TEAMWORK

MITjobtalk: Timmy Galvin, Mari Miyachi, Josh Duncavage

PLAN

Goals

We would like to develop a job forum for the MIT community, where students can openly share their interview and work experiences, ask questions, and better understand the job market. Thus, we have designed a system the focuses around company objects and querying amongst them. Information is organized by company, not by user contribution, such that users can remain anonymous if they please.

Stakeholders

Currently we are the biggest stakeholders in this project since we are building this independently from any organization.

Resources

We imagine our final product not being resource intensive, but we can imagine that processing all of the submitted data and generating dynamic graphs/diagrams could be our most demanding operation.

Tasks

- Design analysis, all members, Thursday Nov. 15
- Create basic framework, all members, Friday Nov. 16
- Certificate login, Timmy, Monday Nov. 19
- Interview forum, Timmy, Tuesday Nov. 20
- Offers forum, Mari, Tuesday Nov. 20
- Q/A forum, Josh, Tuesday Nov. 20
- Search by company, Timmy, Friday Nov. 24
- Sort interview examples, Josh, Friday Nov. 24
- Multiple answers for questions, Mari, Friday Nov. 24
- Graphics for offers, ???
- Upvote answers, ???
- UI polish, ???
- UI details, ???
- Security concerns, ???

Risks

We expect the biggest risk will be individual accountability. We have divided deadlines into individual and group components, and obviously if one individual does not meet their deadline, the group deadline will be late. To mitigate this, we will meet regularly and well before deadlines are due. If one group member is seriously struggling to keep up with the workload, we will reassign tasks.

For this to be viable we will also need some initial data. Not having already acquired the data could be slightly risky in terms of the success of the project.

Minimum viable product
Our minimum viable product will be mainly an implementation of our object model. It should allow users to:

1. Search for companies and create companies if they don't already exist
2. Add interview and offer examples for each company
3. Add questions for particular companies and answer questions for particular companies
4. View interview, offer, and Q/A information
5. Login with MIT email addresses

TEAM CONTRACT

Each team member is expected to complete the tasks by the day previously assigned, and no later. Work will be divided as evenly as possible, so there is really not a lot of room for flexibility in terms of deadlines. Granted, we will discuss and create deadlines far in advance to give everyone fair notice.

Mari:

> I want to make a product that people actually use. Ultimately, Facebook integration would be nice. I'm also hoping that we will be able to effectively divide the work, as that has always been a challenge for me in group projects in the past.

Timmy:

> I really want to be able to tie in everything I've learned from the past three projects and make something I'm proud of. I'm hoping my teammates can help make up for my lack of front-end skills. As with most projects at MIT, I'm concerned the work distribution won't be even, but our team has gotten the ball rolling early to ensure there is a fair division.

Josh:

> I think this is a project that could actually be useful for a good chunk of MIT. Hopefully user engagement will allow us to get a decent chunk of statistics and discussion that will prove useful to the rest of the MIT community. While individual accountability is always a concern for group projects, I think the key to a successful group project is constant communication between the members. This way, everyone can be on top of their work, and also be on the same page as to what direction the project is moving and the end design goals we have in mind as a team.

Team meetings will be held twice a week, on Tuesdays and Fridays, at the student center. Here, we will discuss progress on tasks, any challenges we are running into, and upcoming deadlines. Meetings will hopefully be no longer than 45 minutes.

We all have goals to get high marks in this class, so we hope that quality will be maintained by our individual standards. However, we will also try an informal code review scheme where during our Friday meetings, we each look at another member's code and give brief feedback.

If a deadline is missed, the other team members may need to do extra work to compensate. This is far from ideal, and the other member would potentially have to do extra work during a later component of the project to make up for their tardiness.

Minor design decisions will be left up to the person writing the code for that particular section, but all major design decisions (i.e. those that are not isolated to one section) must be discussed with the group via email or in person. We hope to entertain all opinions, but in an effort to be efficient, all decisions will be decided by majority.

MEETINGS

November 11

Agenda

- Discuss problem analysis, particularly question areas
- Times for meetings in the future

Progress Report

- Working on problem analysis and thinking about slides
- Plan to meet and finish problem analysis/slides by deadline
- Unclear about proper OM representation

Minutes

- Adjust minimal viable product
- Rework OM
- Ensure good communication and teamwork

November 19

Agenda

- Discuss individual tasks for the upcoming week
- Touch on specific challenges we foresee in the upcoming week
- Integrating security features
- Defining our database structure

Progress Report

- Working on login, running into trouble with certificates so for now implementing a email authentication system
- Basic scaffolding for MVP, including interviews, offers, questions, users
- Working on best way to implement UI features like adding new information

Notes

- How to deal with company duplicates? Maybe only person who creates company can edit it. Or no one can, alternatively
- How to ensure accurate information? An admin account maybe, but that could pose a security threat
- UI details: wireframes need to be updated, elegant error messages

November 25

Agenda

- Demo our app as it is now and discuss what needs to be done for Tuesday
- Talk about security design decisions: only can add company if search query yields no results
- Discuss user control over editing information: only can edit info you've submitted yourself
- Go over edited design docs

Progress Report

- MVP implemented: can add companies, interviews, offers, questions, answers
- Search working for companies
- Will be using email authentication for sure - system fully implemented and secure
- Exploring options for graphics

Notes

- Pre-populating the DB would be good - some sort of pre-population script versus moderator
- Use placeholders and scales for interview, option to add if you got the job or not in the offers section
- Different positions will use hashtags
- Josh will work on company search and offer graphics
- Timmy will work on manual testing, small ui tweaks and ensuring a secure experience
- Mari will fix pathing issues, email the Career's office about data, do small ui tweaks

Dec 2

Agenda

- Demo new improvements.
- Discuss what needs to be finished for the deadline
- Ask questions about design decisions
    - Pre-populating the site
    - Admin account
- Go over breakdown of work per person

Progress Report

- Complete UI overhaul; got rid of scaffolding and bootstrap feel
- Ajax/JQuery now fully functional
- Still struggling with finding a good graphical representation for offers - many APIs don't offer the dynamic functions we're looking for
- Splitting up the work has been difficult - we addressed this issue among our group and also with our mentor

Notes

- CrunchBase API for pre-population
- Graph package: d3.js
- We will be using tagging for company roles
- Mostly need to just clean up the code for the next deadline and add commenting and unit testing

Dec 9

Agenda

- Demo our app as it is now and discuss what needs to be done for final submit
- Discuss expectations for our design docs
- Discuss in-class demo tomorrow

Progress Report

- Graph implemented using d3
- Filtering system via job positions for interviews
- Improved forms for interviews that contains more guidance
- Still unsure if Admin account is the proper way to monitor site activity

Notes

- Store position variable after entry in pop-up form. Also make the pop-up form auto-resize to correct size.
- Add better UI on searchbar full list of companies, narrow down search.
- More intuitive for expanding interviews to see questions.
- Scripts for auto populating database.
- Offer graphs need better offer title
- Security concerns => Admin needs to have more constraints. Use a regular user account, and for admin privileges have a separate additional login where they enter a code.
- Add normal comments in addition to method specifications.
- Move more logic from the controller into the model.

REFLECTION

Peer review

We discussed our performances as a team, as we felt that our group was close enough that we could be candid about the feedback.

Timmy

Timmy was a real asset to our team, which was pretty inexperienced with Rails and web development. He consistently submitted the deliverables assigned to him in a timely manner, and made himself available to meet with the group. He was in charge of deploying our site to Heroku, and did a great job of finding and fixing bugs involved with deployment. Timmy also was responsible for a lot of the backend work, and was always willing to help debug the code of his teammates.

Mari

Mari did a lot of the initial coding and documentation, setting the stage for the rest of the project. She also worked on many of the UI improvements, along with the sites AJAX features. She submitted work on time and created the group's agendas, ensuring that progress was made on a weekly basis. She also worked on a lot of the commenting.

Josh

Josh was given some difficult assignments initially, such as the offers graphic, and after the deliverable for this was not met, we all discussed what had happened. Our group realized we had underestimated the difficulty of this task, but also that the division of the workload needed to be adjusted. After this

discussion, Josh was very present in the group's coding, and worked on front-end features and testing.

Evaluation

We succeeded in a couple of aspects: setting clear goals on a weekly basis and defining a reasonable scope for our work. From the start of the project, we met on a bi-weekly basis, once with our mentors and once with just each other. During our personal meetings, we would create assignments and deliverables, such that when we met with our mentors, we would have weekly progress to display. In this respect, we were able to create short term goals that were achievable and fairly evenly divided.

We struggled at times with finding the right balance of work between our team members. After bringing this up with our mentor, we spoke as a group about the dynamic, and the situation did improve dramatically. At the end of our project, we made the decision to conduct more work together, which we found to be very helpful in terms of efficiency and productivity.

Certain larger goals may have been lost in the fact that we made goals each week. Our weekly tasks were pretty narrow, and it seems we may have been capable of achieving more had we created goals that lasted a little longer than a week. After seeing the demonstrations of other groups in the class, we were a little disheartened. We certainly put the 200+ man hours into this project, as expected from the course and as demonstrated from our commit history and meeting agendas. However, it seemed that even with this time commitment, our final product falls short of what other groups achieved. This is disappointing, but as a relatively inexperienced group, we feel that this project taught us more about Rails than the individual projects and was a very beneficial learning experience.

Lessons learned

We learned three main lessons from this project:

1. It is important to create both long and short term goals. Our group succeeded in creating and fulfilling short term goals, such that every week we were making significant progress. However, it would have helped us to look at the bigger picture and try to think of features not as we wanted to add them incrementally, but on a larger scale.
2. Confrontation is key to settling group dynamics. We faced some issues earlier in the project regarding the delegation of the workload. After speaking about the problems together, and being as open as possible, the situation noticeably improved and everyone benefited.
3. Coding together makes a difference! It's great to have someone to help debug and to bounce ideas off of. Next time we engage in group projects we will definitely keep this in mind.