

Group Assignment: Final Report

Computer Architecture Student Association

Team: MissRevsDevs

Elba Garza, Chen Chuan Chang, Charles Chou, Johnson Hung, Philip Shih, Peter Wu, Ruiyu Qu

Team Roles:

Product owner: Elba Garza

Scrum master: Peter Wu

Summary

The Computer Architecture Student Association, or CASA, is a student-led group within the research area of computer architecture. As part of their initiatives for creating a more inclusive community for students in the field, CASA currently holds multiple short-term mentoring programs at conferences, and is currently working on the creation of a long-term (i.e. 1 year) mentoring program. CASA recently published a workshop paper at ISCA 2021 which outlines their goals for all forms of mentoring in the area of computer architecture. Currently, short-term mentoring sign-ups occur at each conference registration, and registrants must *re-register* and *re-fill* their information at every iteration.

CASA hopes to have infrastructure (e.g. mentor/mentee accounts, matching algorithms, mentoring progress reportings) in place for short-term mentoring programming at both conference mentoring events, and for long-term mentorship. An infrastructure which saves participants' information via user accounts/profiles would more easily let students and mentors participate at various instances of mentoring events without having to re-fill information every time. Tracking previous matches between users may also ensure we do not repeatedly match the same mentee and mentor pair. Similar infrastructure for long-term mentoring would help CASA track the efficacy of mentoring relationships, and ensure they are beneficial to both parties, or may require re-matching.

The GitHub repository: <https://github.com/johnson-hung/casa>

The Pivotal Tracker account: <https://www.pivotaltracker.com/n/projects/2536066>

The Heroku deployment: <https://stormy-refuge-76501.herokuapp.com/announcements>

Implemented User Stories

Feature: View announcements (3)

As a participant

So that I can know what the newest information is
I want to see announcements through the home page view

Feature: Clickable Sign-up Button (3)

As a participant

So that I can create a user account
I want to fill out the personal information

Feature: Clickable Log-in Button (3)

As a participant

So that I can log in my account
I want to fill out the username and password

Feature: User Log in (3)

As a user

I can login in my account
I want to fill out the email and password to log in

Feature: User Sign Up (3)

As a participant

So I can create a user account
I want to fill out the personal information (e.g. name, email, password, research interests)
I want to validate the fields in right format
I want to store the password safely.
When I fill in "First name" with "John"
and I fille in "last name" with "john"
and I select the mentor
And I press sign-up
Then I should see an error

Feature: Clickable my account button (2)

As a user

So I can go to my account page
I want to go to my account page and edit my personal information
If I am a organizer

So I can see create announcements and events

Feature: My account for participant (3)

As a participant

So I can see my personal information

Feature: Sign out (2)

As a participant

When I select "Sign out"
I should not see "my account"

Feature: View Events (1)

As a participant

So that I can see events that have not yet happened
I want to see the events through the home page view

Feature: Event Sign-Up Page (3)

As a participant

So that I can sign up for the mentorship at the events
I want to verify user information and submit the request

Feature: Export user event (3)

As a organizer

So I can receive user information for one specific event
I want to download a csv file that contains user information about participants who register this event

Feature: My account for organizer (3)

As a organizer

So I can see my account
And I can find a button to create announcements
And I can find a button to edit ann oucemnts
And I can find a button to create events
And I can find a button to edit events
And I can export the user event

Feature: Edit user information (3)

As a participant

If my sign-up information is outdated
I want to edit my user information

Feature: Edit events for organizer (3)

As a organizer

If my event information is outdated
I want to edit event information

Feature: Create events for organizer (2)

As a organizer

So I can create a event
I want to fill out the information about event (e.g. name, time)

Feature: Add Research Interests (3)

As a organizer

So I can create new research interest options
I want to add options in user sign up page let user have new more research interests option

Changes made for user story:

*The most important functionality of the CASA mentoring website is for participants to be able to have an account to sign up for mentoring events. That is, a participant (either a potential mentor or mentee, doesn't matter) should be able to create an account, or sign in to their account if they had already previously created one. Having functioning user accounts is very important because it helps participants sign up for mentoring events quickly without having to re-fill their information each and every time. Every mentoring event will be an instance for participants to sign-up for. Participants are **not** automatically signed up for each and every mentoring event. At event sign-up, participants will be able to verify or update their information before submitting.*

In Iteration 0 user stories described the capability of embedding Twitter accounts, sharing the website itself via the Twitter API, or having an organizer post announcements. These are all secondary compared to having users create and/or log into their accounts.

The last user story, of having an organizer post announcements is necessary, and implies need for two different account types: organizers and participants. There are a few organizers compared to possibly dozens or hundreds of participants. This differentiation will necessitate further user stories that focus on organizer functionalities.

For the organizer implementation, we set up a privileged account which has extra power to create, edit, delete, and export the announcements and events. These functionalities were implemented in the events edit pages.

Customer meeting date

Time: 9:00-9:30 AM

9/16: First meet up. We discussed what are important user stories.

10/28: Extra communication with the client also occurred on October 30th, where we discussed the new user stories and their higher priority relative to some of the ones we'd shared before.

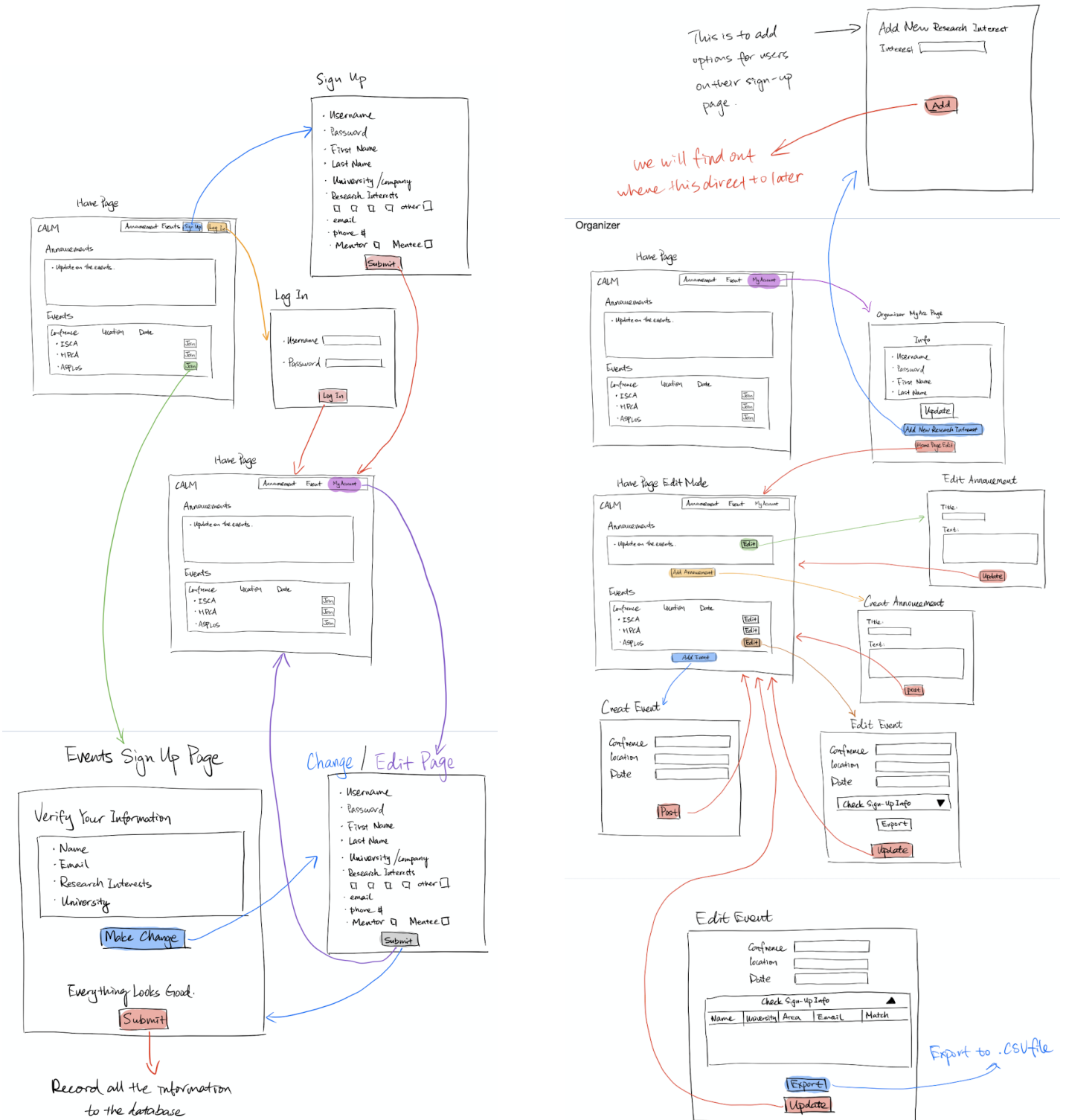
11/4: We have created a home page and template database which contains data about announcements and events. At the home page, the user can see the announcements. We also discussed the layout of the home page and how to implement user login in the page.

11/11: In this meeting we discussed the further implementation of the user account. We decided to separate the normal users and organizers. The normal users should only see the given information. On the other hand, the organizer has extra privilege to create, edit, and delete announcements and events. We also demonstrated further detail for user personal information. We let the organizer have the right to add more research interests, in order to meet the requirements.

12/2: At today's meeting, we presented two features, user sign up and announcements/events post. For the user sign up page, customers required slightly modifying CSS and layout. Besides, we discussed how to implement a privileged mode for organizers to have extra power editing and posting announcements/events.

12/9: We demoed our final product to the client, and discussed future work that needs to be done.

Lo-Fi User Interface



Scrum Iteration

Iteration 1

Points completed: 8 points

Accomplished	Story Points
View Announcements	3
View Events	1
Clickable Log-in Button	3
Clickable Sign-up Button	3

Iteration 2

Points completed: 9 points

Accomplished	Story Points
User Signup	3
User Login	3
Sign Out	2
Add Research Interests	3

Iteration 3

Points completed: 19 points

Accomplished	Story Points
Clickable My Account Button	2
Edit User Information	3
My Account For Participant	3
My Account For Organizer	3
Event Sign-up Page	3
Create events for organizer	2
Edit events for organizer	3

Code Testing

By doing BDD/TDD, we were able to verify all the webpage implementations had their corresponding pathway. We also verified all the buttons and links were able to redirect correctly. Difficulties we had with BDD/TDD included what was necessary to define as a given and what was not. We learned the hard way that we must describe all that is necessary to be given, or else we could not access it. Another aspect was accessing data that was in our database, but was not part of our seeding information/given information. Learning the relationships between the data and BDD/TDD was most essential.

Configuration and Branches

For version control, we always create new branches for features and testing so that the master branch stays consistent. There are 24 branches in total where most of them with completed functionality were merged into the master branch. Although our branches were mainly set for essential testing and feature development, we still had a time when a spike branch was created to help us understand rails routing. We have two major releases: one is the initial application deployment, and the other is the current version with desired functionality integrated.

Issues encountered

The Heroku command line tool is straightforward and easy to understand, but we still face some issues when developing:

First, we followed the instructions on Heroku website, but we still cannot deploy our app to Heroku. After our survey, we found that Heroku expected that the ruby application must be located on the root folder, and we put our application inside the other folder instead. Second, the database we developed locally is sqlite. However, the database run on the production environment is PostgreSQL. It took us some time to realize that the connection to the database should be established by psql instead of sqlite client.

We encountered merge conflicts when using GitHub. After this experience, we understand how to manage different features by using different branches and how to set up merge requests to prevent accidentally merging other branches.

Additional Tools Used

Additional GEMS:

Factory_bot_rails: for login

ZenTest: for testing

Simplecov: for code coverage analysis

Pg: PostgreSQL driver

Links to your poster video and demo video

<https://www.dropbox.com/sh/e7uefs4hacm9iy3/AAD4x5OdOixCah0opkaFta3Na?dl=0>