Comet Patient Monitor

Domain Model

Team Members:
Patrick Haring
Christian Bürgi

Client:
Prof. Dr. Olivier Biberstein

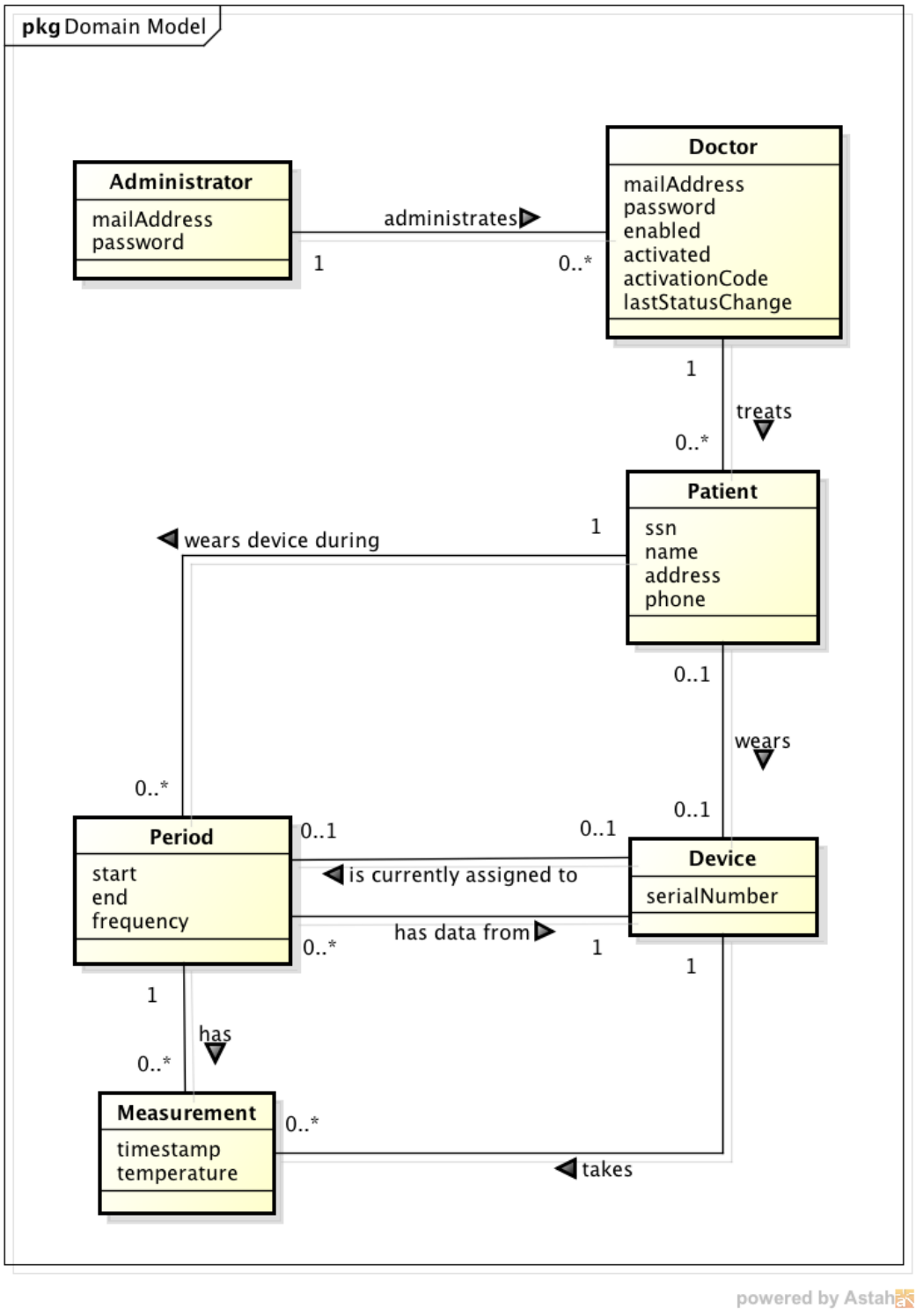
Revision: 98

May 4, 2012

https://svn.bfh.ch/repos/projects/patmon1/trunk/doc/src/domain_model.tex?p=98

Contents

1 Domain Model



2 Description

2.1 Classes

2.1.1 Administrator

The `Administrator` has the attributes `mailAddress` and `password` which are used as credentials to authenticate against the system.

2.1.2 Doctor

Each `Doctor` has an email address and a password. They are stored in the attributes `mailAddress` and `password`, which are used as credentials to authenticate against the system. `Enabled` indicates wether the account is enabled oder disabled. The attributes `activated` and `activationCode` are used for the activation process. `Activated` indicates whether the `Doctor` is activated or not, `activationCode` stores the activation code which was sent to him by email. For a successful login into the system, both `enabled` and `activated` have to be true. In order to delete an account, it has to be disabled for a given time period. Therefore a timestamp called `lastStatusChange` shows when the last status change occurred. A status change is either a change of `enabled` or `activated`.

2.1.3 Patient

Each `Patient` is identified by his social security number (SSN) which is stored the attribute `ssn`. The attributes `name`, `address` and `phone` are mandatory and will help the `Doctor` to identify the `Patient`.

2.1.4 Period

The class `Period` represents a monitoring period during which a `Device` is monitoring data on a `Patient`. The attributes `start` and `end` represent the interval of time and the attribute `frequency` represents the frequency by which the `Measurement`s are taken.

2.1.5 Device

Each `Device` is identified by its serial number, which is stored in the attribute `serialNumber`.

2.1.6 Measurement

The class `Measurement` represents the `temperature` which was taken by a `Device` at a given time `timestamp`.

2.2 Relationships

2.2.1 Administrator administrates Doctor

The `Administrator` can have zero to n `Doctor`s he administrates in the system. In the initial state of the system, there is no `Doctor`. As the system should be able to handle a big amount of `Doctor`s, they are all related to the `Administrator` they were created by.

2.2.2 Doctor treats Patient

`Doctor`s treat from zero to n `Patient`s. When a `Doctor` is created, he doesn't treat any `Patient`. When doing treatments, he gets assigned all `Patient`s he treats. On the other side, `Patient`s can only be treated by a single `Doctor` (and there has to be a `Doctor` who treats the `Patient`).

2.2.3 Patient wears Device

When a `Device` is assigned to a `Patient`, this relationship represents that a specific `Device` is actually worn by a specific `Patient`. A `Device` can only be worn by a single `Patient` at the same time. On the other hand, a `Patient` can only wear a single `Device` at the same time. This relationship can also be represented by "`Patient` wears device during `Period`" and "`Device` is currently assigned to `Period`". Therefore it is intended that this relationship is not implemented in the actual code.

2.2.4 Patient wears device during Period

`Patient`s which are monitored need to have assigned a `Period` in which the monitoring takes place. `Patient`s can have multiple `Period`s if they are treated more than once, but a `Period` can only belong to a single `Patient` and the `Period` has to belong to a `Patient`.

2.2.5 Period has data from Device

In order to detect which `Device` records or has recorded the `Period`'s date, there is a relationship between `Period` and `Device`. `Period`s have exactly one `Device` since there is only one `Device` recording data for it. On the other hand, `Device`s can record data for any number of `Period`s, since they can be reconfigured to a new `Period` after they have completed monitoring for an arbitrary `Period`. The condition, that a `Device` can only be configured for a single `Period` at a time, is not directly represented in the model and has to be checked programmatically. This is done using the relationship "`Device` is currently assigned to `Period`".

2.2.6 Device is currently assigned to Period

`Period`s have a second relationship to `Device`. This relationship shows, that the `Device` is configured for the `Period`. A `Device` can only be assigned to one `Period` at a time. The `Period` which the `Device` is configured to must be the same as in the relationship "`Period` has data from `Device`".

2.2.7 Period has Measurement

`Period`s have a relationship to `Measurement` showing which `Measurement` has been taken for which `Period`. `Measurement`s belong to exactly one `Period` but a `Period` can have any number of `Measurement`s (including none).

2.2.8 Device takes Measurement

`Measurement`s are taken by `Device`s. They belong exactly to one `Device`. On the other hand, `Devices` can take any number of `Measurement`s. This relationship can be reflected using "`Period` has data from `Device`" and "`Period` has `Measurement`". Therefore it is intended that this relationship is not implemented in the actual code.