



RoleForSkill

~*~

Executive Summary/Proposal

16OCT14

David Holiday

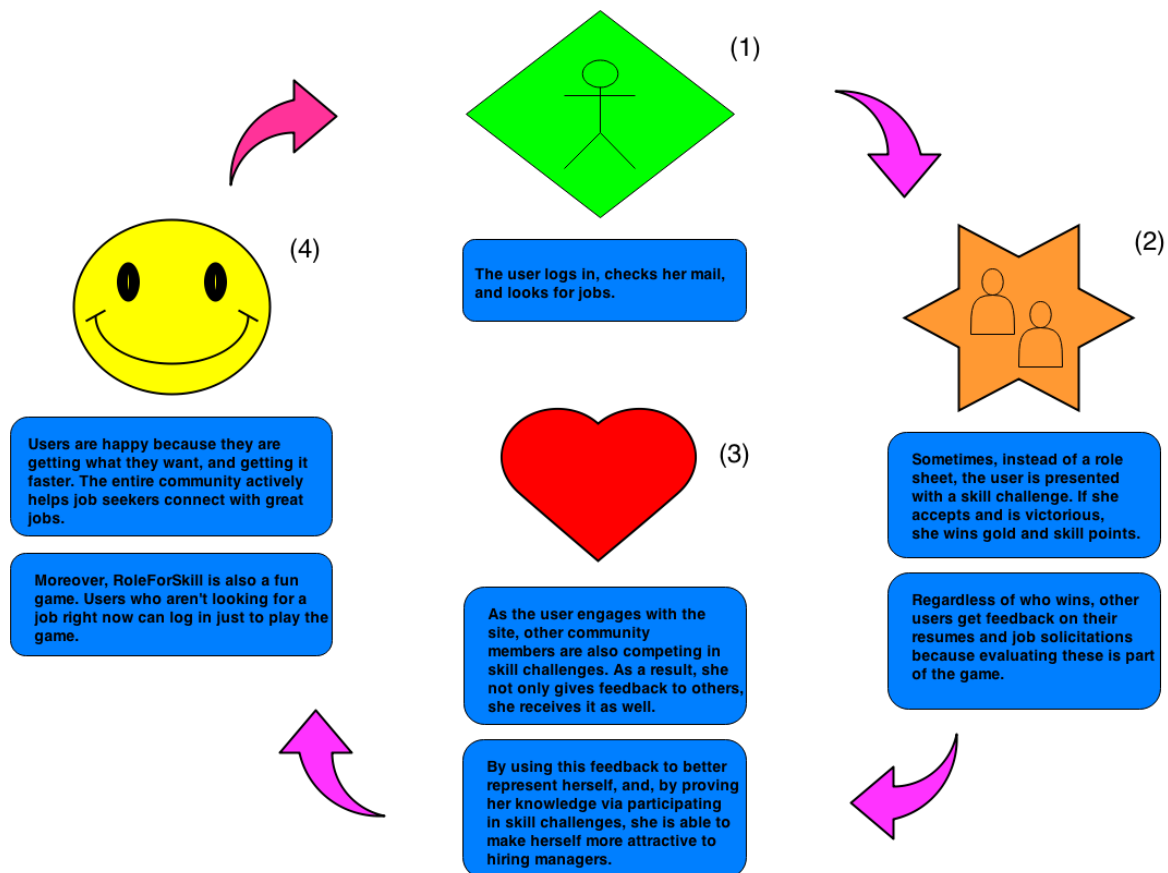
WHAT IS?

RoleForSkill is a lightly gamified job community that leverages the power of community action to quickly connect technology professionals with the jobs they want.

HOW DOES IT WORK?

RoleForSkill employs gaming elements to incentivize users to establish credibility, provide constructive feedback on information submitted by other users, and match role solicitations with skilled candidates. Specifically, the incentivization comes visa-a-vis a competitive trivia game that randomly pairs users of similar ability against one another. Time is a factor in this game in that trivia questions must be answered in a matter of seconds, and items (in-game objects earned by the user which help in playing the game) need to be recharged prior to subsequent use. The game mechanics of RoleForSkill are similar to those employed by 'freemium' games like 'Simpsons: Tapped Out' and "Clash of Clans".

USER EXPERIENCE



The first thing a new user does is create a profile for the avatar that will represent her on the website. This entails picking a skill class, creating a skill sheet, and if applicable, uploading a resume. The minimum amount of effort a new user must expend to pass this point is purposefully kept low so as to encourage engagement. As such the only thing a new user must do is select a skill class.

A skill class is a general category delineating a general professional group the user identifies with. Users can be classified as:

- recruiter (any hiring professional)
- tester (someone who writes code to test production code)
- coder (someone who writes production code)
- operations specialist (anyone working with IT infrastructure)
- support specialist (help desk, technical writer, trainer)
- manager (project manager, department head)
- acolyte (someone still in school or very new to the profession)

Once a user has picked a skill class, the next step is to create a skill sheet. A skill sheet has all the same information that a resume would have, only the information is laid out so a person's present skills are the primary points, and everything else is ancillary. For example, on her resume, a talented C++ programmer would likely feature her education, certifications, previous work experience, and projects with equal prominence. On a skill sheet, her acumen with C++ would be featured prominently. Everything else would be listed as subordinate information there to provide context in support of the user's claim of proficiency. This because when an organization has a role to fill, what they ultimately are looking for from a candidate is an answer to the question 'what value can you immediately bring to the organization if you get the job?'. While a resume does contain this information, it's often buried among other information that's not germane. The purpose of the skill sheet is to give recruiters an executive summary of what the candidate can bring to an organization, right now.

Lastly, if the user selects any skill class other than recruiter, she is given an opportunity to upload a resume. As part of the upload process, she is given the means to easily redact her personally identifying information from the resume document. This is to protect users from spam, and other harm that may come by featuring her home address, telephone number, and other personal information on the internet.

At this point our user is all set. She is presented with a menu bar, a search filter, and two toggle switches. On the menu bar are options to manage her profile, manage her item inventory, engage with the community, and view messages. The search filter allows her to cull the stream of role sheets (or skill sheets if she's a recruiter) to her liking. The toggle switches control the two types of content she sees. One controls whether or not she sees role sheets (skill sheets if she's a recruiter). The other controls whether or not she wants to participate in skill challenges.

By default both content switches are engaged, so we'll follow the thread of user experience as such. The user sees, one-at-a-time, role sheets (or skill sheets if she's a recruiter) that she can review, save for later, or respond to. Randomly, instead of a role/skill sheet, she'll be presented with a challenge which entails a head-to-head trivia game against another player.

THE GAME IN GAMIFY

During the challenge, both players are asked the same seven questions germane to computing. The person who gets the most correct answers wins gold and skill points. Gold can be used to buy items, which give you a competitive edge. Skill points are used to level up your avatar, which gives you access to cooler items, harder questions, etc. Lastly, it should be noted that outside the context of skill challenges, there are specific tasks a user can undertake that will earn them stars. These can be used to add time to trivia question timers, make items recharge instantly, and be converted to gold. Unlike gold or skill points, stars can be directly purchased directly by the user.

Of the seven questions, two are related to the user-submitted data. These take the form of either a role/skill sheet evaluation, or, an evaluation of a proposed match between the two (ie – does this roll sheet represent a job that's a good fit for a person with these skills?). Both players are asked two questions with pre-defined answers associated with both. The goal is to answer the questions the same as your opponent.

The purpose of this is to provide feedback to both recruiters and job seekers as to the perception the community has regarding what they've put into the system. By framing the correct answer to the question in terms of how someone else might respond, we make it more likely the answer given will be an honest reaction. This, in turn, makes the feedback more valuable to the person receiving it, and the value proposition of RoleForSkill higher than that of other job sites. Professionals seeking employment get itemized feedback regarding their resumes from hiring managers. Recruiters get feedback as to how job seekers feel about aspects of their job solicitation. This process, in effect, entails the community working together to quickly match technology professionals with the jobs they want.

WHAT DOES IT MEAN TO LEVEL UP? WHAT INCENTIVE IS THERE TO DO SO?

Leveling up has two effects. First, it grants the user access to greater rewards, different skill challenges, and more privileges on the system. Second, and perhaps more importantly, skill challenges are the means by which a user can establish credibility. For example, put yourself in the shoes of a hiring manager. Your job solicitation got 200+ responses in the first day of it being posted. Most of the candidates have college degrees, work experience, and certifications. Who among them is the hungriest for work? Which ones are the sharpest candidates? Who is the diamond-in-the-rough?

When users participate in skill challenges, they're not only playing a game, they're establishing credibility by proving they have the knowledge and skills listed on their skill sheet. Moreover, if a recruiter is particularly interested in a candidate, they can pay to target questions specifically to that person. In this way, users can be vetted for positions much earlier in the hiring process, and recruiters have a more effective means of finding candidates with niche skill sets.

WHAT IF SOMEONE DOESN'T WANT TO PLAY?

Users are in no way forced to play the game. With a single mouse click, all the user will see are jobs and job seekers. Alternatively, if all they want to do is play, they need only click a different button, and RoleForSkill becomes a fun technology-based trivia game.

HOW IS THIS DIFFERENT THAN A JOB SITE?

Legacy job sites like Dice, Monster, and others, all work in essentially the same way. Recruiters pay for the right to post job solicitations to the website. Job seekers are given the ability to create a profile, upload a resume, and search for solicitations for free. The job site makes money by providing recruiters access to job seekers. Recruiters make money when they place a candidate in a role. Job seekers make money when they find a job.

The problem with this model is that, on paper, it's very difficult to tell which candidates and which job solicitations are of good quality. Neither the hiring manager nor the candidate have any knowledge of the other, except for whatever information is contained in a resume or a job solicitation. As such, job seekers and recruiters alike are systemically encouraged to spam each other in the hopes of getting noticed because intrinsic to the way legacy job sites function is a complete lack of ability for anyone to differentiate themselves based on merit. Additionally, and also germane to the issue of lack of information, nobody has any way of knowing how their resumes or employment ads are being perceived by the target audience. Nobody has any way of knowing whether or not the manner in which they are representing themselves is effective because the only feedback they get is whether or not someone responds.

RoleForSkill is revolutionary in that it addresses both of these challenges, and, makes the whole process a lot more fun. It incentivizes users to work together to provide constructive feedback to their peers, it more quickly matches the right job to the right candidate, and it provides a way for everyone to differentiate themselves based on merit.

WILL THERE BE A MOBILE APP AS WELL?

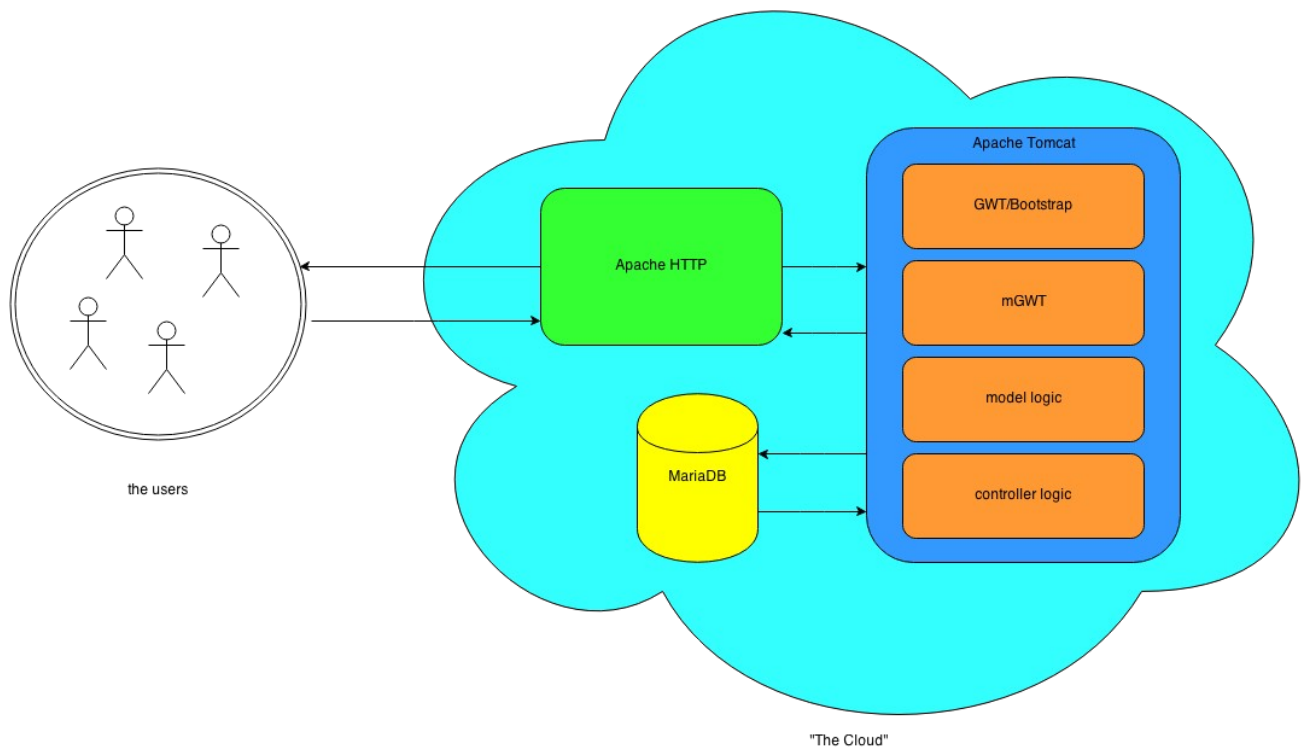
Yes! Certain architectural decisions have been made (which will be discussed later in this document) specifically to facilitate deployment to both iOS and Android. Moreover, one of the reasons for displaying to users only one solicitation at a time is to provide a unified, delightful user-experience across all platform types.

IN GENERAL TERMS, WHAT IS THE BUSINESS MODEL?

Here, again, RoleForSkill works a bit differently than a legacy job site. From a business perspective, a benefit of gamification is a greater degree of flexibility in terms of building streams of revenue. Users who want to place job ads on the system will still be charged, but at a substantially reduced rate vs any other job site. The bulk of RoleForSkills income will be

sourced in users purchasing items that enhance their ability to find what they are looking for on the site. For example, a hiring manager may have an opening for someone with a specific set of skills. As such she may want to target a subset of the job seekers with her solicitation. She may also want to ask specific challenge questions of specific candidates to vet them for the role prior to making contact. Lastly, she may also want to ask specific candidates specific questions about the solicitation itself so she can get a better idea of how her target audience is perceiving the ad. All of these things will occur in a random fashion for free, but if our recruiter wants targeted data, she will need to pay for it.

WHAT IS REQUIRED FROM AN ARCHITECTURAL STANDPOINT?



Architecturally, the stack will consist largely of battle-tested, open-source, Java technologies. The code itself will be built with Maven and tested via TestNG. The database will be MariaDB, which is an open-source fork of MySQL. Apache Tomcat instances will be setup and will handle the heavy-lifting (interfacing with the database, service logic, etc). In front of that will be an Apache HTTP server. The presentation to the user will be handled by GWT/Bootstrap, which translates Java to Javascript and provides stable, attractive, cross-browser experience. Every aspect of the stack is well tested, secure, and proven to handle substantial workloads if need be. Moreover, by utilizing the Model-View-Controller design paradigm, GWT/Bootstrap can easily

be replaced with another UI solution without disturbing the rest of the code. Lastly, once the system has been stood-up, MGWT can be incorporated to deploy RoleForSkill to the iOS and Android platforms as native apps.

WHAT DOES THE MVP CONSIST OF, WHAT'S THE TIMELINE, AND WHAT IS THE COST?

The MVP (Minimum Viable Product) consists of everything referenced in this document except for mobile app deployment. This does not mean, however, that the website won't render well on mobile devices. One of the many reasons GWT was selected for this project was that it automatically renders content in the best possible way for the platform to which content is being delivered.

From a technology standpoint, the core functionality can be achieved by a single developer in eight-to-ten months. The real challenge lies in rigorously testing the game elements to ensure maximum efficacy. As such, the play-testing phase will likely continue well beyond the time-frame proposed in this document.

In broad terms, the time-line is as follows:

wk0-wk6: Stand up instances of Tomcat, Apache Webserver, MariaDB, and use GWT/Bootstrap to create basic 'coming soon' landing page. Create RoleForSkill 'bible' containing database entity diagram, game mechanics, user stories, and design mock ups.

wk6-wk14: Build core back-end functionality to manage users, their submissions, items, etc.

wk14-wk26: Build core back-end functionality to manage game elements.

wk26-40: Play testing.

Because the majority of the work will be undertaken by a single developer, the cost will be only \$35,000 plus any incidental costs (travel expenses if an in-person meeting is desired, stipend for hosting services, etc).