

Functional Languages 12th practice

1. Define a new type for privilege levels for a web service. Two levels are `Unprivileged` and `Admin`.
2. Define a new type `Cookie` for the web service, which represents data stored in clients' browser. A `Cookie` holds information on a client's state. Either the client has logged in with a username and has a privilege level, or the client is logged out.

For instance:

```
LoggedIn "dumbledore" Unprivileged :: Cookie
LoggedOut :: Cookie
```

3. The database of the web service consists of users in form of (username, password, privilege). The database is represented as list of users (list of rows in a relational database).

```
db = [("dumbledore", "abracadabra", Unprivileged), ("root", "secret", Admin), ("john", "apple", Unprivileged)]
```

4. Define a function for creating a new user account. Only an administrator has the right to create a new account. In other cases, don't change the database.

```
register "potter" "immobulus" (LoggedIn "root" Admin) db ==
  [("potter", "immobulus", Unprivileged), ("dumbledore", "abracadabra", Unprivileged), ("root", "secret", Admin), ("john", "apple", Unprivileged)]
register "potter" "immobulus" (LoggedIn "john" Unprivileged) db == db
register "potter" "immobulus" (LoggedIn "hacker" Unprivileged) db == db
```

5. Define a function for querying a user in the database. Use `Maybe`.

```
getUser "john" db == Just ("apple", Unprivileged)
getUser "root" db == Just ("secret", Admin)
getUser "chewbacca" db == Nothing
```

6. Define a function which authenticates a user using username and password. Return a logged-in `Cookie` when the username and password matches. Otherwise, return a logged-out `Cookie`.

```
login "john" "apple" db == LoggedIn "john" Unprivileged
login "john" "pear" db == LoggedOut
login "chewbacca" "alma" db == LoggedOut
```

7. Define a function for changing password. Only a logged-in user can change password and she can only change her own password, not the password of somebody else. A logged-out user cannot change the database.

```
passwd "pear" (LoggedIn "john" Unprivileged) db ==  
  [("dumbledore","abracadabra",Unprivileged), ("root","secret",Admin), ("john","pear",Unprivileged)]  
passwd "berry" LoggedOut db == db  
passwd "misanthrope" (LoggedIn "voldemort" Unprivileged) db == db
```

8. Define a function for removing a user. Only an administrator has the right to remove a user. In other cases, don't change the database.

```
delete "john" (LoggedIn "root" Admin) db == [("dumbledore","abracadabra",Unprivileged), ("root","secret",Admin)]  
delete "dumbledore" (LoggedIn "root" Admin) db == [("root","secret",Admin), ("john","apple",Unprivileged)]
```

9. List all users in the database.

```
users db == ["dumbledore", "root", "john"]
```