The aim of this paper is to answer the questions that starts with why, such as “why was used this particular design pattern” or “why was the problem modeled and solved this way”. It has no pretentions to be an exhaustive or detailed commentary to the framework and his functionalities. For any further question we address the reader to the Javadoc.

## Resource Hierarchy package

The vast majority of classes contained in this package descend from the Resource super-class and implements all the resources available in the ED.

### Patient

A patient is characterized by: name, surname, unique ID, health insurance, severity level, current state, state history; the arrival time and a current location, are contained in the previous fields as clarified in this paragraph.

Name, surname and ID are indeed inherited by the Human Resource class.

The health insurance is defined in a nested-class which provides the insurance “name” or type (either no insurance, silver or gold) and the associated discount rate, as well as all the canonical methods. Pay attention to the fact that the class creator may rise an exception if used improperly, which is to say if is asked to create an insurance of unknown type.

The severity level, is defined in the homonymous nested class that is equipped with the “level” field and all the canonical methods. In particular, the default creator assigns a random level of severity as asked in the specification [TODO].

The current state is nothing more than an event, in the sense described in the “EventHierarchy” paragraph.

In this logic, the current location of the patient can be read in the place field of the state event, the history is the ordered list of old states and the arrival time is the time stamp of the first event in the history array.

The methods, besides the creators, getters and setters, that are detailed in the Javadoc, are the

### SeverityLevel

While is not a resource of the ED, but rather a patient field, I chose not to nest it in the patient class in order to make it accessible also from the event package.

### Health Service

This package also contains to interfaces: Location and ResourceManager.

Location is implemented by the resources where an event may take place: Room etc.

ResourceManager is implemented by the classes that can allocate and free resources.

## Event Hierarchy package

The vast majority of classes contained in this package descend from the Event super-class and implements all the events that can take place.

This package implements the simple factory pattern. An EventFactory is provided for the sake of encapsulation, so to make the code flexible in respect of the addition of new events.

## Probability distribution package

This package implements a strategy pattern to provide client applications with different probability distributions.

The superclass “ProbabilityDistribution” is the abstract strategy implemented by the concrete distributions such as uniform, exp, etc.

The concrete distributions shadows the name attribute with a private static final String variable where the name of the distribution is saved .

The concrete distributions also have to override the getName() and getSample() methods.

For probability distribution that are not implemented in the java.util.random library and has the analytical form of the cumulative distribution function the inverse transform sampling method is implemented. Those probability distribution, such as the exponential one, are equipped, than, with a inverse CDF function.

The package provide a simple Test Case Unit and the possibility to plot the distribution is left for future upgreades.

## Emergency department

### Patient Handler

Threds of this class should serve each patient