If a glass break detector attached to the entrance door detects that the entrance door has been damaged, the system shall enter the alarm state and inform the security company.
Exerise: For the given requirement statement (in natural language) create three interrelated representations to capture:

- **Data** (or static or conceptual) model using UML **class** diagrams
- **Behavioural** model using UML **state** diagrams
- **Functional** model using UML **sequence** diagrams
Static model

GlassBreakDetector
- detects ()
- changesStatus *

EntranceDoor
- status: DoorStatus

System
- mode: AlarmMode
- putAlarm ()

SecurityCompany

<<Enumeration>>
DoorStatus
- UNBROKEN
- DAMAGED

<<Enumeration>>
AlarmMode
- OFF
- ON_DUTY
- ALARM
Behavioural model

: System

mode: AlarmMode

putAlarm()
switchOn()
switchOff()
Behavioural model

:System
- mode: AlarmMode
- putAlarm()
- switchOn()
- switchOff()

:EntranceDoor
- status: DoorStatus
- break()
- fix()
Updated after **Behavioural analysis**

Static model:

- **GlassBreakDetector**
  - detects ()
  - changesStatus

- **System**
  - mode: AlarmMode
  - putAlarm()
  - switchOn()
  - switchOff()

- **EntranceDoor**
  - status: DoorStatus
  - break()
  - fix()

- **SecurityCompany**

<<Enumeration>>

- **DoorStatus**
  - UNBROKEN
  - DAMAGED

<<Enumeration>>

- **AlarmMode**
  - OFF
  - ON_DUTY
  - ALARM
Functional model

:GlassBreakDetector

[EntranceDoor.status=DAMAGED]
changesStatus()

:System

[mode=ALARM] informs()

:SecurityCompany

[System.mode=ALARM]
manages()

:EntranceDoor

[EntranceDoor.status=UNBROKEN]
manages()

repairs()
Static model: Updated after Functional analysis

GlassBreakDetector
-Detects()
-ChangesStatus

System
-Mode: AlarmMode
-PutAlarm()
-SwitchOn()
-SwitchOff()

EntranceDoor
-Status: DoorStatus
-Break()
-Fix()

SecurityCompany

<<Enumeration>>
DoorStatus
UNBROKEN
DAMAGED

<<Enumeration>>
AlarmMode
OFF
ON_DUTY
ALARM