pkg

UserAccount
- username : String {5..32}
- password : String {5..32}

Authorithy
- ADMINISTRATOR : String = "ADMINISTRATOR"
- BROTHERHOOD : String = "BROTHERHOOD"
- MEMBER : String = "MEMBER"
- Authority : String {pattern("^^" + Authority.ADMINISTRATOR + "|" + Authority.BROTHERHOOD + "|"+"Authority.MEMBER"|"+"Authority.CHAPTER"$")}

Tickers must adhere to the following pattern: "yymmdd-xxxxxx", where "yymmdd" refers to the year, month, and day when the corresponding entity is registered, and "xxxxxx" to a random uppercase alpha-numeric string. No two entities may have the same ticker since it's assumed to be a unique external identifier.

Email addresses must adhere to any of the following patterns: "identifier@domain", "alias <identifier@domain>"; administrators may have email addresses of the form "identifier@", or "alias <identifier@>". The identifier is an alpha-numeric string, the domain is a sequence of alpha-numeric strings that are separated by dots, and the alias is a sequence of alpha-numeric strings that are separated by spaces.

ConfigurationParameters
+ name : String {NotBlank}
+ banner : String {NotBlank, URL}
+ sysMessage : String {NotBlank}
+ sysMessageEs : String {NotBlank}
+ countryCode : String {NotBlank, Pattern("^\\+\\d {1,3}$")}
+ finderCache : int {NotBlank,min(1),max(24)}
+ maxResults : int {NotBlank,min(1),max(100)}
+ finderCachedHours : int
+ finderMaxResults : int

Phone numbers should adhere to the following patterns: "+CC (AC) PN", "+CC PN", or "PN": "+CC" denotes a country code in range "+1" up to "+999", "(AC)" denotes an area code in range "(1)" up to "(999)", and "PN" denotes a number that must have at least four digits. Phone numbers with pattern "PN" must be added automatically a default country, which is a parameter that can be changed by administrators. The phone numbers should adhere to the previous patterns, but they are not required to. Whenever a phone number that does not match this pattern is entered, the system must ask for confirmation; if the user confirms the number, it then must be stored.

<<Abstract>>
Actor
+ name : String {NotBlank}
+ middleName : String{0..1}
+ surname : String {NotBlank}
+ photo : String{0..1} {URL}
+ email : String {Unique, NotBlank}
+ phone : String{0..1}
+ address : String{0..1}

EndYear couldn't be a future year.

Default configuration values:
name="Acme Madrugá"
banner="https://tinyurl.com/acme-madruga"
sysMessage="Welcome to Acme Madrugá, the site to organise your procesions."
sysMessageEs="¡Bienvenidos a Acme Madrugá! Tu sitio para organizar procesiones."
countryCode="+34"
finderCache = 1
maxResult=10

{overlapping}

Period Record
+ title : String {NotBlank}
+ description : String {NotBlank}
+ startYear : Integer {min(1500)}
+ endYear : Integer
+ picture : String{0..*} {URL}

Inception Record
+ title : String {NotBlank}
+ description : String {NotBlank}
+ picture : String[1..*] {URL}

A brotherhood cannot enrol any members or organise any processions until it's selected an area. Once the area is selected, it cannot be changed.

Administrator

Chapter
+ title : String {NotBlank}

The position in the brotherhood must be selected from a list that is maintained by the administrator of the system

There are no restrictions on the number of members who can have the same position in a brotherhood

Co-Ordinate

isBrotherOf

Area
+ name : String {Not Blank}
+ photo : String{0..*} {URL}

settleIn

Position
+ name : String {NotBlank}
+ nameEs : String {NotBlank}

Miscellaneous Record
+ title : String {NotBlank}
+ description : String {NotBlank}

History

Identify

Float
+ title : String {NotBlank}
+ description : String {NotBlank}
+ picture : String{0..*} {URL}

have

Brotherhood
+ title : String {NotBlank}
+ establishmentDate : Date
+ picture : String[1..*] {URL}

involves

have

Enrolment
+ moment : Date

makes

Member

have

Finder
+ keyWord : String
+ MinDate : Date
+ MaxDate : Date
+ lastSearch : Date

Legal Record
+ title : String {NotBlank}
+ description : String {NotBlank}
+ legalName : String {NotBlank}
+ vatNumber : Double {min(0)}
+ laws : String[1..*]

Link Record
+ title : String {NotBlank}
+ description : String {NotBlank}

linkedTo

involves

DropOut
+ moment : Date

makes

If a search criterion is not specified, then every procession meets it.

A float that compose a procession must belong to the brotherhood that organize the procession

isComposedOf

organize

Request

Parade
+ title : String {NotBlank}
+ description : String {NotBlank}
+ moment : Date
+ ticker : String {Unique, Pattern("^[0-9]{6}-[A-Z0-9] {5}$")}
+ mode : String {Pattern("^DRAFT|FINAL$")}
+ status : String {Pattern("^SUBMITTED|ACCEPTED|REJECTEDCLEARED$")}
+ explanation : String

have

Request
+ status : String {Pattern("^PENDING|APPROVED|REJECTED$")}
+ row : Integer[0..1] {min(0)}
+ column : Integer[0..1] {min(0)}
+ explanation : String[0..1]

The moment must be a future date

The segments must be contiguous, that is: the destination of a segment must be the origin of the following segment (if any).

Segment
+ originX : Integer {min(-90),max(90)}
+ originY : Integer {min(-180),max(180)}
+ destinationX : Integer {min(-90),max(90)}
+ destinationY : Integer {min(-180),max(180)}
+ originTime : Date
+ destinationTime : Date
+ sequence : Integer {min(1)}

isComposedOf

Path

isComposedOf

have