

# UROPMatcher Design

Team Open Source Fruit Tart ([opensourcefruittart@mit.edu](mailto:opensourcefruittart@mit.edu))

*Elise Xue, Meghana Bhat, Willow Jarvis, Kelly Shen*

## Overview

Most MIT undergraduates participate in a UROP during their time at MIT. However, oftentimes searching for UROPs can be difficult, and a student would have to look through many UROP listings before finding one suitable for them. Our app, UROPMatcher, is a platform where students can easily find UROPs that match their interests and professors can recruit only qualified students for their labs. The premise of the app is to match students with UROPs based on their skills, interests, and experience, and thus students can easily find potential UROPs and labs can easily find candidates who best fit the UROP position.

Current UROP search platforms include MIT's UROP listing site, UROP sites for specific courses (i.e. Course 10, Course 22), and specific mailing lists (i.e. EECS-jobs-announce). These platforms do not provide an easy way to filter UROPs, and oftentimes have outdated UROP postings for positions that have already been claimed or are not needed anymore. MIT's UROP listing site and course UROP websites also list UROPs under a single major, so it is difficult to search and apply for interdisciplinary UROPs.

UROPMatcher solves problems presented by other UROP sites by becoming a single platform for students to apply for UROPs they qualify for and that match their interests. Students create a profile based on their experience, skills, interests, etc. Professors and graduate students can post UROPs needed in their lab, and specify skill requirements, tag project disciplines, etc. Students are then matched with postings based on if they fulfill the skill requirement and if the project matches their interests. This allows students to easily find suitable UROPs without having to scroll through long listings on multiple sites, and allows labs to recruit eligible and interested students. Because our site will contain UROPs from many different departments, students can also apply for interdisciplinary UROPs. The postings on the site are also up-to-date, as old UROP postings are removed. Students also have the option to subscribe to tags so they can easily find UROPs that match their interests. Finding UROPs relevant to a student's interests is a difficult task on other platforms, but tags are used on postings in UROPMatcher so that these UROPs are more visible to students who are interested.

UROPMatcher aims to attract a large enough user base so that we do not add to the problem of becoming another one of the many UROP search platforms available. A future goal for our website is to become MIT sponsored so that we replace the old UROP listings website.

## Concepts

Note: By 'staff', we refer to users who are professors or graduate students.

Match:

- Description
  - Pairing between a student and a UROP
- Purpose:
  - Let students see only relevant UROPs
  - Let staff see only qualified students
- Operational principle:
  - If a student is subscribed to UROPs regarding a specific interest, then they will be matched with UROPs within that interest that they qualify for with their skillset.
  - If a staff member posts for a UROP with a certain interest tag, requiring a certain skillset, their UROP posting will be matched with students with that interest and skillset.
- Misfits:
  - If a student makes false claims to their skills, they can be matched with a UROP that they are not eligible for. On the staff side, their lab may end up recruiting students that do not meet requirements for the position.
    - How to Avoid: Mitigate this misfit by incentivizing students to be more honest in their skill claims. They would benefit most from applying to UROPs most relevant to their skills.

## Security

UROPMatcher does not present too attractive a target for attackers, since it will not contain too much sensitive data like health information, but passwords are stored in the database, so we take measures to ensure security. Even besides that, an insecure site would still allow attackers to alter UROP postings or user info, or force other users to run attacker-written code.

Making an account will be handled through a combination of MIT email verification and the MIT people directory API. Users will require a kerberos to make an account, and will then have to click a link to verify their account, which will be sent to their MIT email, in order to access the site. We will verify that the kerberos exists through the MIT people directory API, and also figure out whether the user is a student or professor or graduate student through that same API. By requiring these steps for access to the site, we also manage to prevent access from people who are not MIT-affiliated, as they will not be able to create and verify an account without a kerberos.

Furthermore, we use the bcrypt library to hash and salt users' passwords so that it becomes computationally infeasible for an attacker to obtain users' passwords. We chose bcrypt because it is the industry standard way to safely store passwords as it takes much longer for an attacker to crack passwords encrypted by bcrypt compared to MD5, SHA-1, SHA-2, etc.

Unfortunately, this does not prevent CSRF attacks from such non-MIT attackers. We will mitigate such CSRF attacks by limiting user sessions to at most 20 minutes of inactivity, after

which the session will time out. (We will make sure to warn users 5 minutes before any session time-out.)

We also intend to handle standard web attacks from MIT users. We will prevent XSS attacks through the use of templates. Thus, any user input data with special characters displayed on the website will be made safe through the use of escape sequences, preventing attacker-written code from being executed in that way. In combination with safe Javascript coding practices, XSS attacks should be blocked. We will also prevent database injection attacks through the use of Mongoose as a library for interacting with our MongoDB database.

### **Design Risks**

This application places a huge amount of trust on its users to use the site for its intended purpose. Placing any amount of trust in users always carries risk. We have identified a few key areas of risk and have planned to address those risks.

One, users may make fake UROP postings. To prevent most such posts, UROPMatcher will only allow professors and graduate students to make UROP postings, rather than allowing any student to do so.

Also, users may make false claims of skills or expertise. This is an issue anywhere regarding job applications, and remains an unsolved problem. The MIT students are incentivized to be honest in order to get matches relevant to them, so we do expect such cases to be outliers. In addition, a match only opens up the avenue of contact between a professor and student. The two must still go through the steps to meet and discuss the potential UROP before it becomes a reality.

The professors and graduate students also may forget to update their UROP postings once they have found students to work in their lab. To deal with this issue of information not being up-to-date, our application requires posters to specify a deadline to apply for the UROP, after which date the UROP posting will be hidden from student view. The deadline must fall before the final UROP application deadline for the semester.

There is also an additional risk of adding to the problem of having too many different UROP sites. If UROPMatcher does not achieve a large enough user base, it will end up becoming yet another unreliable place to possibly check for UROPs. In order to solve this issue, our application not only has to effectively address all the issues with the other UROP sites, but also become the official primary source of UROPs. Upon the completion of our application, we intend to contact the organizers of the current official MIT UROP site in order to replace their current mode of UROP advertisements with a link to UROPMatcher.

### **Design Choice**

Design choices List of problems to resolve in concepts, data model or user interface For each problem: options available, evaluation, which chosen

## Problem: Objective Skill Proficiency Evaluation

Description: Each student may have different standards of proficiency, so two students who have the same skill level might perceive and list their proficiency level differently on our app. As a result, students may not meet qualifications for UROPs because they under-rate their proficiency, and students may qualify for UROPs where they do not actually meet the skill requirement.

Choices:

Choice	Evaluation	Is Chosen
Rate using number (1-5)	Pros: Easy to understand Cons: Easy to have inconsistency across different users. The same level of skill might be a 5 to one user, while being a 3 or 4 to another.	No
Rate using years of experience	Pros: Concrete and deterministic metric for users to list (just need to count years of experience) Cons: Hard to evaluate in terms of proficiency; one year of experience for one student may not be equal to a year of experience for another.	No
Rate using lines of code	Pros: More accurate description of coding ability Cons: Specific to computer science majors; not applicable to other majors	No
Rate using a standard scale marked with clear definition based on amount of knowledge	Pros: Allows users to rate their proficiency based on more objective conditions. Users can more easily tell proficiency if they compare their skills against a clear definition. Cons: Students need more time to read through the proficiency level definitions	Yes  Out of the other options, having clear definitions for each proficiency level is the most flexible amongst all majors/departments, and provides a more objective evaluation of skills.

Problem: Consistency in names of skills/ tags

Description: There might be multiple skills/ tags referring to the same skill/ tag. For example, Javascript and JS, AI and artificial intelligence. We noticed that this is a problem in the skill section on LinkedIn.

Choices:

Choice	Evaluation	Is Chosen
Multiple choice pre-determined list of options, with ways to request a new type of tag/skill	Pros: Make sure there's uniformity in choice of name Cons: many fields, many new skills emerging day-to-day, needs careful curation	No
Collection of all skills/tags posted, and user objects have obj ids of the chosen skills/tags, autocomplete based on previously-used names	Pros: Let users define skills/tags as needed, make sure later users conform Cons: copies of skill/tag objects with same name for each user, flooding of the collection unnecessary	No
Pre-filled collection of already-used skill/tags, autocomplete based on names in the collection	Some curation, but it allows people to be more flexible with naming/ description, and it also lets users set standard for skill/tag names, which is both a pro and con	Yes  More flexible with naming/collection, allows for standard tag/skill names.

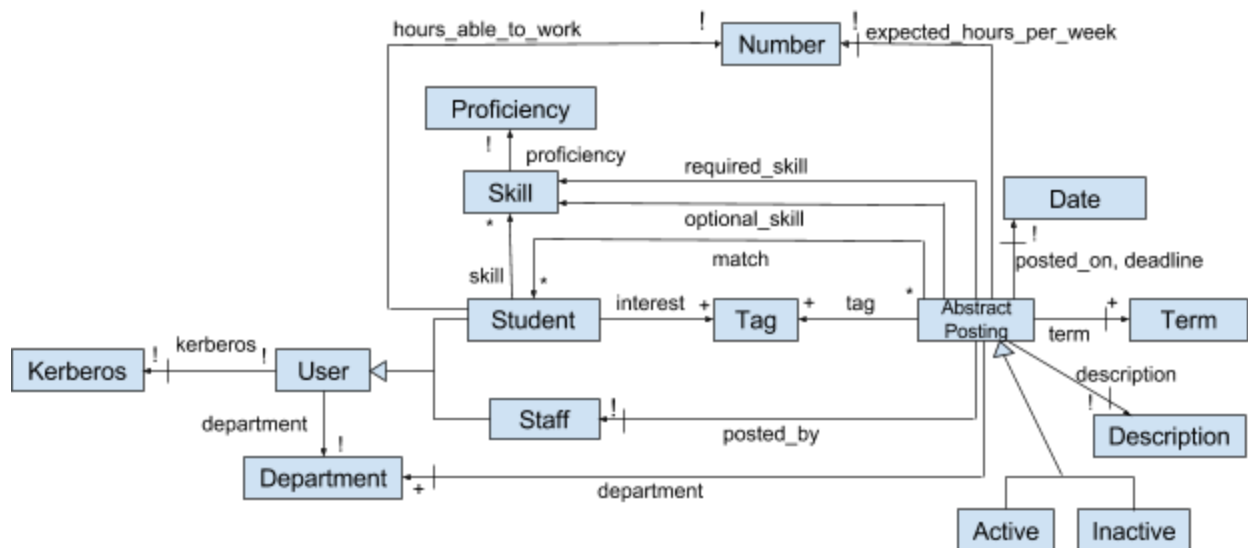
Problem: When should we delete postings?

Choices:

Choice	Evaluation	Is Chosen
Delete postings immediately after they expire	Pros: expired postings do not take up storage space Cons: users won't be able to search through old postings	No
Keep all postings in storage (but mark expired postings as inactive)	Pros: users can search through old listings; system administrator can keep a history of all past postings and matches	Yes  MongoDB allows a huge amount of storage so chances of running out of

	Cons: might exceed MongoDB storage space granted by our free account	storage is extremely low for our app and storage is cheap.
Keep all postings in storage and do a garbage collection after a time period (one, three or five academic years)	Pros: This solution compromises concerns raised from the above two choices; Cons: it might be hard to implement and does not seem necessary given the scale of our app at this point.	No

## Data Model



- Textual Constraints
  - A match is only established when a student's interests match the posting's tags and the student has the required skills.
- Explanations
  - Department is information extracted from MIT's People API using their kerberos. Staff can be a professor, research scientist or graduate student who is looking for undergraduate students to work with.
  - Proficiency of a skill is self-indicated by a student.
  - Tag represents a discipline - a student can use tags to indicate the disciplines they are interested in and a staff can use tags to indicate the disciplines the posting is related to.

- A posting can be related to as many departments as applicable (e.g. 6-3 and CMS, 14 and 18).
- Insights
  - We need an efficient and fair way to let students determine the proficiency of their skills, which we tried to express in our wireframe.
    - We use ratings with a standard scale marked with clear definition based on amount of knowledge. See “Design Choices” for more information.
  - We allowed a posting to have optional and required skills because sometimes UROP mentors are willing to teach optional skills and other times, they are not willing to supervise much.

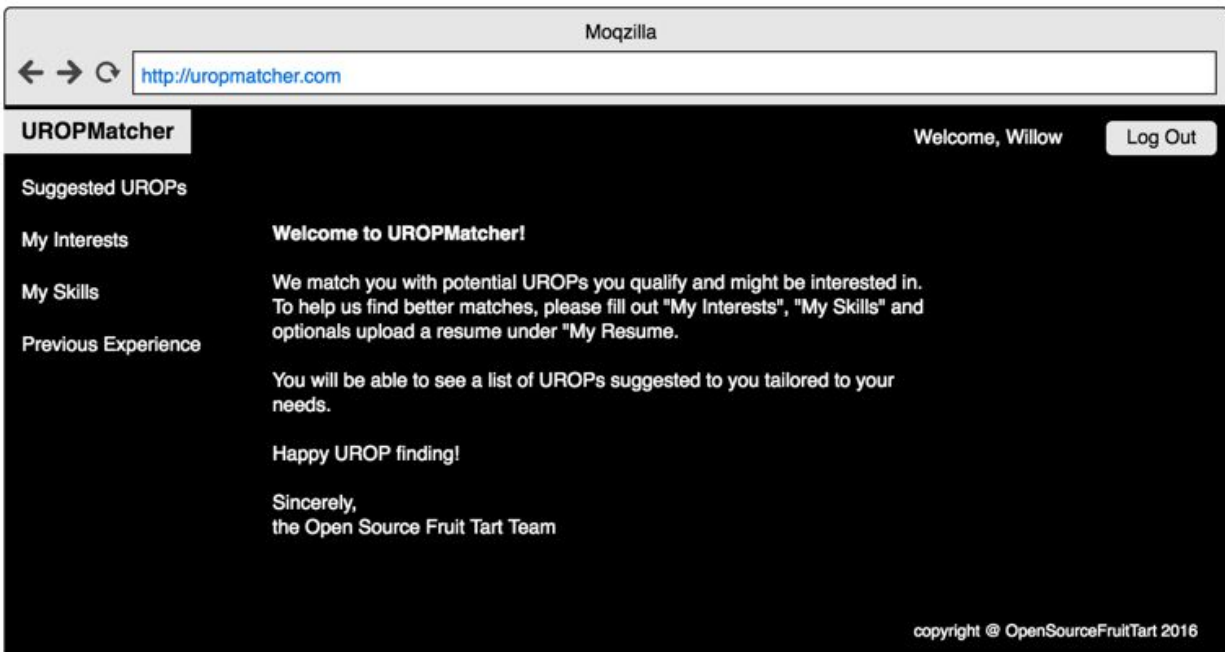
## Wireframes

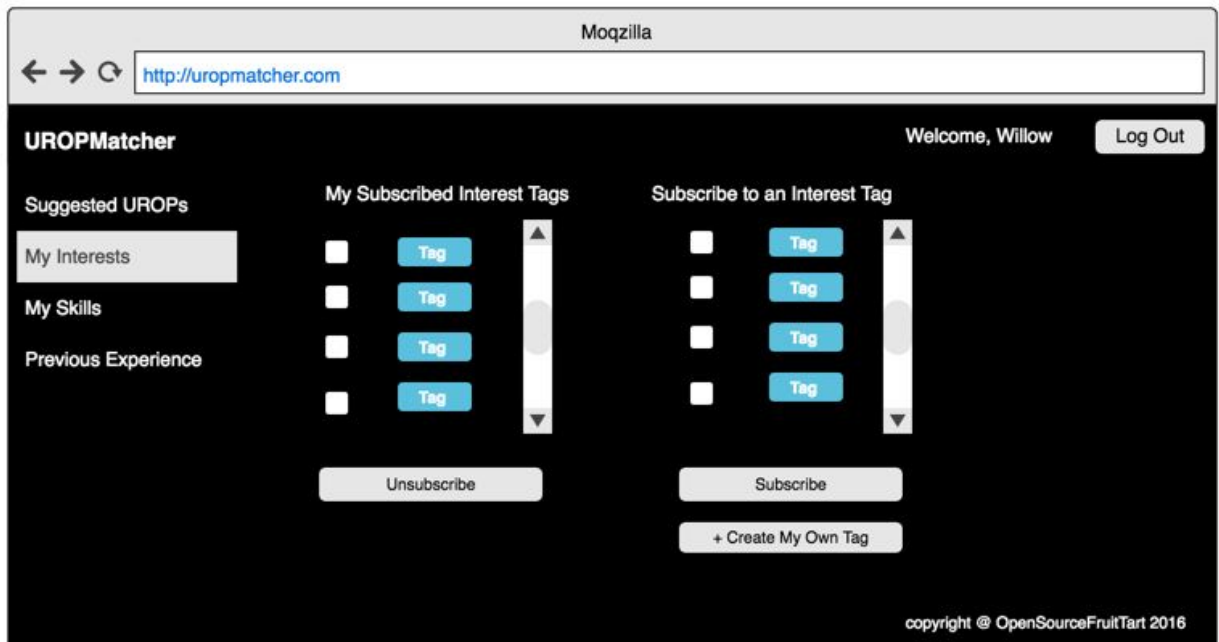
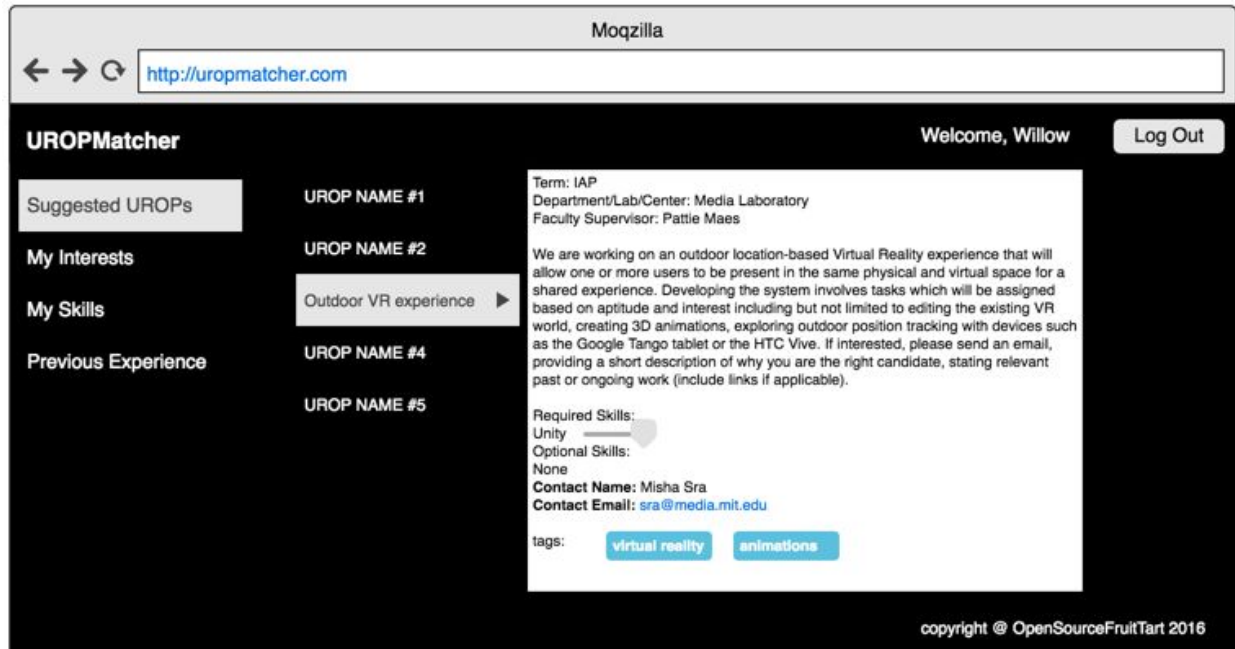
You can also find our wireframes at this link:

<https://app.moqups.com/wjarvis/C8h3kSHb01/view>. You can see specific comments by clicking the numbers in the UI mockups.

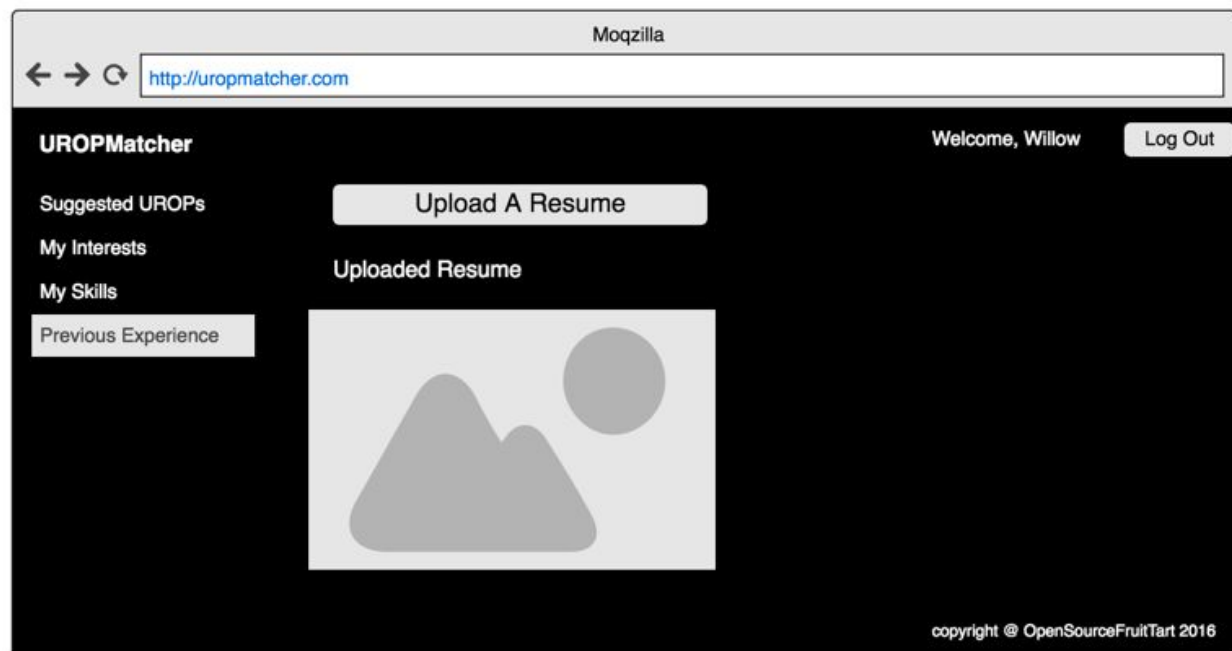
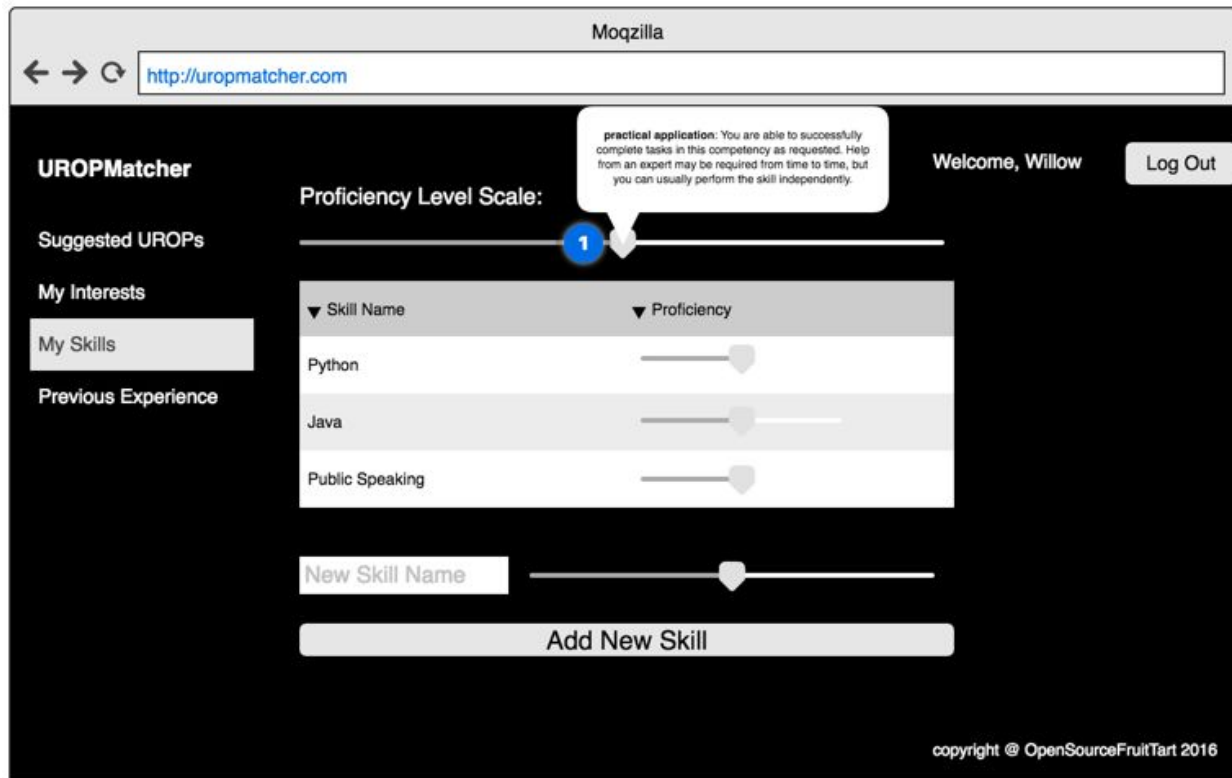
Screenshots are also included below in case the above link doesn't work.

Student Views:









## Staff Views:

Moozilla

← → ↻

http://uopmatcher.com

UROPMatcher

Welcome, Misha

Log Out

My Posted UROPs

Post A UROP

UROP NAME #1

UROP NAME #2

Outdoor VR experience ▶

UROP NAME #4

UROP NAME #5

Description

Suggested Students

Term: IAP  
Department/Lab/Center: Media Laboratory  
Faculty Supervisor: Pattie Maes  
  
We are working on an outdoor location-based Virtual Reality experience that will allow one or more users to be present in the same physical and virtual space for a shared experience. Developing the system involves tasks which will be assigned based on aptitude and interest including but not limited to editing the existing VR world, creating 3D animations, exploring outdoor position tracking with devices such as the Google Tango tablet or the HTC Vive. If interested, please send an email, providing a short description of why you are the right candidate, stating relevant past or ongoing work (include links if applicable).  
  
Required Skills:  
Unity  
Optional Skills:  
None  
  
Contact Name: Misha Sra  
Contact Email: [sra@media.mit.edu](mailto:sra@media.mit.edu)  
  
tags: 

Tag

Tag

I already found someone for this UROP

sourceFruitTart 2016

Moozilla

← → ↻

http://uopmatcher.com

UROPMatcher

Welcome, Misha

Log Out

My Posted UROPs

Post A UROP

Title

Term

Department/Lab/Center Name

Hours Per Week Needed

Description

Seeking 1 Ps in Department/Departments:

Required Skills:  

Python

+ Add A Required Skill

Skill Name

Add This Skill

Optional Skills:  

+ Add An Optional Skill

Contact Name

Contact Email

Tags:  

#robotics

2 animation

Post This UROP

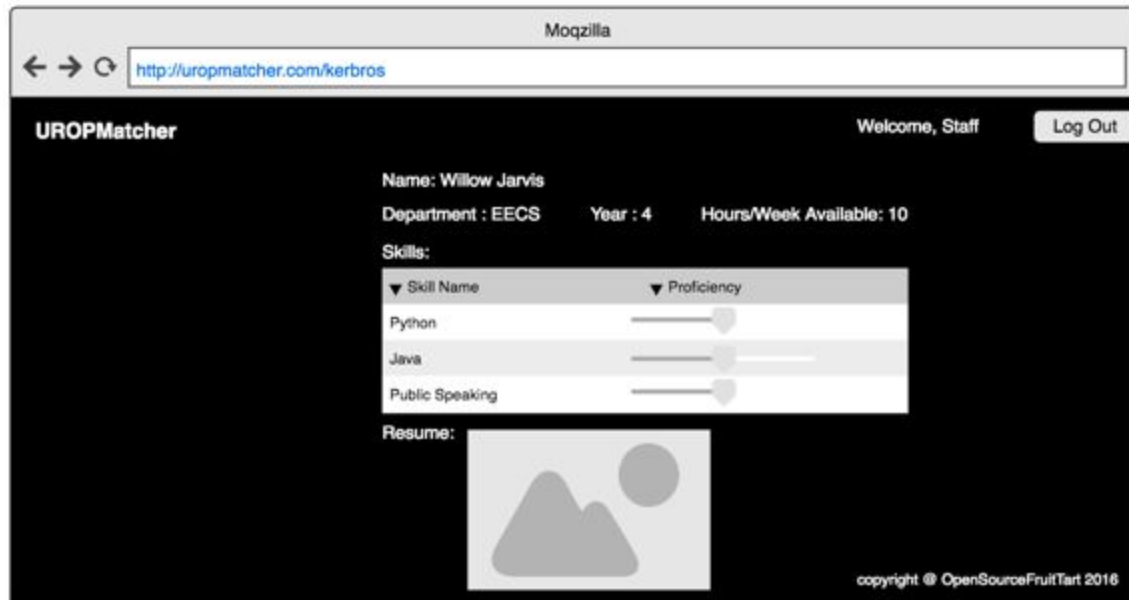
3 Add to Removal Deadline

April 22, 2012

Su	Mo	Tu	We	Th	Fr	Sa
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

OpenSourceFruitTart 2016

Student Profile Page: When UROP seekers click on the matched student's name, he gets redirected to that student's profile page (as seen below).



## Revisions

We initially planned to authenticate users through the use of MIT certificates, which would allow us to avoid the more insecure user-password authentication and make sure that users were MIT students (verifying their credentials as student/staff at the same time). However, we could not implement this since we would have needed a scripts site, which does not support Node.

Therefore, we came up with an alternative scheme to allow us to verify MIT affiliation. We now ask a user to sign up with their MIT email address, which we will check against the MIT People API to make sure that they are a student or a staff of the institute, and that they didn't sign up for the wrong role. We then send a verification email to the user, and after they click the link in the email, their account will be created.

We also made a change to our back-end handling of skills and tags. Initially, we planned to have skills and tags as their own collections, with postings and user objects referring to skill/tag objects. However, we decided instead to embed them in those posting/user objects, and then make separate collections of already-used skill/tag names for skills/tags, solely for the purpose of having a dictionary for autocompletion of user input. The collections will be prefilled by us with example skills/tags, e.g. Java for skills and Machine Learning for tags.

## Conclusion

UROPMatcher addresses all of the key issues identified with current ways to find UROPs. Students no longer have to manually search through many irrelevant or outdated postings in order to possibly find one of interest. In addition, they will more easily be able to identify labs

within their interest with openings that they qualify for, instead of fruitlessly sending out a large amount of emails to different labs. Professors no longer have to go through the hassle of identifying which UROP sites or email lists would bring them the most success with their target groups. Instead, we ease the process of connection between students and professors by matching based on key factors, limiting misfits and security risks along the way. Our design will be the new portal for finding and recruiting for UROPs.