assignTaskTo(
+add(summaryDocument:SummaryDocument)
eventReceivers
0..n
SummaryDocumentReceiver
{interface}
+updateSummaryDocumentCreated(sd:SummaryDocument)
+updateSummaryDocumentDeleted(sd:SummaryDocument)
+updateSummaryDocumentModifie(sd:SummaryDocument)
+updateTask(sd:SummaryDocument,task:Task)
added on
(contains)
0..1
UserManager
User
+isChef():boolean
0..1
(contains)
UserManager
+getCurrentUser():User
(contains)
0..1
(contains)
0..1
(contains)
0..1
0..1
(contains)
TaskManagement
0..1
Task
- assignedTo: String
- typeTask:boolean //inserire business rule
- difficulty :int
- shiftWotk:shiftWork
- recurring:Boolean
Operations methods
+orderTaskByDifficulty(task:Task):ArrayList<Task>
+orderTaskByPriority(task:Task)ArrayList<Task>:
+orderTaskByTiming(task:Task):ArrayList<Task>
+assignTask(typeTask:boolean,name?:Employer , surname?:Employer, shiftWork:ShiftWord,task:Task):ArrayList<Task>
continuous
1
(contains)
1..1
ShiftWorkManagement
ShiftWorkKitchen
-startingDate: Date
-endingDate: Date
-location: String
-lowerBoundInitShift: Date
-upperBoundInitShift: Date
- emergency:Boolean
+orderTaskByDifficulty(task:Task):ArrayList<Task>
+orderTaskByPriority(task:Task)ArrayList<Task>:
+orderTaskByTiming(task:Task):ArrayList<Task>
+assignTask(typeTask:boolean,name?:Employer , surname?:Employer,
shiftWork:ShiftWord,task:Task):ArrayList<Task>
currentShiftWork
ShiftWorkManager
-startingDate: Date
-endingDate: Date
-location: String
-lowerBoundInitShift: Date
-upperBoundInitShift: Date
- emergency:Boolean
+orderTaskByDifficulty(task:Task):ArrayList<Task>
+orderTaskByPriority(task:Task)ArrayList<Task>:
+orderTaskByTiming(task:Task):ArrayList<Task>
+assignTask(typeTask:boolean,name?:Employer , surname?:Employer,
shiftWork:ShiftWord,task:Task):ArrayList<Task>
currentTask
0..1
(contains)
TaskManager
- assignedTo: String
- typeTask:boolean //inserire business rule
- difficulty :int
- shiftWotk:shiftWork
Operations methods
+orderTaskByDifficulty(task:Task):ArrayList<Task>
+orderTaskByPriority(task:Task)ArrayList<Task>:
+orderTaskByTiming(task:Task):ArrayList<Task>
+assignTask(typeTask:boolean,name?:Employer , surname?:Employer, shiftWork:ShiftWord,task:Task):ArrayList<Task>
+getTaskManager()[static] :TaskManager
evento
TasktReceiver
{interface}
+updateTaskAssigned(t:Task)
Employer
-name :String
-surname :String
/da valutare se servono queste info
-expert:boolean
-taskAssigned :boolean
+getName():String
+setName(name:String):void
+getSurname():String
+setSurname(surname:String):void
General Module
summaryDocumentException
(extends)
Exception
General Module
UseCaseLogicException
(extends)
Exception