

6.170 RooMIT Design

Rujia Zha, Peinan Chen, Olga Shestopalova, Alexander Heifetz

[Motivation](#)

[Purposes](#)

[Context Diagram](#)

[Concepts](#)

[Data Model](#)

[Security Concerns](#)

[Threat Model](#)

[User Interface Wireframes](#)

[Design Challenges](#)

Motivation

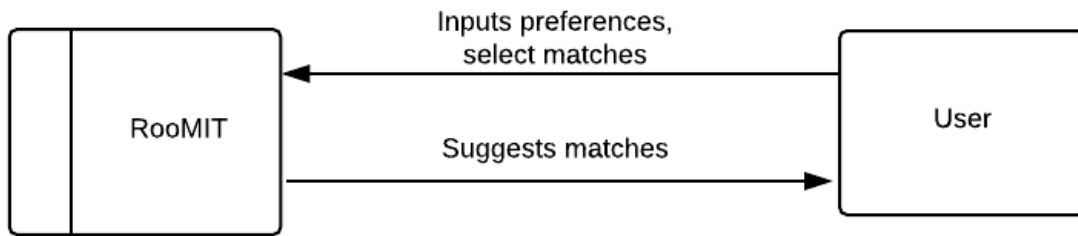
Most MIT students live in dorms (70%) and most of those students have roommates. MIT housing provides a way to lottery into each dorm, but no way to pair with a compatible roommate. RooMIT provides a solution to this problem by matching students according to their preferences- things that students usually find out about each other when they're well into their stay together.

Students will submit their responses to an assortment of preference questions, and then get matched with other students that have similar preferences and from there can request other students to be roommates, and can get accepted or rejected. We support multiple roommate agreements (eg. triples, quads, etc). When users decide they are done with their group of roommates, they can declare themselves matched and remove themselves from the search system. Currently the app is limited to MIT students and so we give out the emails of approved matches so students can contact each other. We aim to eventually be able to communicate with other students via our app instead of email.

Purposes

- **Help ensure roommate compatibility.** Currently the housing system does not have a formalized method to help students find suitable people to live with, so this application will help address this issue.
- **Help find roommates in an efficient and timely manner.** The idealized method for students to find the most suitable roommates would be to talk to everyone who will be living in their residence. This app will alleviate the amount of time it takes to do this by automating the process using key pieces of information.

Context Diagram

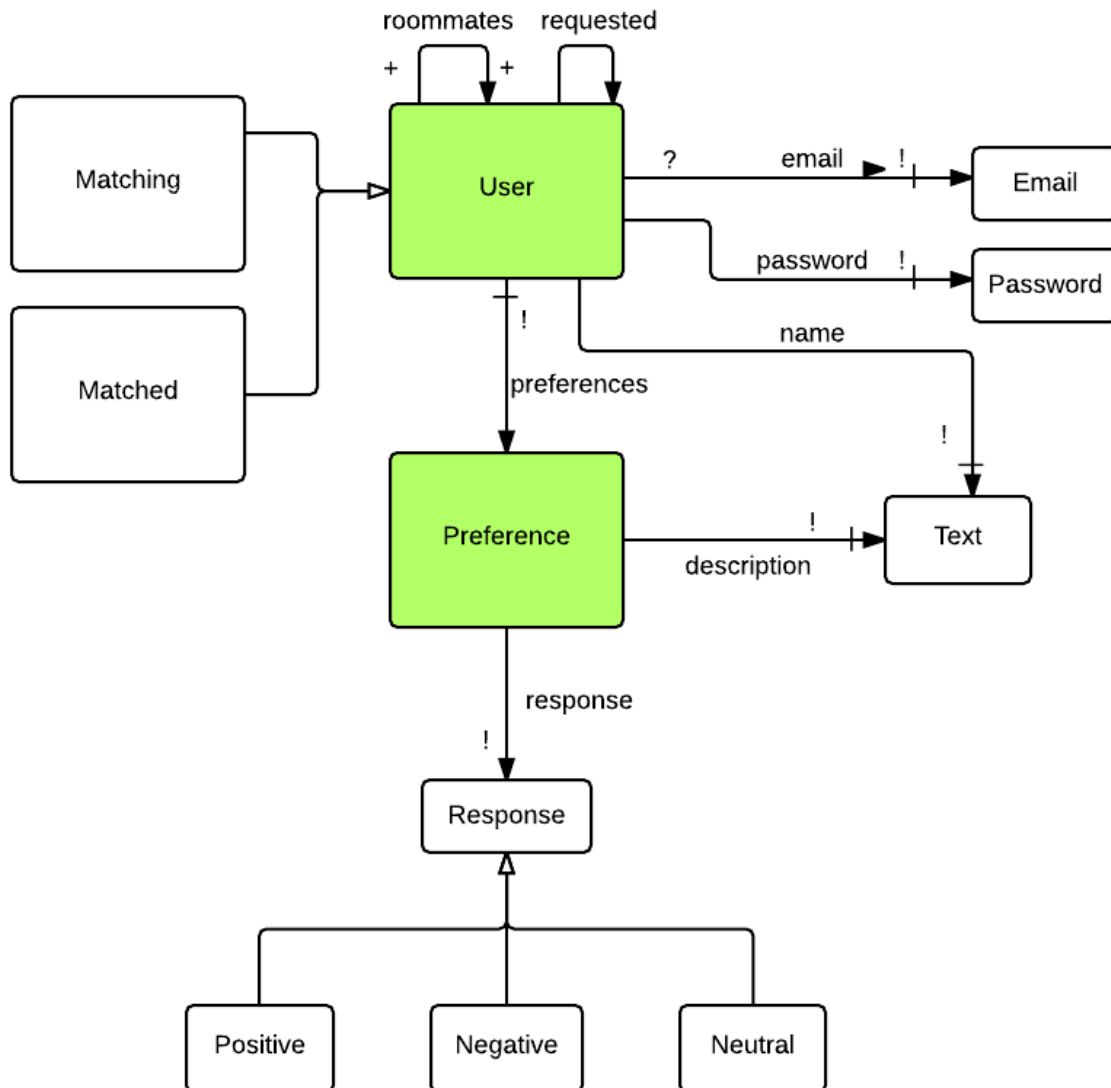


We are not using any external APIs, and will write our own authentication system.

Concepts

- Preference - each preference will have a description, and the user will input their response to the preference. Each user will enter their response to a set list of preferences. Preferences pinpoint the areas where similarity among users are good indicators of roommate compatibility.
- Suggestion - RooMIT will provide roommate suggestions based on how well the users match in terms of their responses to preferences. Suggestions can be made as soon as RooMIT receives the user responses to preferences, which helps users make a decision on roommates in a much more timely manner.

Data Model



- **User** is a student looking for roommates. A User can be matching (looking for roommates) or matched (no longer looking, taken out of the search system)
 - email: unique MIT email
 - password: password
 - roommates: list of Users that have accepted to room with the User
 - requested: list of Users that the User has asked to room with
 - preferences: list of Preferences of the user
- **Preference** is a User's opinion on a living condition. Users are matched to each other based on how similar their Preferences are.
 - description: text description of the Preference. ex: Noise at night
 - response: the User's opinion for the Preference. Can be one of: Positive, Negative, or Neutral (no opinion)

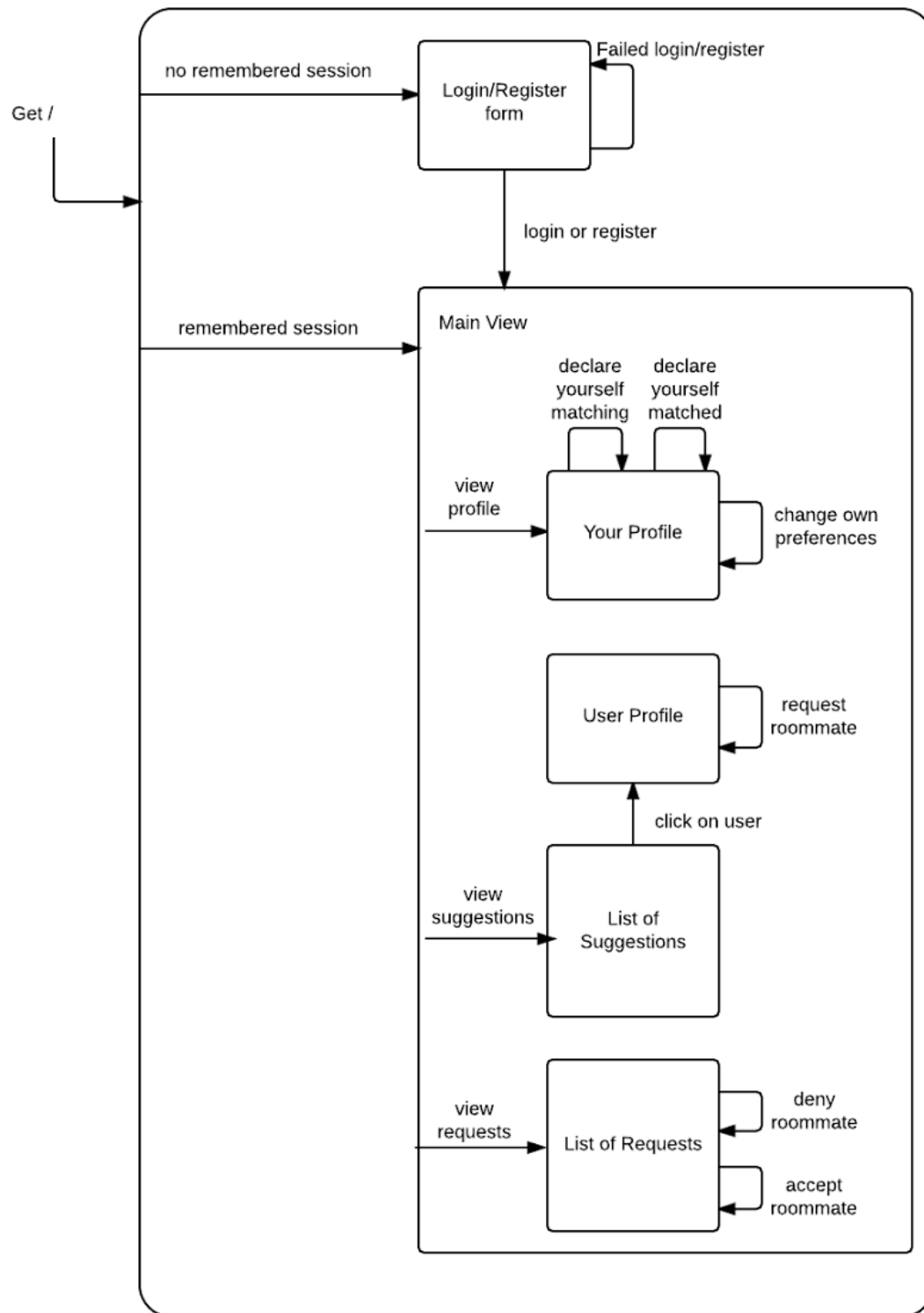
Security Concerns

- Login injection attack: we will only allow email formatted like email (client-side) and sanitize all of our login inputs (server-side).
- Preference injection attack: the options for responses to preferences will be drop down menus (client-side) and we will sanitize our inputs (server-side)
- Non-MIT users: only @mit.edu emails are accepted
- Bogus users: we hope that since we're limiting the site to MIT, we won't have this problem, but in the future we will better mitigate it by sending confirmation emails

Threat Model

- Our app isn't doing anything too crucial so we don't anticipate any serious attacks
- Attacker could gain MIT emails, but this is information anyone can access
- Attacker cannot actually modify anything on MIT Housing via our site, so at worst they just confuse students

User Interface Wireframes



RooMIT is a one-page app. We have two views: the login/register view, and the main view. The main view will display the user's profile by default, and allows navigation to the list of suggested roommates and their profiles, as well as the user's list of pending requests.

The image shows two browser window mockups for the RooMIT application. Both windows have a title bar labeled 'RooMIT' and a URL bar containing 'http://roomit-roomit.rhcloud.com'. The left window displays the login form with the text 'Welcome to RooMIT!', an unchecked checkbox for 'Make a new account', and input fields for 'Username:' and 'Password:' followed by a 'Login' button. The right window displays the register form with the same 'Welcome to RooMIT!' text, but the 'Make a new account' checkbox is checked. It includes input fields for 'Name:', 'Username:', and 'Password:' followed by a 'Register' button.

Login/Register page. Clicking the checkbox will toggle logging in/registering.

The image shows a browser window mockup for the RooMIT main view. The title bar is 'RooMIT' and the URL bar is 'http://roomit-roomit.rhcloud.com'. The page has a navigation bar with four buttons: 'Profile' (highlighted), 'Suggestions', 'Requests', and 'Logout'. The main content area is titled 'Your Profile' and includes a 'Matching:' section with a toggle switch currently turned on. Below this is a 'Roommates:' section listing three names: 'Sam Pam', 'Alex Azalea', and 'Madison Holmes', each with a blue link. The 'Your preferences:' section contains three questions, each with three radio button options: 'Do you want to live in Next House?' (Yes, No, Don't care), 'Do you want to live in MacGregor?' (Yes, No, Don't care), and 'Do you want to live with a girl?' (Yes, No, Don't care). The first and third questions have 'Yes' selected, while the second has 'Don't care' selected.

The logged in user's profile. Here they can change preferences, view current roommates, and toggle whether or not they are matching or not (searching for roommates).

The screenshot shows a web browser window titled "RooMIT" with the address bar displaying "http://roomit-roomit.rhcloud.com". The page content is for a user named "Sam Pam" with email "sampam@mit.edu". At the top right, there are four tabs: "Profile", "Suggestions", "Requests", and "Logout", with "Profile" being the active tab. Below the user's name and email, there is a button labeled "Request roommate". Under the heading "Roommates:", there are three links: "Pat Explora", "Alex Azalea", and "Madison Holmes". Below this, under the heading "Sam's preferences:", there are three rows of questions and answers. The first row is "Do you want to live in Next House?" with the answer "Yes". The second row is "Do you want to live in MacGregor?" with the answer "Don't Care". The third row is "Do you want to live with a girl?" with the answer "No". There are three sets of three dots between the second and third rows of questions. The browser window has standard navigation buttons (back, forward, stop, home) and a search bar.

RooMIT

http://roomit-roomit.rhcloud.com

Sam Pam
sampam@mit.edu

Profile Suggestions Requests Logout

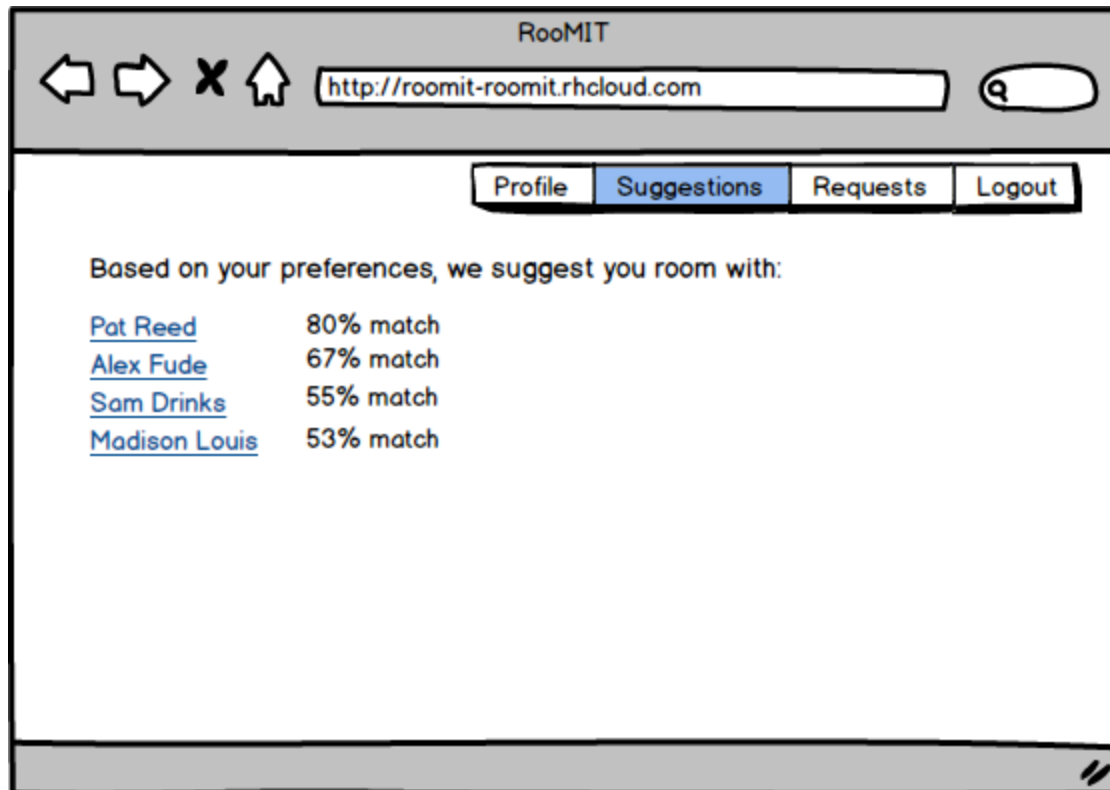
Request roommate

Roommates: [Pat Explora](#)
[Alex Azalea](#)
[Madison Holmes](#)

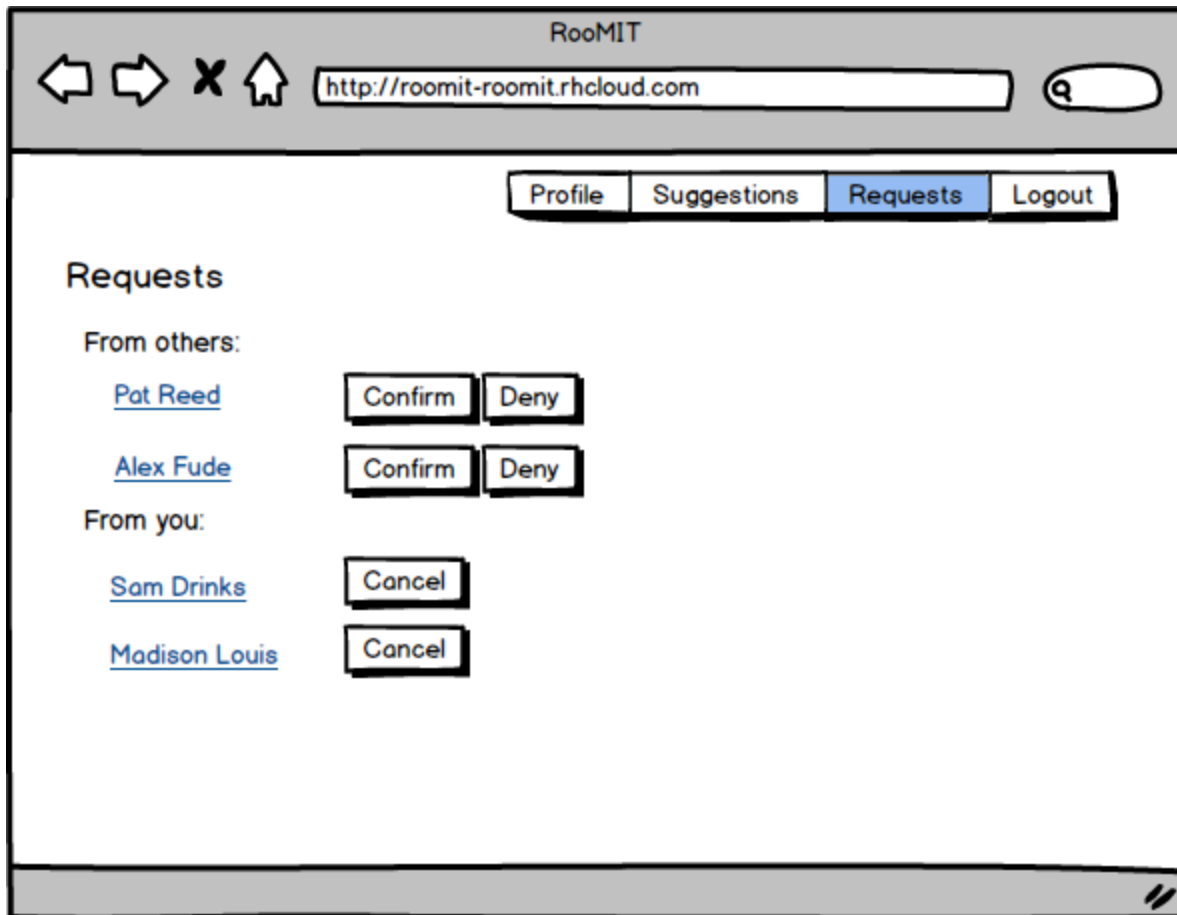
Sam's preferences:

Do you want to live in Next House?	Yes
Do you want to live in MacGregor?	Don't Care
...	
Do you want to live with a girl?	No
...	
...	

User profile. Here their contact info and preferences can be viewed, and they can be requested as a roommate.



Suggestions page. This page displays potential roommate matches. By clicking on the users' names, their profiles can be viewed.



Requests page. Here the user can approve/deny requests from other users, as well as view or cancel their own requests.

Design Challenges

- Should we incorporate dorms?
 - Options:
 - *Incorporate dorms.* It matters what dorm your roommate would like to live in, and it limits the pool of potential roommates to those who only want to live in the same dorm. Each dorm will be listed as a separate preference.
 - *Don't incorporate dorms.* Dorms are not central to our purpose and may add unnecessary complexity.
 - Decision:
 - We made the dorm you live in/would like to live in a preference, so we would only match people with similar dorm interests. This would prevent the app from suggesting roommates who in reality will not want to live with each other because of differences in dorm interests.
- How should preferences work?

- Options:
 - *Make preferences yes/no/no opinions.* This option effectively categorizes user responses and helps simplify our matching algorithm.
 - *Make preferences a scale from 1-5 and no opinions.* This allows for more nuance in user response, but will introduce more complexity.
- Decision:
 - For MVP, we made options yes/no/no opinion to simplify the matching process, but later we will make preferences 1-5 and have users mark which preferences they care or don't care about.
- What should we use for login?
 - Options:
 - *MIT certificates.* This idea is difficult to implement but would limit the users to MIT students.
 - *Google login.* This would handle security issues, but it would not limit users to MIT students.
 - *Use our own system.* This would take some time and security management, but could limit to MIT.
 - Decision:
 - We decided on the last option because we don't want to allocate the majority of our work time to implementing MIT certificates, and we do want to limit the users to MIT students, so we can't use Google login.
- How many roommates should we allow?
 - Options:
 - *Only two.* Most dorms only have singles and doubles, and this is easier to implement.
 - *Unlimited.* There are doubles, singles, and quads for now, but the system is ever-changing depending on rooming supply. This option is therefore more realistic.
 - Decision:
 - We decided to implement the latter because it doesn't add that much complexity, but improves the app's usability by a lot.
- How do we address the issue of gender?
 - Options:
 - Add a field to user for their gender orientation. The app will then only consider users of the same gender. This choice will abide by certain dorm rules that only people of the same gender can be roommates.
 - Add preferences for having roommates of each gender. This accommodates for the LGBT community and the move on campus to allow for mixed gender roommates (eg. Senior House).
 - Decision:
 - We decided on adding preferences for having roommates of each gender, because this allows for flexibility in case dorm policies on gender change in the future.