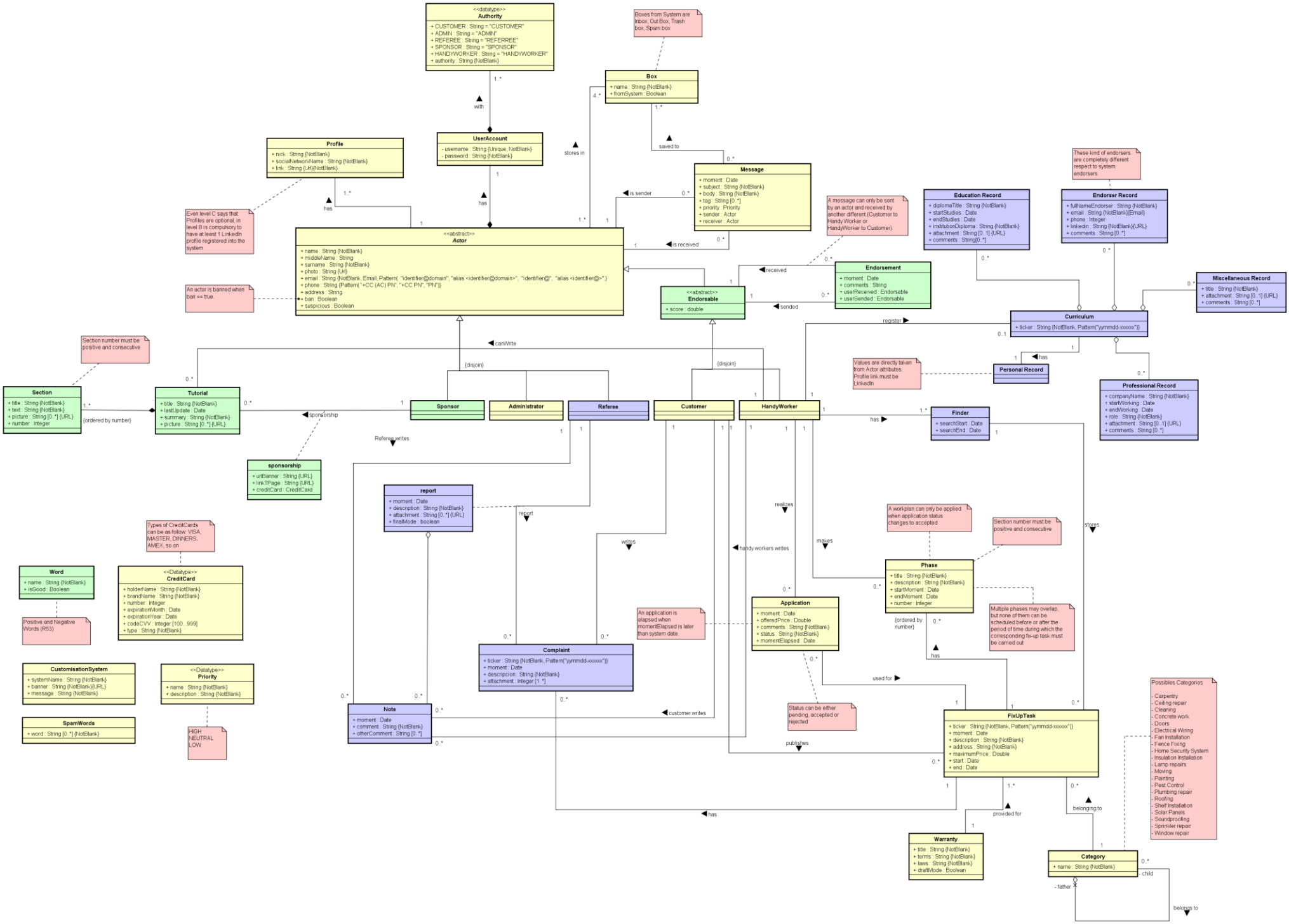
<<datatype>>
Authority
+ CUSTOMER : String = "CUSTOMER"
+ ADMIN : String = "ADMIN"
+ REFEREE : String = "REFEREE"
+ SPONSOR : String = "SPONSOR"
+ HANDYWORKER : String = "HANDYWORKER"
+ authority : String {NotBlank}
Boxes from System are Inbox, Out Box, Trash box, Spam box
Box
+ name : String {NotBlank}
+ fromSystem : Boolean
with
stores in
saved to
UserAccount
- username : String {Unique, NotBlank}
- password : String {NotBlank}
Profile
+ nick : String {NotBlank}
+ socialNetworkName : String {NotBlank}
+ link : String {Url}{NotBlank}
Message
+ moment : Date
+ subject : String {NotBlank}
+ body : String {NotBlank}
+ tag : String [0..*]
+ priority : Priority
+ sender : Actor
+ receiver : Actor
is sender
is received
has
Even level C says that Profiles are optional, in level B is compulsory to have at least 1 Linkedin profile registered into the system
<<abstract>>
Actor
+ name : String {NotBlank}
+ middleName : String
+ surname : String {NotBlank}
+ photo : String {Url}
+ email : String {NotBlank, Email, Pattern("identifier@domain", "alias <identifier@domain>", "identifier@", "alias <identifier@>" }
+ phone : String {Pattern("+CC (AC) PN", "+CC PN", "PN")}
+ address : String
+ ban : Boolean
+ suspicious : Boolean
An actor is banned when ban == true.
A message can only be sent by an actor and received by another different (Customer to Handy Worker or HandyWorker to Customer).
received
sended
<<abstract>>
Endorsable
+ score : double
Endorsement
+ moment : Date
+ comments : String
+ userReceived : Endorsable
+ userSended : Endorsable
Education Record
+ diplomaTitle : String {NotBlank}
+ startStudies : Date
+ endStudies : Date
+ institutionDiploma : String {NotBlank}
+ attachment : String [0..1] {URL}
+ comments : String[0..*]
These kind of endorsers are completely different respect to system endorsers
Endorser Record
+ fullNameEndorser : String {NotBlank}
+ email : String {NotBlank}{Email}
+ phone : Integer
+ linkedin : String {NotBlank}{URL}
+ comments : String [0..*]
Miscellaneous Record
+ title : String {NotBlank}
+ attachment : String [0..1] {URL}
+ comments : String [0..*]
Curriculum
+ ticker : String {NotBlank, Pattern("yymmdd-xxxxxx")}
register
has
Personal Record
Values are directly taken from Actor attributes. Profile link must be LinkedIn
Professional Record
+ companyName : String {NotBlank}
+ startWorking : Date
+ endWorking : Date
+ role : String {NotBlank}
+ attachment : String [0..1] {URL}
+ comments : String [0..*]
Section number must be positive and consecutive
Section
+ title : String {NotBlank}
+ text : String {NotBlank}
+ picture : String [0..*] {URL}
+ number : Integer
{ordered by number}
canWrite
{disjoin}
Tutorial
+ title : String {NotBlank}
+ lastUpdate : Date
+ summary : String {NotBlank}
+ picture : String [0..*] {URL}
sponsorship
sponsorship
+ urlBanner : String {URL}
+ linkTPage : String {URL}
+ creditCard : CreditCard
Sponsor
Administrator
Referee
Customer
HandyWorker
Finder
+ searchStart : Date
+ searchEnd : Date
Referee writes
report
+ moment : Date
+ description : String {NotBlank}
+ attachment : String [0..*] {URL}
+ finalMode : boolean
report
writes
handy workers writes
realizes
makes
stores
A workplan can only be applied when application status changes to accepted
Section number must be positive and consecutive
Phase
+ title : String {NotBlank}
+ description : String {NotBlank}
+ startMoment : Date
+ endMoment : Date
+ number : Integer
{ordered by number}
Multiple phases may overlap, but none of them can be scheduled before or after the period of time during which the corresponding fix-up task must be carried out
Application
+ moment : Date
+ offeredPrice : Double
+ comments : String {NotBlank}
+ status : String {NotBlank}
+ momentElapsed : Date
An application is elapsed when momentElapsed is later than system date.
Status can be either pending, accepted or rejected
used for
has
Types of CreditCards can be as follow: VISA, MASTER, DINNERS, AMEX, so on
Word
+ name : String {NotBlank}
+ isGood : Boolean
Positive and Negative Words (R53)
<<Datatype>>
CreditCard
+ holderName : String {NotBlank}
+ brandName : String {NotBlank}
+ number : Integer
+ expirationMonth : Date
+ expirationYear : Date
+ codeCVV : Integer [100...999]
+ type : String {NotBlank}
Complaint
+ ticker : String {NotBlank, Pattern("yymmdd-xxxxxx")}
+ moment : Date
+ descripcion : String {NotBlank}
+ attachment : Integer [1..*]
customer writes
Note
+ moment : Date
+ comment : String {NotBlank}
+ otherComment : String [0..*]
publishes
FixUpTask
+ ticker : String {NotBlank, Pattern("yymmdd-xxxxxx")}
+ moment : Date
+ description : String {NotBlank}
+ address : String {NotBlank}
+ maximumPrice : Double
+ start : Date
+ end : Date
CustomisationSystem
+ systemName : String {NotBlank}
+ banner : String {NotBlank}{URL}
+ message : String {NotBlank}
<<Datatype>>
Priority
+ name : String {NotBlank}
+ description : String {NotBlank}
HIGH
NEUTRAL
LOW
SpamWords
+ word : String [0..*] {NotBlank}
Possibles Categories
- Carpentry
- Ceiling repair
- Cleaning
- Concrete work
- Doors
- Electrical Wiring
- Fan Installation
- Fence Fixing
- Home Security System
- Insulation Installation
- Lamp repairs
- Moving
- Painting
- Pest Control
- Plumbing repair
- Roofing
- Shelf Installation
- Solar Panels
- Soundproofing
- Sprinkler repair
- Window repair
has
provided for
belonging to
Warranty
+ title : String {NotBlank}
+ terms : String {NotBlank}
+ laws : String {NotBlank}
+ draftMode : Boolean
Category
+ name : String {NotBlank}
- child
- father
belongs to