section *homestay* **parents** *standard_toolkit*

Contents

1 General

section *general* **parents** *homestay*

We want to have a few basic types here. We need to know general information about all of the users in the system:

- Email address, this is the unique identifier in the system
- Password, this has restrictions:
    - At least 8 characters in length
    - 1 digit
    - 1 uppercase character
    - 1 lowercase character
- First name, should be non-empty
- Last name, should be non-empty

$$[Email, Password, FirstName, LastName]$$

Using these basic types we construct more complex types in the system. The applicant is the user of the system. Password tokens are used to reset the password.

$$Applicant == Email \times Password \times FirstName \times LastName$$
$$PToken == \mathbb{N} \times Email$$

We need to be able to notify the user of what is happening in the application. So, we enumerate the possible responses from actions taken.

$$
\begin{array}{l}
Response ::= \\
\quad InvalidToken \\
\quad | \ PasswordResetSuccessful \\
\quad | \ LoginSuccessful
\end{array}
$$

We want an easy way to update an applicant. This helper function updates the applicant in the set of all applicants, identified by the email address.

function 42 **leftassoc** $(\_ \otimes \_)$

$$\_ \otimes \_ : \mathbb{P}\, Applicant \times Applicant \to \mathbb{P}\, Applicant$$

$$
\forall\, a_1 : Applicant;\ as : \mathbb{P}\, Applicant \bullet
$$
$$
\exists\, p_0, p_1 : Password;
$$
$$
fn_0, fn_1 : FirstName;
$$
$$
ln_0, ln_1 : LastName;
$$
$$
e : Email;
$$
$$
a_0 : Applicant\ |
$$
$$
a_1 = (e, p_1, fn_1, ln_1) \wedge (a_0 = (e, p_0, fn_0, ln_0) \in as) \bullet
$$
$$
as \otimes a_1 = (as \setminus \{a_0\}) \cup \{a_1\}
$$

Our initial state is just a bunch of empty sets.

__ *HomestayInitial* _____

$Applicants : \mathbb{P}\, Applicant$
$Emails : \mathbb{P}\, Email$
$Passwords : \mathbb{P}\, Password$
$Valid, Invalid : \mathbb{P}\, PToken$

$Applicants = \varnothing$
$Emails = \varnothing$
$Passwords = \varnothing$
$Valid = Invalid = \varnothing$

_____

In the database we need to ensure that there is each email address is unique, and that the password tokens are either valid or invalid.

__ *HomestayDatabase* _____

$HomestayInitial$

$\forall\, e_0, e_1 : Email;\ p_0, p_1 : Password;\ fn_0, fn_1 : FirstName;\ ln_0, ln_1 : LastName \bullet$
$\quad e_0 = e_1 \Rightarrow (p_0 = p_1 \wedge fn_0 = fn_1 \wedge ln_0 = ln_1)$
$Valid \cap Invalid = \varnothing$

_____

2 Login

When we go to create a new account, we need some information from the user. We update all of our sets to reflect the new addition. After they have successfully created an account, they are taken to the main menu.

CreateAccount
$\Delta HomestayDatabase$
$E? : Email$
$FN? : FirstName$
$LN? : LastName$
$P? : Password$

$E? \notin Emails$
$Emails' = Emails \cup \{E?\}$
$Passwords' = Passwords \cup \{P?\}$
$Applicants' = Applicants \cup \{(E?, P?, FN?, LN?)\}$
$Valid' = Valid$
$Invalid' = Invalid$

To login, a user needs to enter their email and password. At this point they are taken to the main menu.

Login
$\Xi HomestayDatabase$
$E? : Email$
$P? : Password$
$FN : FirstName$
$LN : LastName$
$Resp! : Response$

$E? \in Emails$
$P? \in Passwords$
$(E?, P?, FN, LN) \in Applicants$
$Resp! = LoginSuccessful$

Users can reset their password if they forget it. We take an email address, generate a password token, then dish that off to the email address. The user then finds the email with the reset token/link and proceeds to reset their password.

This is supposed to be one more step of indirection so that the user wont have their password reset at random. Of course, if the email address is already compromised, it doesn't make much difference.

```
┌─ ForgotPassword ──────────────────────────────────
│ ΔHomestayDatabase
│ E? : Email
│ T! : PToken
│ ID : ℕ
│ P : Password
│ FN : FirstName
│ LN : LastName
├───────────────────────────────────────────────────
│ E? ∈ Emails
│ (E?, P, FN, LN) ∈ Applicants
│ ID = # Valid + # Invalid + 1
│ T! = ID ↦ E?
│ Valid' = Valid ∪ {T!}
│ Invalid' = Invalid
└───────────────────────────────────────────────────
```

Once the user has the password token, they can enter their new password. We let them know that the reset was successful.

```
┌─ ResetPassword ───────────────────────────────────
│ ΔHomestayDatabase
│ P?, P : Password
│ T? : PToken
│ Resp! : Response
│ ID : ℕ
│ E : Email
│ FN : FirstName
│ LN : LastName
│ App₀, App₁ : Applicant
├───────────────────────────────────────────────────
│ T? ∈ Valid
│ E = second  T?
│ Valid' = Valid \ {T?}
│ Invalid' = Invalid ∪ {T?}
│ App₀ = (E, P, FN, LN) ∈ Applicants
│ App₁ = (E, P?, FN, LN)
│ Applicants' = Applicants ⊗ App₁
│ Resp! = PasswordResetSuccessful
└───────────────────────────────────────────────────
```

3 Student

section *student* **parents** *general*

4   Host

section *host* **parents** *general*

5 Administrator

section *admin* **parents** *general*

6 Group Chat

section $group_chat$ **parents** $general$

7 Profile

section *profile* **parents** *general*