

```

1  {-# LANGUAGE OverloadedStrings #-}
2  {-|
3   This is the Haskell implementation of the Datatypes
4   described in the UML-class-diagram from storedData.pdf.
5   It is also converted into a .pdf and included into storedData.pdf.
6   |-}
7  module OpenBrain.Data where
8
9  import Data.Aeson ((.=), ToJSON(..), object)
10 import Data.Function (on)
11 import qualified Data.Aeson as Aeson
12
13 import OpenBrain.Data.Id
14 import OpenBrain.Data.Hash
15 import OpenBrain.Data.Json
16 import OpenBrain.Data.Salt
17
18 data Description = Description {
19     descriptionId :: DescriptionId
20 , author        :: Author
21 , headline      :: Headline
22 , description   :: String
23 , creationTime  :: Timestamp
24 , deletionTime  :: Timestamp
25 } deriving (Show)
26
27 data Article = Article {
28     articleId    :: ArticleId
29 , content       :: String
30 , children      :: [ArticleId]
31 , aDescription  :: Description
32 } deriving (Show)
33
34 data Relation = Relation {
35     relationId   :: RelationId
36 , source        :: ArticleId
37 , target        :: ArticleId
38 , rType         :: RelationType
39 , rDescription  :: Description
40 } deriving (Show)
41
42 data Collection = Collection {
43     collectionId :: CollectionId
44 , articles      :: [ArticleId]
45 , cDescription  :: Description
46 } deriving (Show)
47
48 data Discussion = Discussion {
49     discussionId :: DiscussionId
50 , participants  :: [UserId]
51 , deadline      :: Timestamp
52 , weights       :: [(UserId, Weight, RelationId)]
53 , result        :: Maybe Result
54 , dCollection   :: Collection
55 } deriving (Show)
56
57 data Result = Result {
58     resultId     :: ResultId
59 , choices       :: [(CollectionId, Votes)]
60 , voters        :: [(UserId, Voted)]
61 } deriving (Show)
62
63 data User = User {
64     userId       :: UserId
65 , username      :: String
66 , userhash      :: Hash
67 , usersalt      :: Salt
68 , userCreation  :: Timestamp
69 , lastLogin     :: Timestamp
70 , isAdmin       :: Bool

```

```

71 , profile      :: Maybe ArticleId
72 , session      :: Maybe SessionKey
73 } deriving (Show)
74
75 {-| Type aliases: |-}
76 type Author     = UserId
77 type Count      = Int
78 type Headline   = String
79 type Heir       = UserId
80 type IsAdmin    = Bool
81 type Limit      = Int
82 type Offset     = Int
83 type RelationType = String
84 type SessionKey = String
85 type Timestamp  = Integer
86 type Username   = String
87 type Voted      = Bool
88 type Votes      = Int
89 type Weight     = Int
90
91 {-| Instances of Eq: |-}
92 instance Eq Description where
93     (==) = (==) `on` descriptionId
94 instance Eq Article where
95     (==) = (==) `on` articleId
96 instance Eq Relation where
97     (==) = (==) `on` relationId
98 instance Eq Collection where
99     (==) = (==) `on` collectionId
100 instance Eq Discussion where
101     (==) = (==) `on` discussionId
102 instance Eq Result where
103     (==) = (==) `on` resultId
104 instance Eq User where
105     (==) = (==) `on` userId
106
107 {-| Instances of Ord: |-}
108 instance Ord Description where
109     compare = compare `on` descriptionId
110 instance Ord Article where
111     compare = compare `on` articleId
112 instance Ord Relation where
113     compare = compare `on` relationId
114 instance Ord Collection where
115     compare = compare `on` collectionId
116 instance Ord Discussion where
117     compare = compare `on` discussionId
118 instance Ord Result where
119     compare = compare `on` resultId
120 instance Ord User where
121     compare = compare `on` userId
122
123 {-| Instances of ToJSON: |-}
124 instance ToJSON Description where
125     toJSON d = object [
126         "descriptionId" .= descriptionId d
127         , "author"       .= author       d
128         , "headline"    .= headline     d
129         , "description"  .= description  d
130         , "creationTime" .= creationTime d
131         , "deletionTime" .= deletionTime d
132     ]
133 instance ToJSON Article where
134     toJSON a = merge (toJSON $ aDescription a) o
135     where
136         o = object [
137             "articleId" .= articleId a
138             , "content"  .= content   a
139             , "children" .= children  a
140         ]

```

```

141 instance ToJSON Relation where
142     toJSON r = merge (toJSON $ rDescription r) o
143     where
144         o = object [
145             "relationId" .= relationId r
146             , "source"     .= source     r
147             , "target"     .= target     r
148             , "rType"      .= rType      r
149         ]
150 instance ToJSON Collection where
151     toJSON c = merge (toJSON $ cDescription c) o
152     where
153         o = object ["collectionId" .= collectionId c, "articles" .= articles c]
154 instance ToJSON Discussion where
155     toJSON d = merge (toJSON $ dCollection d) o
156     where
157         o = object [
158             "discussionId" .= discussionId d
159             , "participants" .= participants d
160             , "deadline"     .= deadline     d
161             , "weights"      .= weights      d
162             , "result"       .= result       d
163         ]
164 instance ToJSON Result where
165     toJSON r = object [
166         "resultId" .= resultId r
167         , "choices" .= choices  r
168         , "voters"  .= voters   r
169     ]
170 instance ToJSON User where
171     toJSON u = object [
172         "userId"      .= userId      u
173         , "username"  .= username    u
174         , "userCreation" .= userCreation u
175         , "lastLogin"  .= lastLogin   u
176         , "isAdmin"    .= isAdmin     u
177         , "profile"    .= profile     u
178     ]

```