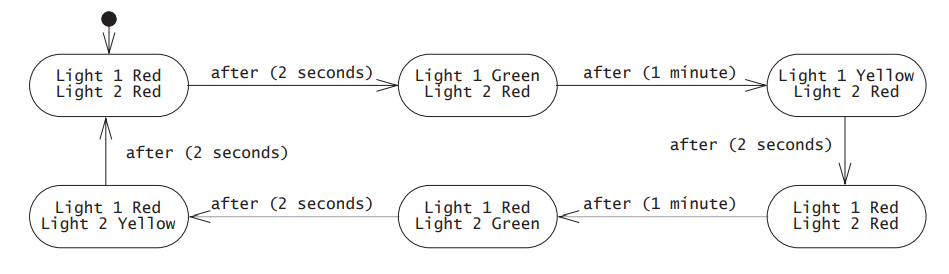
**Assignment 5**

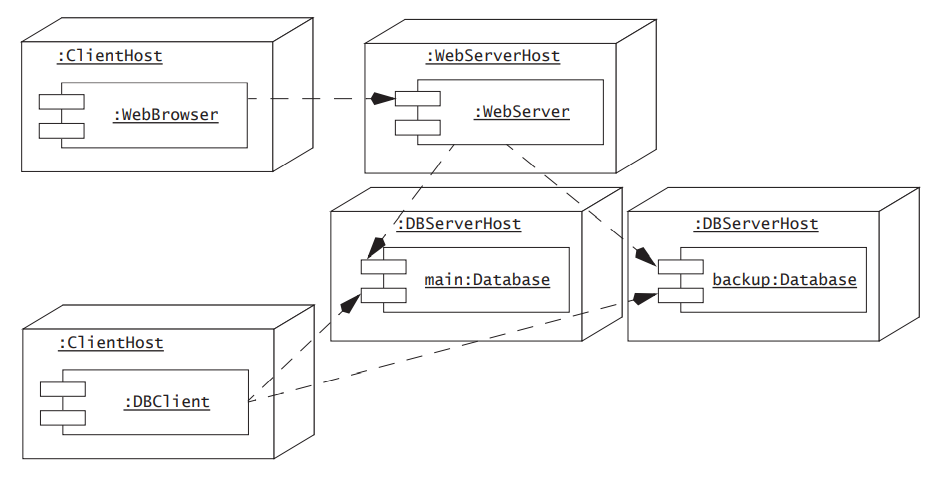


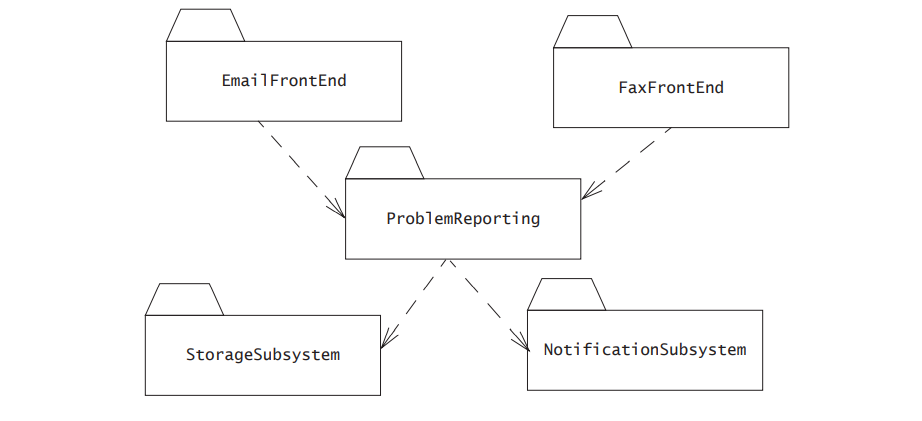


***A SIMPLE DIGITAL WATCH***  
1).The digital watch has two buttons A and B  
2).A selects the modes once pressed advances to next mode in sequence.  
3).B advances the hours,minutes once each time it is pressed.

Diagram

Description automatically generated







* int size() returns the number of elements in the list.

**context** List::size() **post**: result = elements->size

* void add(Object e) adds an object at the end of the list.

**context** List::add(e) **post**: contains(e) and size() = @pre.size() + 1

* void remove(Object e) removes an object from the end of the list.

**context** List::remove(e) **post**:

(@pre.contains(e) implies size() = @pre.size() - 1) and

(not @pre.contains(e) implies size() = @pre.size()) and

result = @pre.contains(e)

* boolean contains(Object e) returns true if the object is contained in the list.

**context** List::contains(e) **post**: result = elements->includes(e)

* Object get(int idx) returns the object located at index idx, 0 being the index of the first object in the list.

**context** List::get(idx) **pre**: idx >= 0 and idx < size()

**context** List::get(idx) **post**: result = elements->at(idx+1)



* int size() returns the number of elements in the set.

**context** Set::size() **post**: result = elements->size()

* void add(Object e) adds an object to the set. If the object is already in the set, does nothing.

**context** Set::add(e) **post**:

contains(e) and

(@pre.contains(e) implies size() = @pre.size()) and

(not @pre.contains(e) implies size() = @pre.size()+1)

* void remove(Object e) removes an object from the set

**context** Set::remove(e) **post**:

not contains(e) and

(@pre.contains(e) implies size() = @pre.size() - 1) and

(not @pre.contains(e) implies size() = @pre.size())

* boolean contains(Object e) returns true if the object is contained in the set.

**context** Set::contains(e) **post**: elements->includes(e)