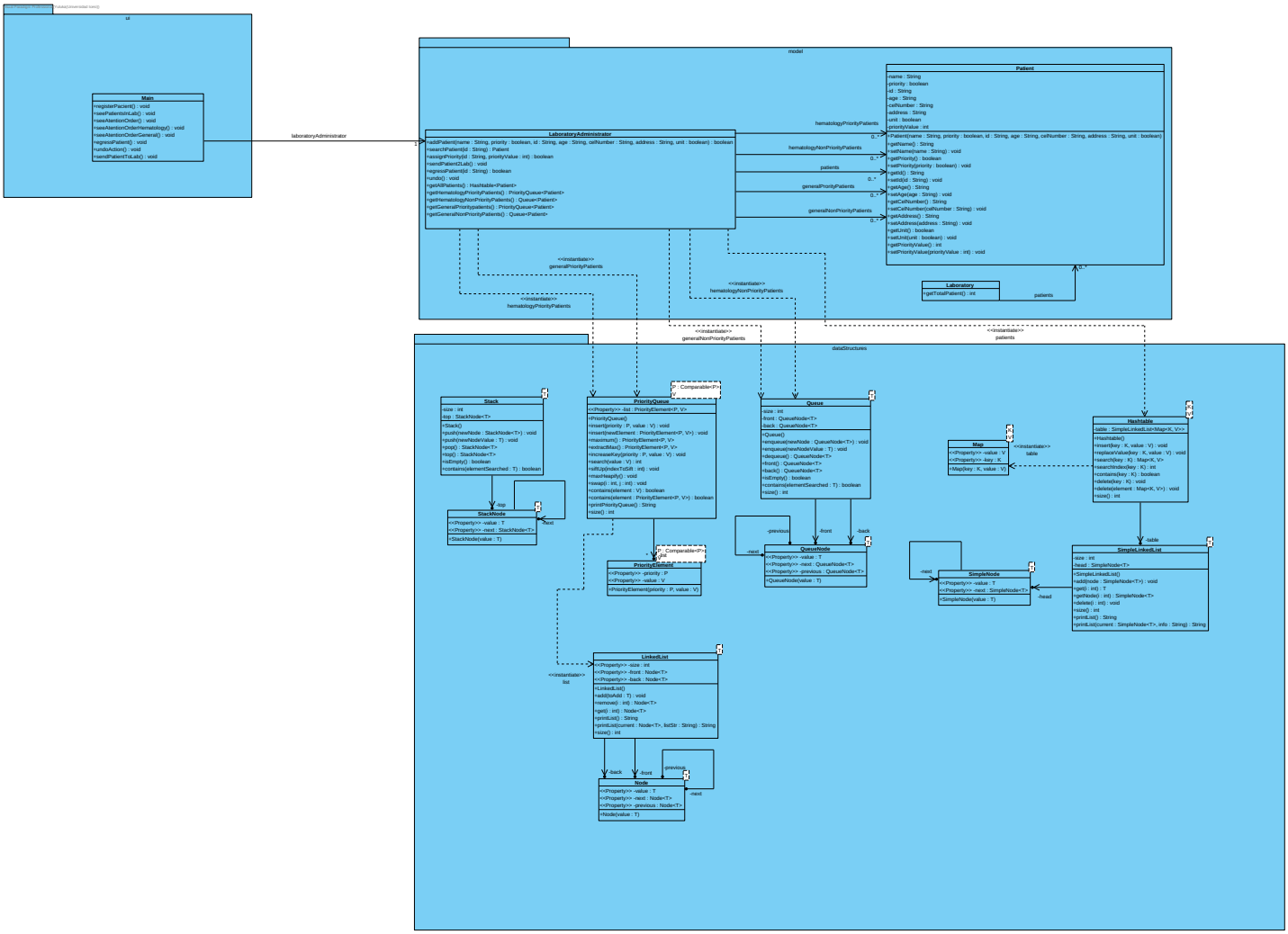
ui
Main
+registerPacient() : void
+seePatientsInLab() : void
+seeAttentionOrder() : void
+seeAttentionOrderHematology() : void
+seeAttentionOrderGeneral() : void
+egresoPatient() : void
+undoAction() : void
+sendPatientToLab() : void
laboratoryAdministrator
model
LaboratoryAdministrator
+addPatient(name : String, priority : boolean, id : String, age : String, cellNumber : String, address : String, unit : boolean) : boolean
+searchPatientId(id : String) : Patient
+assignPriority(id : String, priorityValue : int) : boolean
+sendPatient2Lab() : void
+egresoPatientId(id : String) : boolean
+undo() : void
+getAllPatients() : Hashtable<Patient>
+getHematologyPriorityPatients() : PriorityQueue<Patient>
+getHematologyNonPriorityPatients() : Queue<Patient>
+getGeneralPrioritypatients() : PriorityQueue<Patient>
+getGeneralNonPriorityPatients() : Queue<Patient>
hematologyPriorityPatients
hematologyNonPriorityPatients
patients
generalPriorityPatients
generalNonPriorityPatients
0..*
Patient
-name : String
-priority : boolean
-id : String
-age : String
-cellNumber : String
-address : String
-unit : boolean
-priorityValue : int
+Patient(name : String, priority : boolean, id : String, age : String, cellNumber : String, address : String, unit : boolean)
+getName() : String
+setName(name : String) : void
+getPriority() : boolean
+setPriority(priority : boolean) : void
+getId() : String
+setId(id : String) : void
+getAge() : String
+setAge(age : String) : void
+getCellNumber() : String
+setCellNumber(cellNumber : String) : void
+getAddress() : String
+setAddress(address : String) : void
+getUnit() : boolean
+setUnit(unit : boolean) : void
+getPriorityValue() : int
+setPriorityValue(priorityValue : int) : void
Laboratory
+getTotalPatient() : int
patients
<<instantiate>>
generalPriorityPatients
<<instantiate>>
hematologyPriorityPatients
<<instantiate>>
hematologyNonPriorityPatients
<<instantiate>>
generalNonPriorityPatients
<<instantiate>>
patients
dataStructures
Stack
-size : int
-top : StackNode<T>
+Stack()
+push(newNode : StackNode<T>) : void
+push(newNodeValue : T) : void
+pop() : StackNode<T>
+top() : StackNode<T>
+isEmpty() : boolean
+contains(elementSearched : T) : boolean
-top
StackNode
<<Property>> -value : T
<<Property>> -next : StackNode<T>
+StackNode(value : T)
-next
PriorityQueue
P : Comparable<P>
V
<<Property>> -list : PriorityElement<P, V>
+PriorityQueue()
+insert(priority : P, value : V) : void
+insert(newElement : PriorityElement<P, V>) : void
+maximum() : PriorityElement<P, V>
+extractMax() : PriorityElement<P, V>
+increaseKey(priority : P, value : V) : void
+search(value : V) : int
+isEqNodeToEdit : int) : void
+maxHeapify() : void
+swap() : int, j : int) : void
+contains(element : V) : boolean
+contains(element : PriorityElement<P, V>) : boolean
+printPriorityQueue() : String
+size() : int
PriorityElement
P : Comparable<P>
V
<<Property>> -priority : P
<<Property>> -value : V
+PriorityElement(priority : P, value : V)
Queue
-size : int
-front : QueueNode<T>
-back : QueueNode<T>
+Queue()
+enqueue(newNode : QueueNode<T>) : void
+enqueue(newNodeValue : T) : void
+dequeue() : QueueNode<T>
+front() : QueueNode<T>
+back() : QueueNode<T>
+isEmpty() : boolean
+contains(elementSearched : T) : boolean
+size() : int
-previous
-front
-back
-next
QueueNode
<<Property>> -value : T
<<Property>> -next : QueueNode<T>
<<Property>> -previous : QueueNode<T>
+QueueNode(value : T)
Map
K
V
<<Property>> -value : V
<<Property>> -key : K
+Map(key : K, value : V)
<<instantiate>>
table
Hashtable
K
V
-table : SimpleLinkedList<Map<K, V>>
+Hashtable()
+insert(key : K, value : V) : void
+replace/Value(key : K, value : V) : void
+search(key : K) : Map<K, V>
+searchIndex(key : K) : int
+contains(key : K) : boolean
+delete(key : K) : void
+delete(element : Map<K, V>) : void
+size() : int
-table
-next
SimpleNode
<<Property>> -value : T
<<Property>> -next : SimpleNode<T>
+SimpleNode(value : T)
-head
SimpleLinkedList
-size : int
-head : SimpleNode<T>
+SimpleLinkedList()
+add(node : SimpleNode<T>) : void
+get() : int) : T
+getNode() : int) : SimpleNode<T>
+delete() : int) : void
+size() : int
+printList() : String
+printList(current : SimpleNode<T>, info : String) : String
<<instantiate>>
list
LinkedList
<<Property>> -size : int
<<Property>> -front : Node<T>
<<Property>> -back : Node<T>
+LinkedList()
+add(toAdd : T) : void
+remove() : int) : Node<T>
+get() : int) : Node<T>
+printList() : String
+printList(current : Node<T>, listStr : String) : String
+size() : int
-back
-front
-previous
-next
Node
<<Property>> -value : T
<<Property>> -next : Node<T>
<<Property>> -previous : Node<T>
+Node(value : T)