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

Contents

1 General

section *general* **parents** *homestay*, *updates*, *projections*

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

$$
\begin{array}{l}
[\mathit{Email} \\
, \mathit{Password} \\
, \mathit{FirstName} \\
, \mathit{LastName} \\
, \mathit{StreetNumber} \\
, \mathit{StreetName} \\
, \mathit{City} \\
, \mathit{State} \\
, \mathit{ZipCode} \\
, \mathit{Phone} \\
, \mathit{Photo} \\
, \mathit{Text}]
\end{array}
$$

We create a couple of synonyms for photo types.

$$
\begin{array}{l}
\mathit{PostingPhoto} == \mathit{Photo} \\
\mathit{UserPhoto} == \mathit{Photo}
\end{array}
$$

We're going to construct a date type.

$$
\begin{aligned}
&Month ::= January \\
&\quad | \ February \\
&\quad | \ March \\
&\quad | \ April \\
&\quad | \ May \\
&\quad | \ June \\
&\quad | \ July \\
&\quad | \ August \\
&\quad | \ September \\
&\quad | \ October \\
&\quad | \ November \\
&\quad | \ December \\
&Day == \mathbb{N} \\
&Year == \mathbb{N} \\
&Date == Month \times Day \times Year
\end{aligned}
$$

We'll need a couple of flags.

We want to know whether or not the user is an admin or not. This is used so that they get access to certain features in the application.

$$
\begin{aligned}
&AdminFlag ::= \\
&\quad Admin \\
&\quad | \ NotAdmin \\
&InappropriateFlag ::= \\
&\quad Zero \\
&\quad | \ First \\
&\quad | \ Second \\
&\quad | \ Third \\
&\quad | \ Remove
\end{aligned}
$$

We need to know about the host stuff.

$$
\begin{aligned}
&Smoking ::= \\
&\quad EnjoysSmoking \\
&\quad \mid NonSmoking \\
&Pets ::= \\
&\quad NoPets \\
&\quad \mid YesPets \\
&Children ::= \\
&\quad One \\
&\quad \mid Two \\
&\quad \mid ThreePlus \\
&Diet ::= \\
&\quad GlutenFree \\
&\quad \mid Omnivore \\
&\quad \mid Pescatarian \\
&\quad \mid Vegetarian \\
&\quad \mid Vegan \\
&\quad \mid OtherDiet \\
&Religion ::= \\
&\quad Agnostic \\
&\quad \mid Athiest \\
&\quad \mid Buddhist \\
&\quad \mid Christian \\
&\quad \mid Catholic \\
&\quad \mid Mormon \\
&\quad \mid Muslim \\
&\quad \mid OtherReligion \\
&ExactOrArea ::= \\
&\quad ExactLocation \\
&\quad \mid Area
\end{aligned}
$$

We need to know about the student stuff.

$$MilesFromCampus == \mathbb{N}$$

We need to know how the two people feel about each other.

$$
\begin{aligned}
&MatchFlag ::= \\
&\quad Accept \\
&\quad \mid Reject
\end{aligned}
$$

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.

$$Applicant == Email \times Password \times FirstName \times LastName \times AdminFlag$$
$$PToken == \mathbb{N} \times Email$$
$$IFlag == Email \times InappropriateFlag$$
$$Message == (Applicant \times Applicant) \times Text$$

Now want to create some stuff specific to the host, but that needs to be used by other parts of the application.

$$Price == \mathbb{N}$$
$$SizeOfRoom == \mathbb{N} \times \mathbb{N}$$
$$Address == StreetNumber \times StreetName \times City \times State \times ZipCode$$
$$AddressInfo == Address \times ExactOrArea$$
$$Availability == Date \times Date$$
$$HostPreference == Applicant \times Smoking \times Pets \times Children \times Diet \times Religion$$
$$Posting == HostPreference \times AddressInfo \times SizeOfRoom \times Price \times Availability \times PostingPhoto$$

Now want to create some stuff specific to the host, but that needs to be used by other parts of the application.

$$StudentPreference == Applicant \times (Smoking \times Pets \times Children \times Diet \times Religion)$$
$$Search == StudentPreference$$

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.

$$Response ::= InvalidToken \mid PasswordResetSuccessful \mid LoginSuccessful \mid InvalidAvailability \mid ValidAvailability$$

1.1 Application State

Our initial state is just a bunch of empty sets.

HomestayInitial

$$
\begin{array}{l}
Applicants : \mathbb{P}\ Applicant \\
Chats : \mathbb{P}\ Message \\
ChatRequests : \mathbb{P}(Applicant \times Applicant) \\
Emails : \mathbb{P}\ Email \\
HostPreferences : \mathbb{P}\ HostPreference \\
IFlags : \mathbb{P}\ IFlag \\
Invalid : \mathbb{P}\ PToken \\
Matches : \mathbb{P}(Applicant \times Applicant \times MatchFlag) \\
Passwords : \mathbb{P}\ Password \\
PhoneNumbers : \mathbb{P}(Applicant \times Phone) \\
Postings : \mathbb{P}\ Posting \\
Searches : \mathbb{P}\ Search \\
UserPhotos : \mathbb{P}(Applicant \times Photo) \\
Valid : \mathbb{P}\ PToken \\
\hline
Applicants = \varnothing \\
Chats = \varnothing \\
ChatRequests = \varnothing \\
Emails = \varnothing \\
HostPreferences = \varnothing \\
IFlags = \varnothing \\
Invalid = \varnothing \\
Matches = \varnothing \\
Passwords = \varnothing \\
PhoneNumbers = \varnothing \\
Postings = \varnothing \\
Searches = \varnothing \\
UserPhotos = \varnothing \\
Valid = \varnothing
\end{array}
$$

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

$$
\begin{array}{l}
HomestayInitial \\
\hline
\forall\, e_0, e_1 : Email;\ p_0, p_1 : Password;\ fn_0, fn_1 : FirstName;\ ln_0, ln_1 : LastName \bullet \\
\quad e_0 = e_1 \Rightarrow (p_0 = p_1 \land fn_0 = fn_1 \land ln_0 = ln_1) \\
Valid \cap Invalid = \varnothing
\end{array}
$$

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** $(\_ \otimes \_)$

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

$$\begin{array}{l}
\forall\, a_1 : Applicant;\ as : \mathbb{P}\, Applicant \bullet \\
\quad \exists\, p_0, p_1 : Password; \\
\quad fn_0, fn_1 : FirstName; \\
\quad ln_0, ln_1 : LastName; \\
\quad ad_0, ad_1 : AdminFlag; \\
\quad e : Email; \\
\quad a_0 : Applicant \mid \\
\qquad a_1 = (e, p_1, fn_1, ln_1, ad_1) \land (a_0 = (e, p_0, fn_0, ln_0, ad_0) \in as) \bullet \\
\qquad\quad as \otimes a_1 = (as \setminus \{a_0\}) \cup \{a_1\}
\end{array}$$

1.2 General Operators

It gets kind of silly to have to rewrite these operators and functions each time, so we have some helpers here.

1.2.1 General Projections

section *projections* **parents** *homestay*

$$[A, B, C]$$

$$\begin{array}{l}
firstOf3 == \lambda\, a : A;\ b : B;\ c : C \bullet a \\
secondOf3 == \lambda\, a : A;\ b : B;\ c : C \bullet b \\
thirdOf3 == \lambda\, a : A;\ b : B;\ c : C \bullet c
\end{array}$$

$$[A, B, C, D]$$

$$\begin{array}{l}
firstOf4 == \lambda\, a : A;\ b : B;\ c : C;\ d : D \bullet a \\
secondOf4 == \lambda\, a : A;\ b : B;\ c : C;\ d : D \bullet b \\
thirdOf4 == \lambda\, a : A;\ b : B;\ c : C;\ d : D \bullet c \\
fourthOf4 == \lambda\, a : A;\ b : B;\ c : C;\ d : D \bullet d
\end{array}$$

$$[A, B, C, D, E]$$

$$\begin{array}{l}
firstOf5 == \lambda\, a : A;\ b : B;\ c : C;\ d : D;\ e : E \bullet a \\
secondOf5 == \lambda\, a : A;\ b : B;\ c : C;\ d : D;\ e : E \bullet b \\
thirdOf5 == \lambda\, a : A;\ b : B;\ c : C;\ d : D;\ e : E \bullet c \\
fourthOf5 == \lambda\, a : A;\ b : B;\ c : C;\ d : D;\ e : E \bullet d \\
fifthOf5 == \lambda\, a : A;\ b : B;\ c : C;\ d : D;\ e : E \bullet e
\end{array}$$

$$[A, B, C, D, E, F]$$

$$\begin{array}{l} firstOf6 == \lambda\, a : A;\ b : B;\ c : C;\ d : D;\ e : E;\ f : F \bullet a \\ secondOf6 == \lambda\, a : A;\ b : B;\ c : C;\ d : D;\ e : E;\ f : F \bullet b \\ thirdOf6 == \lambda\, a : A;\ b : B;\ c : C;\ d : D;\ e : E;\ f : F \bullet c \\ fourthOf6 == \lambda\, a : A;\ b : B;\ c : C;\ d : D;\ e : E;\ f : F \bullet d \\ fifthOf6 == \lambda\, a : A;\ b : B;\ c : C;\ d : D;\ e : E;\ f : F \bullet e \\ sixthOf6 == \lambda\, a : A;\ b : B;\ c : C;\ d : D;\ e : E;\ f : F \bullet f \end{array}$$

1.2.2 General Updates

section *updates* **parents** *homestay*

Updating for 3-tuples.

$$[A, B, C]$$

$$updateFirstOf3 : (A \times B \times C) \times A \rightarrow A \times B \times C$$

$$\forall\, abc : A \times B \times C;\ a : A \bullet \exists\, a_0 : A;\ b : B;\ c : C \mid abc = (a_0, b, c) \bullet updateFirstOf3(abc, a) = (a, b, c)$$

$$[A, B, C]$$

$$updateSecondOf3 : (A \times B \times C) \times B \rightarrow A \times B \times C$$

$$\forall\, abc : A \times B \times C;\ b : B \bullet \exists\, a : A;\ b_0 : B;\ c : C \mid abc = (a, b_0, c) \bullet updateSecondOf3(abc, b) = (a, b, c)$$

$$[A, B, C]$$

$$updateThirdOf3 : (A \times B \times C) \times C \rightarrow A \times B \times C$$

$$\forall\, abc : A \times B \times C;\ c : C \bullet \exists\, a : A;\ b : B;\ c_0 : C \mid abc = (a, b, c_0) \bullet updateThirdOf3(abc, c) = (a, b, c)$$

Updating for 4-tuples.

$$[A, B, C, D]$$

$$updateFirstOf4 : (A \times B \times C \times D) \times A \rightarrow A \times B \times C \times D$$

$$\forall\, abcd : A \times B \times C \times D;\ a : A \bullet \exists\, a_0 : A;\ b : B;\ c : C;\ d : D \mid abcd = (a_0, b, c, d) \bullet updateFirstOf4(abcd, a) = (a, b, c, d)$$

$$[A, B, C, D]$$

$$updateSecondOf4 : (A \times B \times C \times D) \times B \rightarrow A \times B \times C \times D$$

$$\forall\, abcd : A \times B \times C \times D;\ b : B \bullet \exists\, a : A;\ b_0 : B;\ c : C;\ d : D \mid abcd = (a, b_0, c, d) \bullet updateSecondOf4(abcd, b) = (a, b, c, d)$$

$$[A, B, C, D]$$

$$updateThirdOf4 : (A \times B \times C \times D) \times C \rightarrow A \times B \times C \times D$$

$$\forall\, abcd : A \times B \times C \times D;\ c : C \bullet \exists\, a : A;\ b : B;\ c_0 : C;\ d : D \mid abcd = (a, b, c_0, d) \bullet updateThirdOf4(abcd, c) = (a, b, c, d)$$

$$[A, B, C, D]$$

$$updateFourthOf4 : (A \times B \times C \times D) \times D \rightarrow A \times B \times C \times D$$

$$\forall\, abcd : A \times B \times C \times D;\ d : D \bullet \exists\, a : A;\ b : B;\ c : C;\ d_0 : D \mid abcd = (a, b, c, d_0) \bullet updateFourthOf4(abcd, d) = (a, b, c, d)$$

Updating for 5-tuples.

$$[A, B, C, D, E]$$

$$updateFirstOf5 : (A \times B \times C \times D \times E) \times A \rightarrow A \times B \times C \times D \times E$$

$$\forall\, abcde : A \times B \times C \times D \times E;\ a : A \bullet \exists\, a_0 : A;\ b : B;\ c : C;\ d : D;\ e : E \mid abcde = (a_0, b, c, d, e) \bullet updateFirstOf5(abcde, a) = (a, b, c, d, e)$$

$$[A, B, C, D, E]$$

$$updateSecondOf5 : (A \times B \times C \times D \times E) \times B \rightarrow A \times B \times C \times D \times E$$

$$\forall\, abcde : A \times B \times C \times D \times E;\ b : B \bullet \exists\, a : A;\ b_0 : B;\ c : C;\ d : D;\ e : E \mid abcde = (a, b_0, c, d, e) \bullet updateSecondOf5(abcde, b) = (a, b, c, d, e)$$

$$[A, B, C, D, E]$$

$$updateThirdOf5 : (A \times B \times C \times D \times E) \times C \rightarrow A \times B \times C \times D \times E$$

$$\forall\, abcde : A \times B \times C \times D \times E;\ c : C \bullet \exists\, a : A;\ b : B;\ c_0 : C;\ d : D;\ e : E \mid abcde = (a, b, c_0, d, e) \bullet updateThirdOf5(abcde, c) = (a, b, c, d, e)$$

$$[A, B, C, D, E]$$

$$updateFourthOf5 : (A \times B \times C \times D \times E) \times D \rightarrow A \times B \times C \times D \times E$$

$$\forall\, abcde : A \times B \times C \times D \times E;\ d : D \bullet \exists\, a : A;\ b : B;\ c : C;\ d_0 : D;\ e : E \mid abcde = (a, b, c, d_0, e) \bullet updateFourthOf5(abcde, d) = (a, b, c, d, e)$$

$$[A, B, C, D, E]$$

$$updateFifthOf5 : (A \times B \times C \times D \times E) \times E \rightarrow A \times B \times C \times D \times E$$

$$\forall\, abcde : A \times B \times C \times D \times E;\ e : E \bullet \exists\, a : A;\ b : B;\ c : C;\ d : D;\ e_0 : E \mid abcde = (a, b, c, d, e_0) \bullet updateFifthOf5(abcde, e) = (a, b, c, d, e)$$

Updating for 6-tuples.

$$[A, B, C, D, E, F]$$

$$updateFirstOf6 : (A \times B \times C \times D \times E \times F) \times A \rightarrow A \times B \times C \times D \times E \times F$$

$$\forall\, abcdef : A \times B \times C \times D \times E \times F;\ a : A \bullet \exists\, a_0 : A;\ b : B;\ c : C;\ d : D;\ e : E;\ f : F \mid abcdef = (a_0, b, c, d, e, f) \bullet updateFirstOf6(abcdef, a) = (a, b, c, d, e, f)$$

$$[A, B, C, D, E, F]$$

$$updateSecondOf6 : (A \times B \times C \times D \times E \times F) \times B \rightarrow A \times B \times C \times D \times E \times F$$

$$\forall\, abcdef : A \times B \times C \times D \times E \times F;\ b : B \bullet \exists\, a : A;\ b_0 : B;\ c : C;\ d : D;\ e : E;\ f : F \mid abcdef = (a, b_0, c, d, e, f) \bullet updateSecondOf6(abcdef, b) = (a, b, c, d, e, f)$$

$$[A, B, C, D, E, F]$$

$$updateThirdOf6 : (A \times B \times C \times D \times E \times F) \times C \rightarrow A \times B \times C \times D \times E \times F$$

$$\forall\, abcdef : A \times B \times C \times D \times E \times F;\ c : C \bullet \exists\, a : A;\ b : B;\ c_0 : C;\ d : D;\ e : E;\ f : F \mid abcdef = (a, b, c_0, d, e, f) \bullet updateThirdOf6(abcdef, c) = (a, b, c, d, e, f)$$

$$[A, B, C, D, E, F]$$

$$updateFourthOf6 : (A \times B \times C \times D \times E \times F) \times D \rightarrow A \times B \times C \times D \times E \times F$$

$$\forall\, abcdef : A \times B \times C \times D \times E \times F;\ d : D \bullet \exists\, a : A;\ b : B;\ c : C;\ d_0 : D;\ e : E;\ f : F \mid abcdef = (a, b, c, d_0, e, f) \bullet updateFourthOf6(abcdef, d) = (a, b, c, d, e, f)$$

$$[A, B, C, D, E, F]$$

$$updateFifthOf6 : (A \times B \times C \times D \times E \times F) \times E \rightarrow A \times B \times C \times D \times E \times F$$

$$\forall\, abcdef : A \times B \times C \times D \times E \times F;\ e : E \bullet \exists\, a : A;\ b : B;\ c : C;\ d : D;\ e_0 : E;\ f : F \mid abcdef = (a, b, c, d, e_0, f) \bullet updateFifthOf6(abcdef, e) = (a, b, c, d, e, f)$$

$$[A, B, C, D, E, F]$$

$$updateSixthOf6 : (A \times B \times C \times D \times E \times F) \times F \rightarrow A \times B \times C \times D \times E \times F$$

$$\forall\, abcdef : A \times B \times C \times D \times E \times F;\ f : F \bullet \exists\, a : A;\ b : B;\ c : C;\ d : D;\ e : E;\ f_0 : F \mid abcdef = (a, b, c, d, e, f_0) \bullet updateSixthOf6(abcdef, f) = (a, b, c, d, e, f)$$

1.3 Morphisms

1.3.1 Month

$$
\begin{array}{|l}
month2Nat : Month \rightarrow \mathbb{N} \\
\hline
month2Nat\ January = 1 \land \\
month2Nat\ February = 2 \land \\
month2Nat\ March = 3 \land \\
month2Nat\ April = 4 \land \\
month2Nat\ May = 5 \land \\
month2Nat\ June = 6 \land \\
month2Nat\ July = 7 \land \\
month2Nat\ August = 8 \land \\
month2Nat\ September = 9 \land \\
month2Nat\ October = 10 \land \\
month2Nat\ November = 11 \land \\
month2Nat\ December = 12
\end{array}
$$

$$
\begin{array}{|l}
nat2Month : \mathbb{N} \pfun Month \\
\hline
nat2Month\ 1 = January \land \\
nat2Month\ 2 = February \land \\
nat2Month\ 3 = March \land \\
nat2Month\ 4 = April \land \\
nat2Month\ 5 = May \land \\
nat2Month\ 6 = June \land \\
nat2Month\ 7 = July \land \\
nat2Month\ 8 = August \land \\
nat2Month\ 9 = September \land \\
nat2Month\ 10 = October \land \\
nat2Month\ 11 = November \land \\
nat2Month\ 12 = December
\end{array}
$$

1.3.2 InappropriateFlag

$$
\begin{array}{|l}
InFlag2Nat : InappropriateFlag \rightarrow \mathbb{N} \\
\hline
InFlag2Nat\ Zero = 0 \land \\
InFlag2Nat\ First = 1 \land \\
InFlag2Nat\ Second = 2 \land \\
InFlag2Nat\ Third = 3 \land \\
InFlag2Nat\ Remove = 4
\end{array}
$$

$$
\begin{array}{|l}
Nat2InFlag : \mathbb{N} \pfun InappropriateFlag \\
\hline
Nat2InFlag\ 0 = Zero \land \\
Nat2InFlag\ 1 = First \land \\
Nat2InFlag\ 2 = Second \land \\
Nat2InFlag\ 3 = Third \land \\
Nat2InFlag\ 4 = Remove
\end{array}
$$

2 Login

section *login* **parents** *general*

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.

CreateUserAccount

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

$$E? \notin Emails$$
$$Emails' = Emails \cup \{E?\}$$
$$Passwords' = Passwords \cup \{P?\}$$
$$Applicants' = Applicants \cup \{(E?, P?, FN?, LN?, NotAdmin)\}$$
$$PhoneNumbers' = PhoneNumbers$$
$$UserPhotos' = UserPhotos$$
$$Valid' = Valid$$
$$Invalid' = Invalid$$
$$Postings' = Postings$$
$$HostPreferences' = HostPreferences$$
$$Searches' = Searches$$

CreateAdminAccount

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

$E? \notin Emails$
$AD? = Admin$
$Emails' = Emails \cup \{E?\}$
$Passwords' = Passwords \cup \{P?\}$
$Applicants' = Applicants \cup \{(E?, P?, FN?, LN?, AD?)\}$
$PhoneNumbers' = PhoneNumbers$
$UserPhotos' = UserPhotos$
$Valid' = Valid$
$Invalid' = Invalid$
$Postings' = Posting$
$HostPreferences' = HostPreferences$
$Searches' = Searches$

A person can create either a user account, or they can create an admin account. The idea is that the admin account isn't something that you can specify, but you must be given a link to sign up for. The link you follow gives the admin flag.

$$CreateAccount == CreateUserAccount \lor CreateAdminAccount$$

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

Login

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

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

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

$\Delta HomestayDatabase$
$E? : Email$
$T! : PToken$
$ID : \mathbb{N}$
$P : Password$
$FN : FirstName$
$LN : LastName$
$AD : AdminFlag$

$E? \in Emails$
$(E?, P, FN, LN, AD) \in Applicants$
$ID = \# Valid + \# Invalid + 1$
$T! = ID \mapsto E?$
$Valid' = Valid \cup \{T!\}$
$Invalid' = Invalid$
$Applicants' = Applicants$
$Emails' = Emails$
$Passwords' = Passwords$
$PhoneNumbers' = PhoneNumbers$
$UserPhotos' = UserPhotos$
$Postings' = Postings$
$HostPreferences' = HostPreferences$
$Searches' = Searches$

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

ResetPassword

$\Delta HomestayDatabase$
$P?, P : Password$
$T? : PToken$
$Resp! : Response$
$ID : \mathbb{N}$
$E : Email$
$FN : FirstName$
$LN : LastName$
$AD : AdminFlag$
$App_0, App_1 : Applicant$

$T? \in Valid$
$E = second\ \ T?$
$Valid' = Valid \setminus \{T?\}$
$Invalid' = Invalid \cup \{T?\}$
$App_0 = (E, P, FN, LN, AD) \in Applicants$
$App_1 = (E, P?, FN, LN, AD)$
$Applicants' = Applicants \otimes App_1$
$Resp! = PasswordResetSuccessful$
$Emails' = Emails$
$Passwords' = Passwords$
$PhoneNumbers' = PhoneNumbers$
$Postings' = Postings$
$HostPreferences' = HostPreferences$
$Searches' = Searches$

3 Student

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

The student will search by entering his/her/its preference information. Preferences include: Children, Diet, Pets, Religion, and Smoking.

If the student likes what he/she/it sees, he/she/it can save the search.

SaveSearch

$$\Delta HomestayDatabase$$
$$S? : Search$$

$$Searches' = Searches \cup \{S?\}$$
$$Applicants' = Applicants$$
$$Chats' = Chats$$
$$Emails' = Emails$$
$$HostPreferences' = HostPreferences$$
$$IFlags' = IFlags$$
$$Invalid' = Invalid$$
$$Matches' = Matches$$
$$Passwords' = Passwords$$
$$PhoneNumbers' = PhoneNumbers$$
$$Postings' = Postings$$
$$UserPhotos' = UserPhotos$$
$$Valid' = Valid$$

RecallSearches

$$\Xi HomestayDatabase$$
$$App? : Applicant$$
$$S! : \mathbb{P}\, Search$$

$$S! = \{App?\} \lhd Searches$$

SubmitChatRequest

$\Delta HomestayDatabase$

$From?, To? : Applicant$

$ChatRequests' = ChatRequests \cup \{(From?, To?)\}$

$Applicants = \varnothing$

$Chats = \varnothing$

$Emails = \varnothing$

$HostPreferences = \varnothing$

$IFlags = \varnothing$

$Invalid = \varnothing$

$Matches = \varnothing$

$Passwords = \varnothing$

$PhoneNumbers = \varnothing$

$Postings = \varnothing$

$Searches = \varnothing$

$UserPhotos = \varnothing$

$Valid = \varnothing$

When performing the search, we need to match all of the student's preferences with any hosts. In the actual application we will want to have priority levels as well as this simple matching.

The priority levels will weight the hosts for the automatic search.

$dropApplicant : HostPreference \rightarrow (Smoking \times Pets \times Children \times Diet \times Religion)$

$$\forall hp : HostPreference \bullet$$
$$\quad \exists a : Applicant;\ s : Smoking;\ p : Pets;\ c : Children;\ d : Diet;\ r : Religion \mid$$
$$\quad hp = (a, s, p, c, d, r) \bullet$$
$$\quad\quad dropApplicant\ hp = (s, p, c, d, r)$$

PerformSearch

$\Xi HomestayDatabase$

$S? : StudentPreference$

$HostPrefs! : \mathbb{P}\ HostPreference$

$A, A_0 : Applicant$

$StudentSmoking : Smoking$

$StudentPets : Pets$

$StudentChildren : Children$

$StudentDiet : Diet$

$StudentReligion : Religion$

$A_0 \mapsto (StudentSmoking, StudentPets, StudentChildren, StudentDiet, StudentReligion) = S?$

$HostPrefs! = \{(A, StudentSmoking, StudentPets, StudentChildren, StudentDiet, StudentReligion)\}$

4 Host

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

The host section allows the user to view and modify preferences specific to hosting a location. There are some general preferences for each host, e.g. smoking, and pets. Then there are preferences for each host location, e.g. price and availability.

4.1 Preferences

$$
\begin{array}{|l}
updateHostPrefs : \mathbb{P}\, HostPreference \times HostPreference \rightarrow \mathbb{P}\, HostPreference \\
\hline
\forall\, hp : HostPreference;\ hps : \mathbb{P}\, HostPreference \bullet \\
\quad \exists\, hp_0 : HostPreference \mid \\
\qquad firstOf 6\ hp_0 = firstOf 6\ hp \bullet \\
\qquad\quad updateHostPrefs(hps, hp) = (hps \setminus \{hp_0\}) \cup \{hp\} \\
\end{array}
$$

EditHostSmoking

$$
\begin{array}{|l}
\Delta HomestayDatabase \\
H? : HostPreference \\
H : HostPreference \\
S? : Smoking \\
\hline
H? \in HostPreferences \\
H = updateSecondOf 6(H?, S?) \\
HostPreferences' = updateHostPrefs(HostPreferences, H) \\
Applicants' = Applicants \\
Emails' = Emails \\
Passwords' = Passwords \\
PhoneNumbers' = PhoneNumbers \\
UserPhotos' = UserPhotos \\
Valid' = Valid \\
Invalid' = Invalid \\
Postings' = Postings \\
Searches' = Searches \\
\\
\end{array}
$$

EditHostPets

$\Delta HomestayDatabase$
$H?, H : HostPreference$
$P? : Pets$

$H? \in HostPreferences$
$H = updateThirdOf6(H?, P?)$
$HostPreferences' = updateHostPrefs(HostPreferences, H)$
$Applicants' = Applicants$
$Emails' = Emails$
$Passwords' = Passwords$
$PhoneNumbers' = PhoneNumbers$
$UserPhotos' = UserPhotos$
$Valid' = Valid$
$Invalid' = Invalid$
$Postings' = Postings$
$Searches' = Searches$

EditHostChildren

$\Delta HomestayDatabase$
$H?, H : HostPreference$
$C? : Children$

$H? \in HostPreferences$
$H = updateFourthOf6(H?, C?)$
$HostPreferences' = updateHostPrefs(HostPreferences, H)$
$Applicants' = Applicants$
$Emails' = Emails$
$Passwords' = Passwords$
$PhoneNumbers' = PhoneNumbers$
$UserPhotos' = UserPhotos$
$Valid' = Valid$
$Invalid' = Invalid$
$Postings' = Postings$
$Searches' = Searches$

EditHostDiet

$\Delta HomestayDatabase$
$H?, H : HostPreference$
$D? : Diet$

$H? \in HostPreferences$
$H = updateFifthOf6(H?, D?)$
$HostPreferences' = updateHostPrefs(HostPreferences, H)$
$Applicants' = Applicants$
$Emails' = Emails$
$Passwords' = Passwords$
$PhoneNumbers' = PhoneNumbers$
$UserPhotos' = UserPhotos$
$Valid' = Valid$
$Invalid' = Invalid$
$Postings' = Postings$
$Searches' = Searches$

EditHostReligion

$\Delta HomestayDatabase$
$H?, H : HostPreference$
$R? : Religion$

$H? \in HostPreferences$
$H = updateSixthOf6(H?, R?)$
$HostPreferences' = updateHostPrefs(HostPreferences, H)$
$Applicants' = Applicants$
$Emails' = Emails$
$Passwords' = Passwords$
$PhoneNumbers' = PhoneNumbers$
$UserPhotos' = UserPhotos$
$Valid' = Valid$
$Invalid' = Invalid$
$Postings' = Postings$
$Searches' = Searches$

$EditHostPreferences == EditHostSmoking \lor$
$EditHostPets \lor$
$EditHostChildren \lor$
$EditHostDiet \lor$
$EditHostReligion$

4.2 Validation

So long as the start date is before the end date, we consider the Availability to be valid.

ValiDateGood

$$
\begin{array}{l}
Avail? : Availability \\
AvailResp! : Response \\
Start, End : Date \\
Y_0, Y_1 : Year \\
M_0, M_1 : Month \\
D_0, D_1 : Day \\
\hline
((M_0, D_0, Y_0), (M_1, D_1, Y_1)) = Avail? \\
(Y_0 < Y_1) \lor \\
\quad (Y_0 = Y_1 \land month2Nat\ M_0 < month2Nat\ M_1) \lor \\
\quad (Y_0 = Y_1 \land M_0 = M_1 \land D_0 < D_1) \\
AvailResp! = ValidAvailability
\end{array}
$$

If the start date is after the end date, then no bueno.

ValiDateBad

$$
\begin{array}{l}
Avail? : Availability \\
AvailResp! : Response \\
Start, End : Date \\
Y_0, Y_1 : Year \\
M_0, M_1 : Month \\
D_0, D_1 : Day \\
\hline
((M_0, D_0, Y_0), (M_1, D_1, Y_1)) = Avail? \\
(Y_0 > Y_1) \lor \\
\quad (Y_0 = Y_1 \land month2Nat\ M_0 > month2Nat\ M_1) \lor \\
\quad (Y_0 = Y_1 \land M_0 = M_1 \land D_0 > D_1) \\
AvailResp! = InvalidAvailability
\end{array}
$$

$$ValiDate == ValiDateGood \lor ValiDateBad$$

4.3 Postings

To create a new posting, the host enters all the relevant information for hosting

NewPosting

$\Delta HomestayDatabase$
$ValiDate$
$H? : HostPreference$
$Addr? : Address$
$E? : ExactOrArea$
$WIDTH? : \mathbb{N}$
$LENGTH? : \mathbb{N}$
$PR? : Price$
$Avail? : Availability$
$PH? : Photo$
$P : Posting$
$AI : AddressInfo$
$S : SizeOfRoom$
$AvailResp! : Response$

$AvailResp! = ValidAvailability$
$AI = Addr? \mapsto E?$
$S = WIDTH? \mapsto LENGTH?$
$P = (H?, AI, S, PR?, Avail?, PH?)$
$Postings' = Postings \cup \{P\}$
$Applicants' = Applicants$
$Emails' = Emails$
$Passwords' = Passwords$
$PhoneNumbers' = PhoneNumbers$
$UserPhotos' = UserPhotos$
$Valid' = Valid$
$Invalid' = Invalid$
$HostPreferences' = HostPreferences$
$Searches' = Searches$

Deleting a post removes the post from Postings

DeletePosting

$\Delta HomestayDatabase$
$P? : Posting$

$P? \in Postings$
$Postings' = Postings \setminus \{P?\}$
$Applicants' = Applicants$
$Emails' = Emails$
$Passwords' = Passwords$
$PhoneNumbers' = PhoneNumbers$
$UserPhotos' = UserPhotos$
$Valid' = Valid$
$Invalid' = Invalid$
$HostPreferences' = HostPreferences$
$Searches' = Searches$

5 Administrator

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

Administrator login only needs to act like an inbox with "flag as inappropriate" or "accept/reject match" messages from students. The administrator will have a link to the group chat that is flagged and can add messages or end the chat if it is inappropriate. If the student and host decide to chat, once they send in their accept/reject match to the admin, the admin will have final approval to accept or reject the match.

$$
\begin{aligned}
&AcceptMessage ::= \\
&\quad OneAccept \\
&\quad \mid TwoAccepts \\
&\quad \mid NoAccepts \\
&\quad \mid Inappropriate
\end{aligned}
$$

$$
\begin{array}{|l}
AdminMatch : AcceptMessage \pfun MatchFlag \\
\hline
AdminMatch\ OneAccept = Reject \land \\
AdminMatch\ NoAccepts = Reject \land \\
AdminMatch\ TwoAccepts = Accept
\end{array}
$$

When we need to get rid of a user, we have some steps to take

- Remove the applicant.
- Remove the email corresponding to the applicant.
- Remove the password corresponding to the applicant.

RemoveUser

$\Delta HomestayDatabase$
$A? : Applicant$
$E : Email$
$P : Password$

$E = firstOf5\ A?$
$P = secondOf5\ A?$
$Applicants' = Applicants \setminus \{A?\}$
$Emails' = Emails \setminus \{E\}$
$Passwords' = Passwords \setminus \{P\}$
$HostPreferences' = HostPreferences$
$PhoneNumbers' = PhoneNumbers$
$UserPhotos' = UserPhotos$
$Valid' = Valid$
$Invalid' = Invalid$
$Postings' = Postings$

AdminInappropriate

$\Delta HomestayDatabase$
$M? : AcceptMessage$
$E? : Email$
$I : IFlag$
$In, In2 : InappropriateFlag$

$M? = Inappropriate$
$\{E? \mapsto In\} = \{E?\} \lhd IFlags$
$In \neq Remove$
$In2 = Nat2InFlag((InFlag2Nat\ In) + 1)$
$IFlags' = IFlags \oplus \{E? \mapsto In2\}$
$In2 \neq Remove$
$Applicants' = Applicants$
$Emails' = Emails$
$Passwords' = Passwords$
$PhoneNumbers' = PhoneNumbers$
$UserPhotos' = UserPhotos$
$Valid' = Valid$
$Invalid' = Invalid$

InappropriateUser

$\Delta HomestayDatabase$
$RemoveUser$
$M? : AcceptMessage$
$E? : Email$
$I : IFlag$
$In, In2 : InappropriateFlag$

$M? = Inappropriate$
$\{E? \mapsto In\} = \{E?\} \lhd IFlags$
$In \neq Remove$
$In2 = Nat2InFlag((InFlag2Nat\ In) + 1)$
$IFlags' = IFlags \oplus \{E? \mapsto In2\}$
$In2 = Remove$

6 Group Chat

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

We want the user to be able to see all the chats they are a part of. When they see the chats, they should have a few options.

- Add a new message to a chat.
- Approve the union of Student and Host synergy.
- Disapprove the union of Student and Host synergy.
- Delete the chat.

AddMessage

$$\Delta HomestayDatabase$$
$$From?, To? : Applicant$$
$$Msg? : Text$$

$$Chats' = Chats \cup \{(From?, To?) \mapsto Msg?\}$$
$$Applicants' = Applicants$$
$$Emails' = Emails$$
$$HostPreferences' = HostPreferences$$
$$IFlags' = IFlags$$
$$Invalid' = Invalid$$
$$Matches' = Matches$$
$$Passwords' = Passwords$$
$$PhoneNumbers' = PhoneNumbers$$
$$Postings' = Postings$$
$$UserPhotos' = UserPhotos$$
$$Valid' = Valid$$

ApproveSynergy

$\Delta HomestayDatabase$
$Match? : MatchFlag$
$App_0?, App_1? : Applicant$

$Match? = Accept$
$Matches' = Matches \cup \{(App_0?, App_1?, Match?)\}$
$Applicants' = Applicants$
$Chats' = Chats$
$Emails' = Emails$
$HostPreferences' = HostPreferences$
$IFlags' = IFlags$
$Invalid' = Invalid$
$Passwords' = Passwords$
$PhoneNumbers' = PhoneNumbers$
$Postings' = Postings$
$UserPhotos' = UserPhotos$
$Valid' = Valid$

DisapproveSynergy

$\Delta HomestayDatabase$
$Match? : MatchFlag$
$App_0?, App_1? : Applicant$

$Match? = Reject$
$Matches' = Matches \cup \{(App_0?, App_1?, Match?)\}$
$Applicants' = Applicants$
$Chats' = Chats$
$Emails' = Emails$
$HostPreferences' = HostPreferences$
$IFlags' = IFlags$
$Invalid' = Invalid$
$Passwords' = Passwords$
$PhoneNumbers' = PhoneNumbers$
$Postings' = Postings$
$UserPhotos' = UserPhotos$
$Valid' = Valid$

DeleteChat

$\Delta HomestayDatabase$

$From?, To? : Applicant$

$Chats' = \{(From?, To?)\} \lhd Chats$

$Applicants' = Applicants$

$Emails' = Emails$

$HostPreferences' = HostPreferences$

$IFlags' = IFlags$

$Invalid' = Invalid$

$Matches' = Matches$

$Passwords' = Passwords$

$PhoneNumbers' = PhoneNumbers$

$Postings' = Postings$

$UserPhotos' = UserPhotos$

$Valid' = Valid$

7 Profile

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

The profile section has functionality to update/change basic user information: First Name, Last Name, User Photo, Email, and Phone Number.

We just need to replace the first name for the applicant.

EditFirstName

$$
\begin{array}{l}
\Delta HomestayDatabase \\
FN? : FirstName \\
App_0?, App_1 : Applicant \\
\hline
App_0? \in Applicants \\
App_1 = updateThirdOf5(App_0?, FN?) \\
Applicants' = Applicants \otimes App_1 \\
Emails' = Emails \\
Passwords' = Passwords \\
PhoneNumbers' = PhoneNumbers \\
Valid' = Valid \\
Invalid' = Invalid \\
HostPreferences' = HostPreferences \\
Postings' = Postings \\
Searches' = Searches \\
UserPhotos' = UserPhotos
\end{array}
$$

We just need to replace the last name for the applicant.

EditLastName

$\Delta HomestayDatabase$
$LN? : LastName$
$App_0?, App_1 : Applicant$

$App_0? \in Applicants$
$App_1 = updateFourthOf5(App_0?, LN?)$
$Applicants' = Applicants \otimes App_1$
$Emails' = Emails$
$Passwords' = Passwords$
$PhoneNumbers' = PhoneNumbers$
$UserPhotos' = UserPhotos$
$Valid' = Valid$
$Invalid' = Invalid$
$HostPreferences' = HostPreferences$
$Postings' = Postings$
$Searches' = Searches$

We just need to replace the photo for the applicant.

EditPhoto

$\Delta HomestayDatabase$
$UP? : UserPhoto$
$App? : Applicant$

$UserPhotos' = UserPhotos \oplus \{App? \mapsto UP?\}$
$Applicants' = Applicants$
$Emails' = Emails$
$Passwords' = Passwords$
$PhoneNumbers' = PhoneNumbers$
$Valid' = Valid$
$Invalid' = Invalid$
$HostPreferences' = HostPreferences$
$Postings' = Postings$
$Searches' = Searches$

We have to check that the email is not currently used in the system. Then update the email for the applicant and ensure the changes cascade throughout the system.

EditEmail

$\Delta HomestayDatabase$
$FN : FirstName$
$LN : LastName$
$P : Password$
$E? , E : Email$
$AD : AdminFlag$

$\exists App_0, App_1 : Applicant \mid App_0 = (E, P, FN, LN, AD) \in Applicants \land$
$App_1 = (E?, P, FN, LN, AD) \notin Applicants \bullet$
$Applicants' = Applicants \setminus \{App_0\} \land Applicants' = Applicants \cup \{App_1\}$
$Emails' = Emails$
$Passwords' = Passwords$
$PhoneNumbers' = PhoneNumbers$
$UserPhotos' = UserPhotos$
$Valid' = Valid$
$Invalid' = Invalid$
$HostPreferences' = HostPreferences$
$Postings' = Postings$
$Searches' = Searches$

We just need to update the phone number for the applicant.

EditPhone

$\Delta HomestayDatabase$
$PNUM? : Phone$
$App? : Applicant$

$PhoneNumbers' = PhoneNumbers \oplus \{App? \mapsto PNUM?\}$
$Applicants' = Applicants$
$Emails' = Emails$
$Passwords' = Passwords$
$UserPhotos' = UserPhotos$
$Valid' = Valid$
$Invalid' = Invalid$
$HostPreferences' = HostPreferences$
$Postings' = Postings$
$Searches' = Searches$