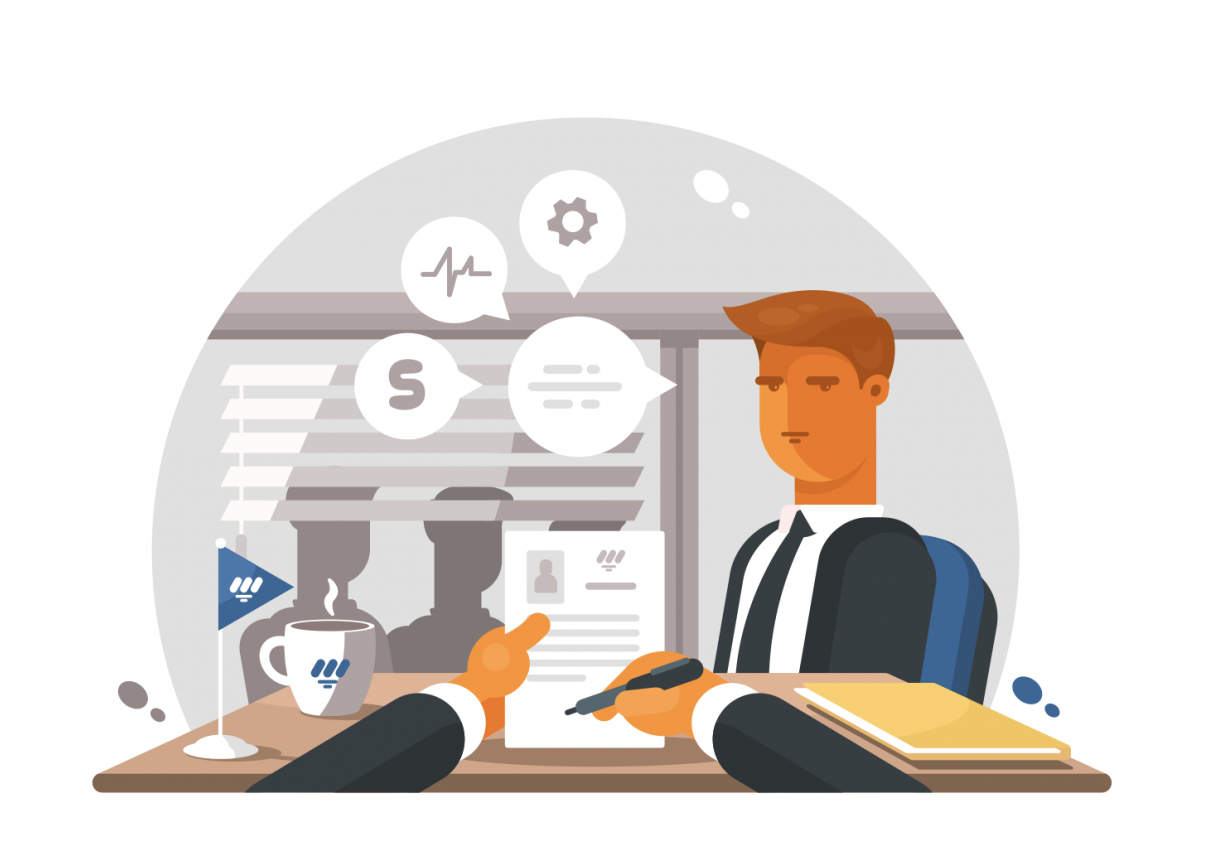
*Jobsearch!*



A Project by Larry Ruh for Larry Ruh

[What is Jobsearch!? 3](#_Toc9690884)

[The Methodology 4](#_Toc9690885)

[“One Man Scrum” 4](#_Toc9690886)

[The Tools 4](#_Toc9690887)

[Project Management Information - taiga.io 4](#_Toc9690888)

[The Web Application Technologies 5](#_Toc9690889)

[HTML, CSS, Javascript 5](#_Toc9690890)

[PHP 5](#_Toc9690891)

[JQuery 5](#_Toc9690892)

[MySQL 6](#_Toc9690893)

[MAMP 6](#_Toc9690894)

[Selenium 6](#_Toc9690895)

[Git, GitHub 6](#_Toc9690896)

[Jenkins 6](#_Toc9690897)

[DevOps 6](#_Toc9690898)

[The Business Area 7](#_Toc9690899)

[High Level Requirements 7](#_Toc9690900)

[The Core Team 8](#_Toc9690901)

[Scrum Master 8](#_Toc9690902)

[Developer 10](#_Toc9690903)

[Title Page 10](#_Toc9690904)

[Job Application Search 11](#_Toc9690905)

[Add/Edit Job Application 11](#_Toc9690906)

[Recruiter Search 13](#_Toc9690907)

[Add/Edit Recruiter 14](#_Toc9690908)

[Company Search 15](#_Toc9690909)

[Add/Edit Companies 16](#_Toc9690910)

[QA 18](#_Toc9690911)

[What’s Next? 18](#_Toc9690912)

[Who Is This Larry Ruh Person anyway? 18](#_Toc9690913)

[Where can I get this awesome system? 18](#_Toc9690914)

[Disclaimer 18](#_Toc9690915)

# What is Jobsearch!?

I was laid off in July of 2018 and thus embarked upon my new job of finding a job and learning a new industry, the job search industry! As I learned and my job search evolved, I had lots of information to keep and tedious tasks I had to perform and being an IT manager, naturally started thinking about how I can make this more efficient and try to automate some parts of it, because, well, that’s the shiny pebble in the pond for me, efficiency and consistency. There are a few aspects to this Jobsearch! side project that I wanted to point out.

1. I wanted to use this to demonstrate a hands on approach to all aspects of the SDLC using Agile methodology. I have been involved in many aspects as a developer and Systems Analyst and all aspects as a manager but wanted to get my hands really dirty and take it from A to Z all myself, I’ll get into the details of this throughout the document.
2. While I was managing, I stupidly allowed my technical skills to get rusty, I used to be a web programming ninja and wanted to use this project to sharpen my old skills and learn some new ones that have become mainstream since I moved into management and stopped programming. I’ve found that my analytical and fundamental programming skills are fully in-tact and I am having an absolute ball writing this code and using these new tools and languages, I wish they had them when I was programming.
3. Lastly, I wanted to use this project to demonstrate to potential employers that after 27 years in IT, I know my may way around, not just programming but all aspects of the SDLC and Agile from technical, business, project management and service delivery aspects, not to mention my management skills which is a big part of who I am but not really part of what this project was about (after all, I was only managing myself here ☺ ). This was to get my boots back on the ground. Incidentally, it all came back to me very quickly how much I loved to develop applications and would go back to an App Dev position in a heartbeart! **So…potential employers….I’m smart, experienced, hardworking and extremely hungry. I am currently employed but not doing what I love to do which is both develop and manage, so I continue to look in those areas.**

Jobsearch! is a small project consisting a few different systems to address the following issues during my job search that I had, I’ve only started and still have a long way to go. It has a few areas worth briefly mentioning before we get into the details.

1. Jobsearch Functionality:
   1. I was searching multiple sites for multiple positions in multiple cities, that’s a lot of searches, so I wanted to write something that would hit the job boards’ APIs based on my search criteria and return and consolidate results so I could see and link to all jobs relevant to me in one place. That would save me hours in searching from site to site.
   2. I was using a spreadsheet system to document jobs I applied for and as my list grew, it was getting ugly to find out what positions were still open, closed, etc from state to state. I needed to see, sort, filter, enter and edit all jobs in all states in one place.
   3. I wanted to be able to export these jobs I applied for to an excel spreadsheet based on filter criteria for recruiters who want to know where you’ve applied.
2. Networking Functionality:
   1. I was talking to a lot of recruiters and was using spreadsheets to track those relationships also. Needed a better way to do that.
   2. I was keeping an eye on many companies and had established relationships at those companies, also keeping in spreadsheets, needed a better way to do that.

# The Methodology

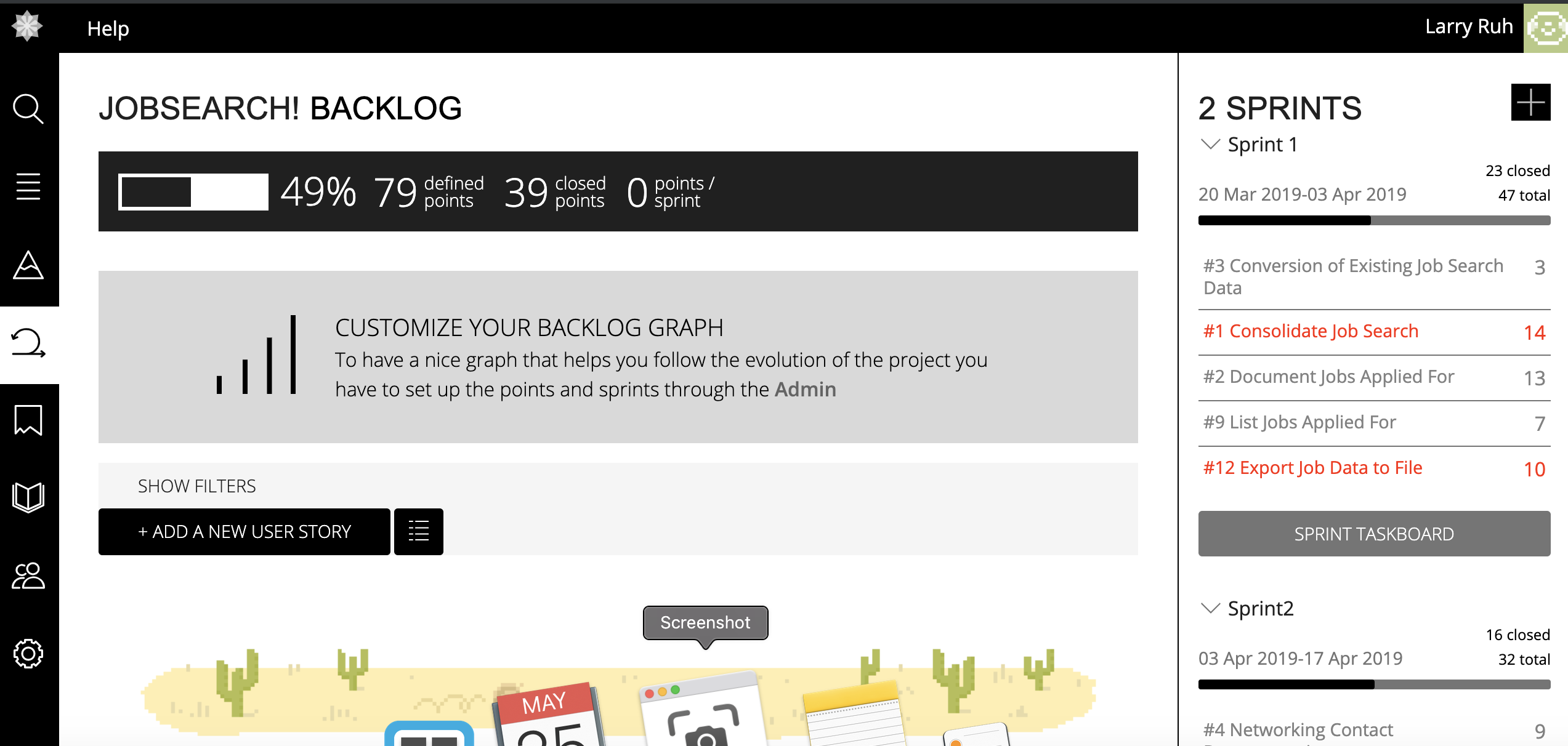
## “One Man Scrum”

Since I was alone in my endeavor and I wanted to demonstrate all aspects of the Agile scrum process, I had to wear many hats (actually, all of them), Stakeholder, Product Owner, Scrum Master, BA, developer, QA analyst and automation engineer and Configuration or Implementation manager. Hence the “One Man Scrum” Methodology. Now I did not hold formal kickoffs, standups or sprint review meetings with myself but I did have backlog refinement and sprint planning sessions where I would jump back and forth between different roles and try to analyze things from all perspectives. It was a fun exercise, yes, I’m a geek!

# The Tools

## Project Management Information - taiga.io

This is a great online (free) Agile project management site that I found. It allowed me to define team members, epics, stories, sprints, log & track issues, it has a storyboard and a Wiki to elaborate and document, etc. All the things a PMI system should do and I found the interface very easy and intuitive to use. Here is a screenshot of the dashboard page.



## The Web Application Technologies

I kept the web application pretty basic, it consists of the following tools: HTML5, CSS, JQuery, PHP, mySQL. I used Brackets as an editor (which was awesome), and phpMyAdmin for the DBMS console.

A word about a few of these technologies:

HTML, CSS, Javascript – Been working with these since the mid 90’s. Obviously paying a little more attention the CSS these days as its becoming more robust and integrated in most platforms, but still using it mostly to apply style and formatting.

PHP – Back the day, ASP was my tool of choice and I could pretty much do anything I wanted with ASP and javascript and few ancillary tools (maybe COM, or third party components). Since ASP is gone now, I had to find a new server side tool and PHP sure fits the bill. It seems to be a pretty robust language that gives the programmer a lot of flexibility without a ton of overhead. I thought it was pretty easy to learn and documentation was abundant on the Internet and stackoverflow. I can definitely get done with PHP what I used to do with ASP, just need get past the first time use pain points and I’ll be rockin’ and rollin’.

JQuery – I Love Javascript! JQuery makes Javascript a little easier to write which I like but at the same time it kind of takes away the need to take the “long way around” which leads to a deep understanding of the DOM. I like the way JQuery allows you to gather tags, classes, etc and act on them en masse with a single function call or attribute assignment. Also like the way it handles events all in one place so you don’t have to follow the flow of functionality all over the page from in tag event function calls.

MySQL – This was my first project using MySQL and I found it very easy to use, pretty standard RDBMS functionality and it performs well using ANSI SQL so all good there. I used phpMyAdmin for the interface it was fine, basic, very workable. I’ll get into MySQL Workbench on my next project.

MAMP – I used MAMP to host my application locally on my MAC, loved it! Need more of this sort of bundling. Easy to install, invisible to use. I plan on moving into Java web services in a little bit and I’m hoping Apache handles those without the insane configuration issues that come with full blown linux based web server installation.

Selenium – Next on my list is to write a full regression suite and automate using Selenium Webdriver.

Git, GitHub – I have 2 repos that I maintain on github for this project, jobsearch\_dev and jobsearch, with the latter being the “Production” Site. I have this set up locally the same way with 2 folders under my localhost. My aim here is to become intimately familiar with GitHub and deploying files from one environment to another.

Jenkins – I will use Jenkins to automate changes and deployments, I will incorporate running automated functional and regression suites before deploying, then moving files (maybe using xldeploy) and updating repos.

DevOps – This is an area that fascinates me so I’m setting this up for hands on experience in automating the implementation process. My plan is to run enhancements through the dev site and have an automated process to deploy to production and test. When I get this done, I’ll be able to sleep at night ☺

# The Business Area

Putting on my Stakeholder hat, there were a few things I wanted to make easier while searching for a job, as my time is valuable, automating or reducing time taken to do tedious things I needed to do would allow me to spend more time applying, training, interview prepping, etc. that’s the value I would derive from this system.

I wrote a High Level Requirements document outlining this functionality that can be found at github.com/larryruh/jobsearch/hlrequirements.docx. Its pretty short and to the point so here it is:

## High Level Requirements

Job Search Business Process

1. Search Internet Job Sites for Jobs
   1. Consolidate job search criteria and sites
      1. Ability to search selected states
      2. Ability to search selected Job Titles
      3. List all in one place
2. Apply for Jobs
   1. Manual Process
   2. Maybe be able to generate a custom resume and CL from a template.
3. Document Jobs Applied For (Add, Edit)
   1. Provide a form to enter new, and edit existing jobs applied for.
4. Conversion of existing Spreadsheet.
   1. Import existing excel file into new data structure.

Networking Business Process

1. Document Companies I want to keep an eye on
   1. Provide a form to enter new, and edit existing Companies to monitor
2. Document Recruiters & Contact Info
   1. Provide a form to enter new recruiter and contact info.

Reporting Requirements

1. List of all jobs applied for based on certain criteria
   1. By State
   2. By Date
   3. By Status
   4. By Job Type
2. Export List to Spreadsheet and download
3. List Recruiters
   1. Provide search criteria
   2. Should provide all recruiter info, including an email link and link to their company career site
   3. Should provide a link to be able to edit.
4. List Companies
   1. Provide filter (or search criteria)
   2. Should provide all company info, including a link to their career site
   3. Should provide a link to be able to edit.

# The Core Team

Scrum Master – With my Scrum Master and Stakeholder hats on, I broke those high level requirements into Epics, there were 3 of them with the following stories in them:

### Epics and Stories

1. Job Search Business Process – EPIC - Items that had to do with searching, applying and documenting job application.
   1. Story - Consolidate job search
   2. Story – Document jobs applied for
   3. Story – Conversion of existing job search data (structuring DB and importing spreadsheet data)
2. Networking Business Process – EPIC – Items pertaining to documenting recruiter and Company Contacts
   1. Story – Networking contact documentation
   2. Story – Document companies to watch for jobs
3. Reporting – EPIC – Making it easy to find information I have documented.
   1. Story – List jobs applied for
   2. Story – List recruiters
   3. Story – List Companies
   4. Story – Export Job Data to File

I won’t list all the story text here but here is an example of the Consolidate Job Search story:

*As a job searcher, I want to be able to see jobs from different job sites all in one place based on my search criteria so that I dont have to navigate to every site and search at every location for every job.*

*Acceptance Criteria:*

*The following search criteria should be able to be entered:*

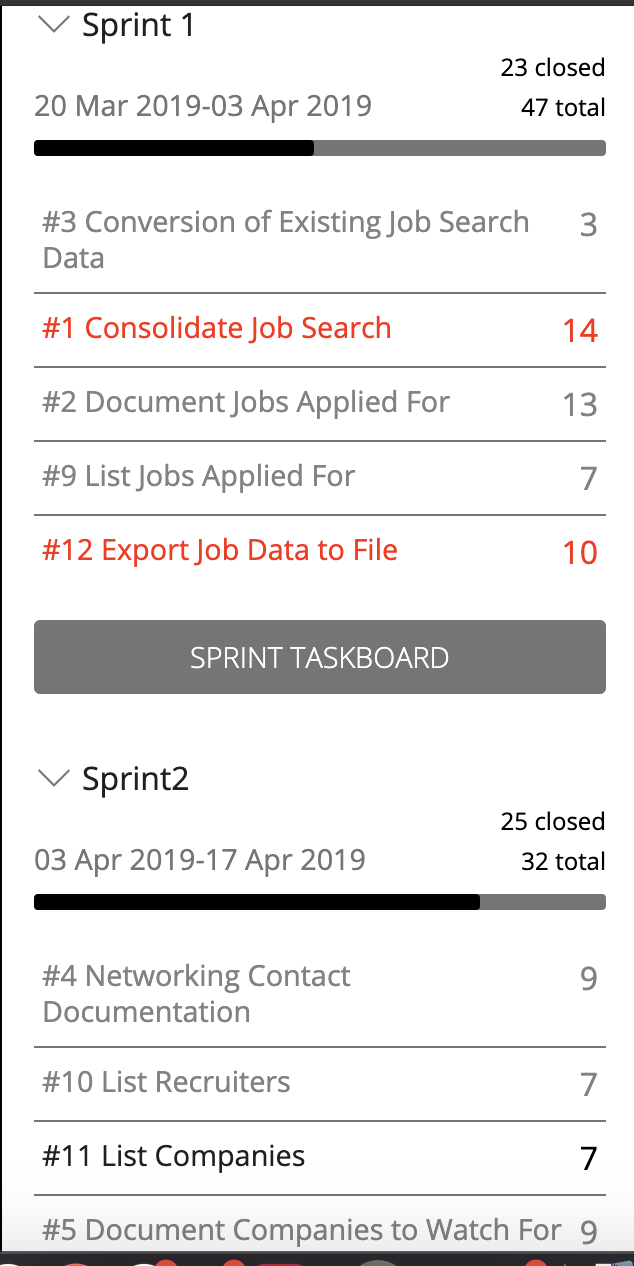
1. *Job Title*
2. *Locations to be searched*
3. *Company to be searched*
4. *Job Sites to be searched (optional)*

*Results should be easily readable and easy to navigate.*

*Jobs should be able to be saved on the job sites (if API functionality allows)*

### Sprints

The “core team” thought that 2 Sprints should be sufficient. Sprints were kind of tough to define, since I didn’t have a real team and I only worked on this when time allowed, velocity was, let’s say inconsistent, the normal 2 week sprint could have no work done and yield 0 points or lots of work done and yield all points in the sprint. It was also different as the pointing tool on Taiga.io broke down pointing into 5 sections, UX, Design, Front, Back, Product Owner. So by the time I allocated points to all those categories, each story ended up being more points than I would have given it if I was playing pointing poker or using other pointing methodology but in the end, points are arbitrary and the more points allocated, the higher the velocity when it gets done. Here is the Sprint breakdown:

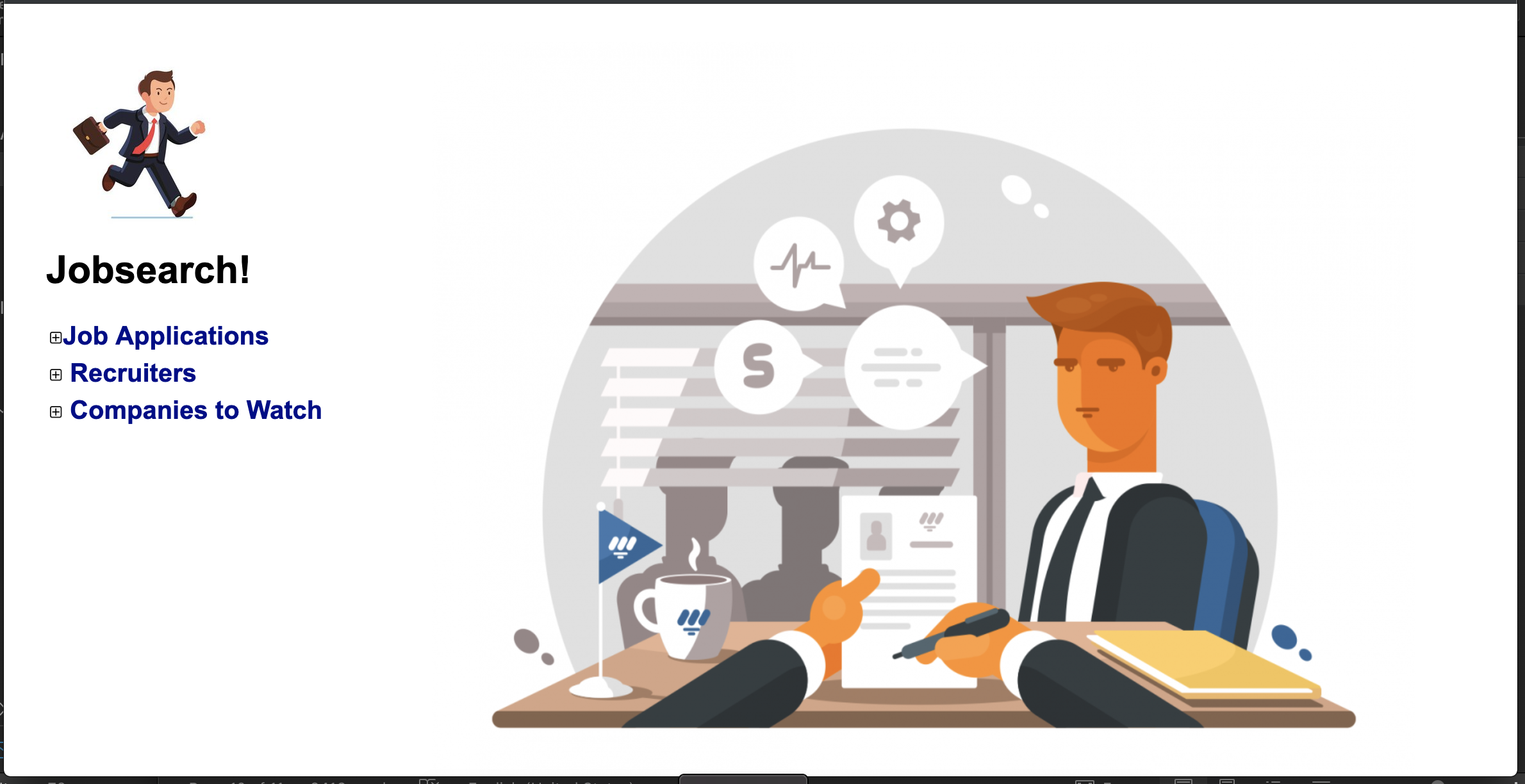


Note: For the critics in the audience, Its not lost on me that I didn’t complete Sprint 1 before I started Sprint 2, but the red stories have blocker issues that prevented me from moving forward with them, so I will have to move these to a “day 2” project.

## Developer

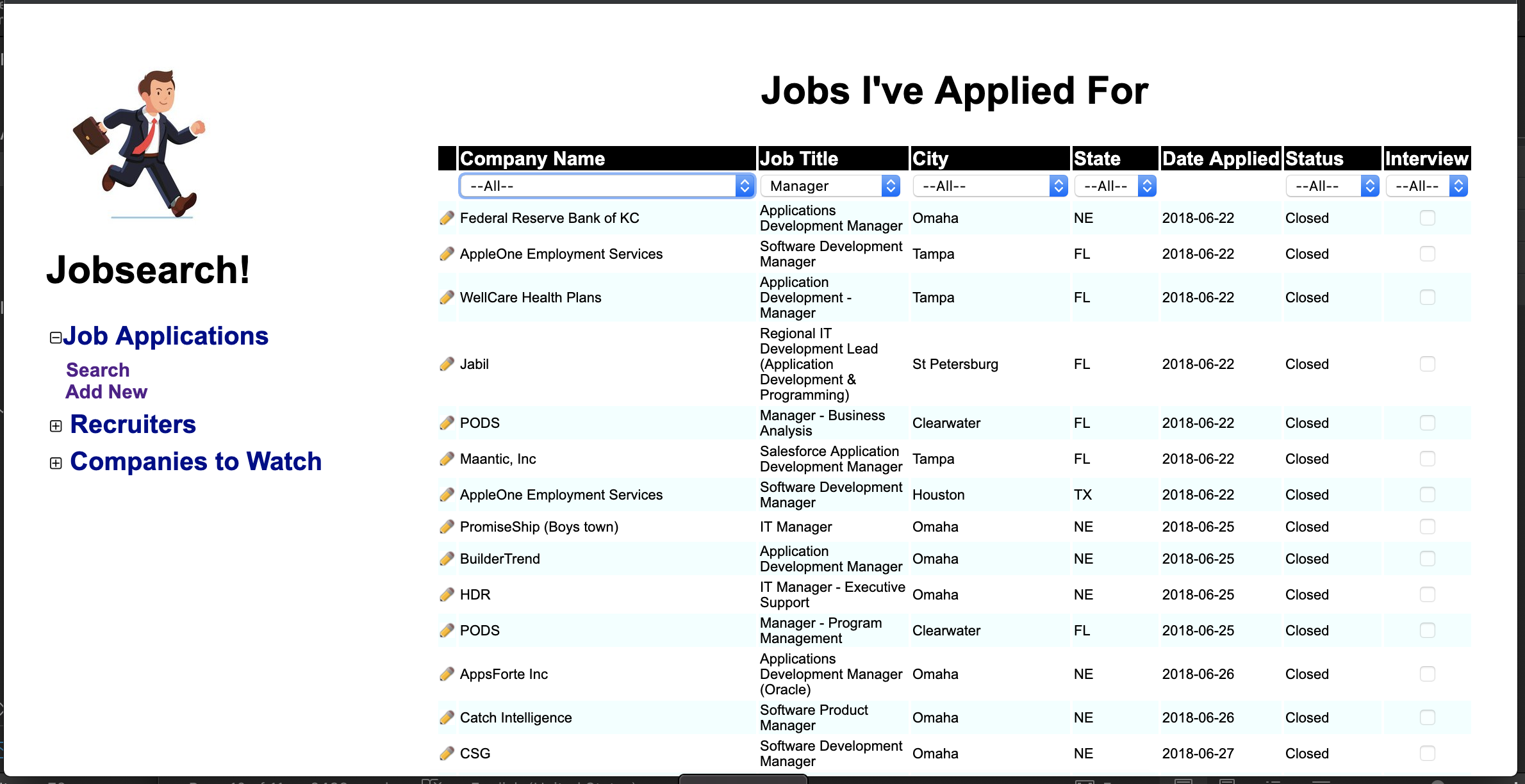
I already covered the technologies in the Tools section above so I won’t repeat it here. I will however give you a bunch of screenshots of what I developed and which stories each screen covers, here I go:

### Title Page



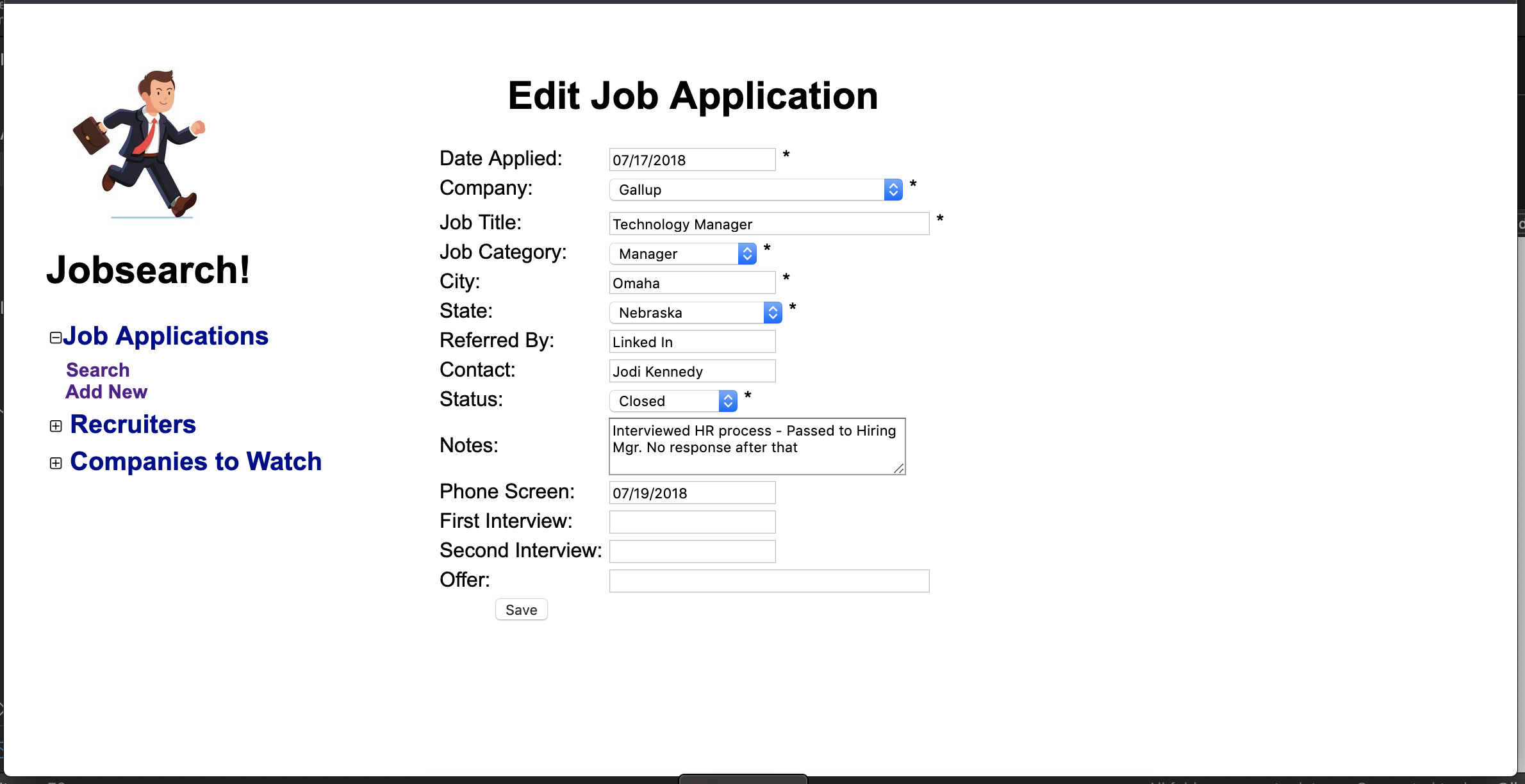
1. Note the menu on the left, allows expandable, easy to navigate functionality so I can get around quickly.

### Job Application Search



1. Displays a list of Jobs I’ve applied for since day 1 of my job search (data converted from spreadsheets)
2. Filters under the column headers allow me to filter results based on criteria and refreshes when a new filter option is selected. This filter shows all the manager positions I applied for.
3. Clicking on the pencil icon takes me to the detail screen where I can see more info or edit the information in case the status changes or I get an interview, etc.
4. Users are not allowed to delete, because these are historical data and keeps a record.
5. Clicking on the column header should sort based on that column and current filter but I haven’t quite gotten to that yet (its on my issues/enhancements list)
6. This screen addresses the following stories:
   1. #9 – List Jobs Applied for –> Acceptance criteria: I can see a list of all the positions I applied for based on the search criteria input
   2. #3 – Conversion of existing job search data

### Add/Edit Job Application



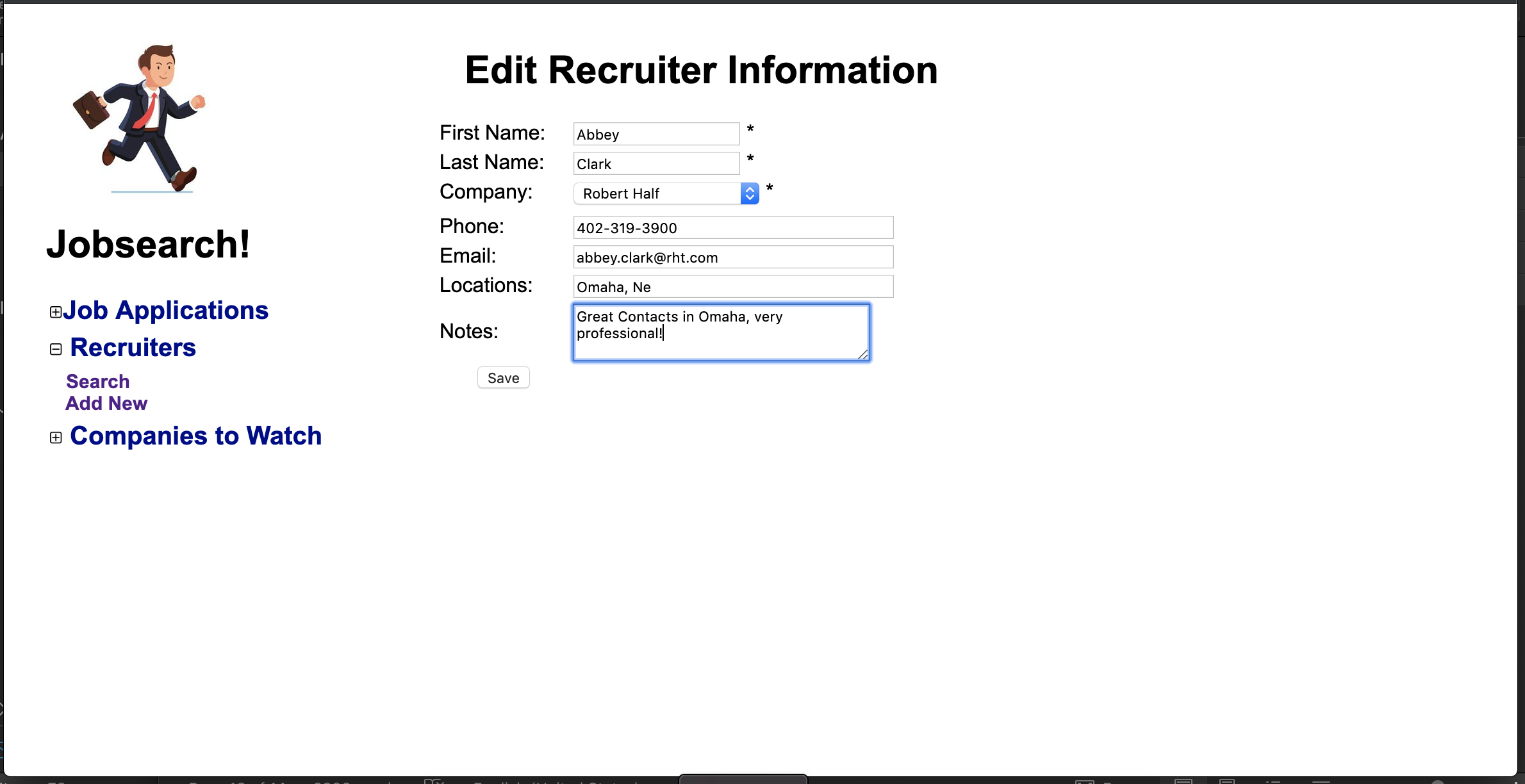
1. This screen allows me either add or edit a job application experience. Clicking on the Add New menu item gives me this screen with all field blanked out. If user clicks the pencil icon from the search screen, that job app data fills in as shown here.
2. Some data is available via dropdown for quicker, more consistent data entry.
3. Used JQuery to validate required fields based on data type.
4. All add/edit functionality is contained in one page called edit\_job.php
5. This screen addresses the following stories
   1. #2 – Document Jobs Applied for – Acceptance Criteria:
      1. Navigate to a form
      2. fill out the form with Job information
      3. Validate that data is correct format
      4. Submit the form and empty out the fields so a new job can be entered.

### Recruiter Search



1. Yes, I know this looks familiar, I prefer to think of it as a consistent format ☺
2. This displays a list of recruiters I know and some info about them including the locations where they have relationships.
3. Filters are similar to the job app search page. The Location filters by state abbreviation so if a location contains that state abbreviation, it pulls it. Its not perfect but it works for my purposes. In a business application, I would normalize the state data in the database with an associative table but for me, its not worth the time it would take. (gotta hate that legacy data) ☺
4. Users are allowed to delete contacts from this page by clicking the X icon and confiming they want to delete.
5. This addresses the following:
   1. #10 List Recruiters – Acceptance Criteria :
      1. I can see the list of recruiter information based on the input search criteria
      2. I can click on a link to email the recruiter (I know, I know, its on my issues list)
   2. #3 – Conversion of existing job search data

### Add/Edit Recruiter



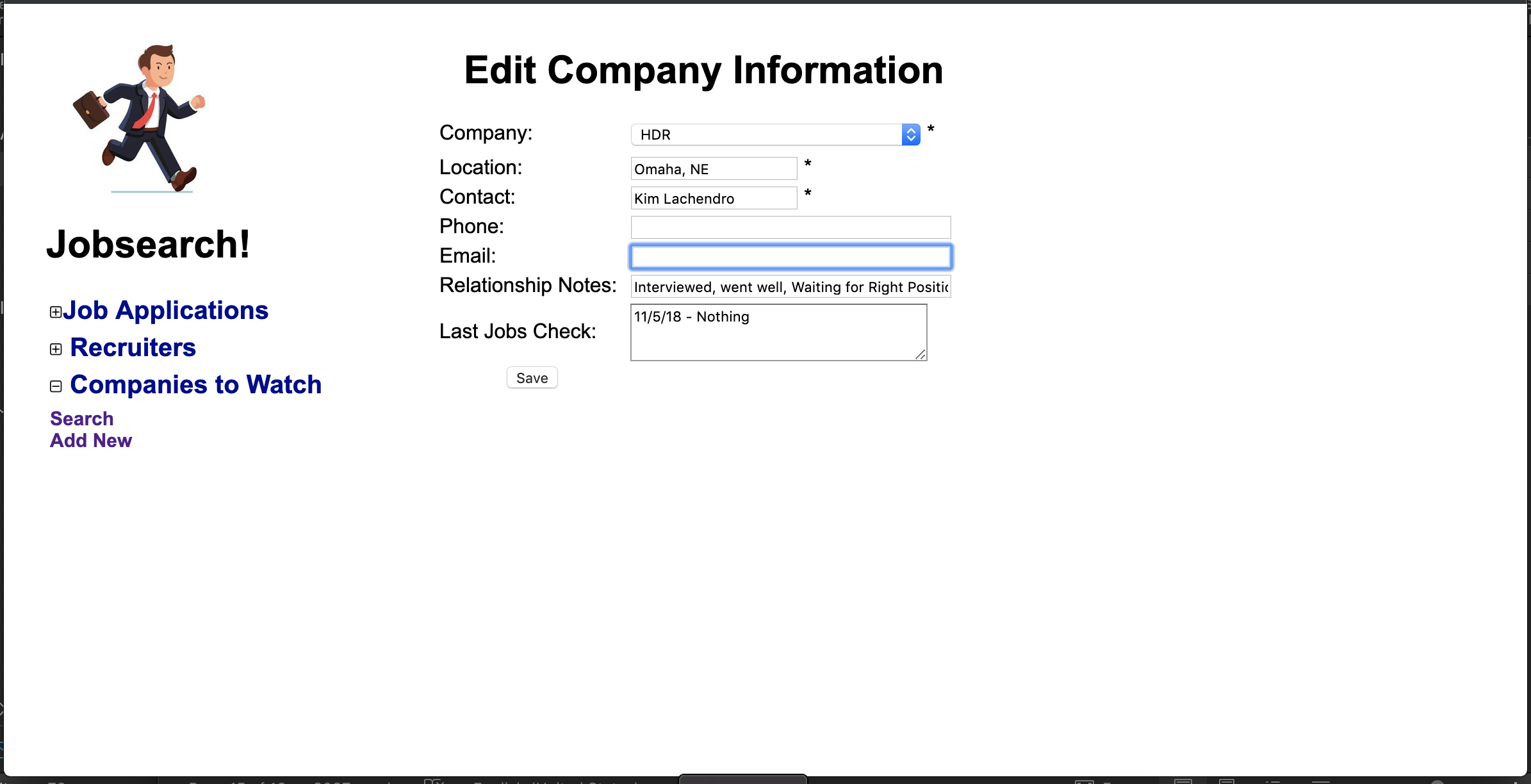
1. Allows me to Add or Edit recruiter in the same manner as the job app edit page
2. This screen addresses the following stories
   1. #4 – Networking Contact Documentation – Acceptance criteria
      1. Navigate to a form
      2. fill out the form with contact information
      3. Validate that data is in correct format
      4. Submit the form and empty out the fields so a new contact can be entered.

### Company Search



1. Same list format and functionality as other search screens for Companies I’m Watching.
2. This screen addresses the following Stories
   1. #11 – List Companies – Acceptance Criteria:
      1. I can see the list of company information based on the input search criteria
      2. I can click a link to navigate to their company career site (issues listed)
      3. I can click a link to email a contact there (issues listed)
   2. #3 – Conversion of existing job search data

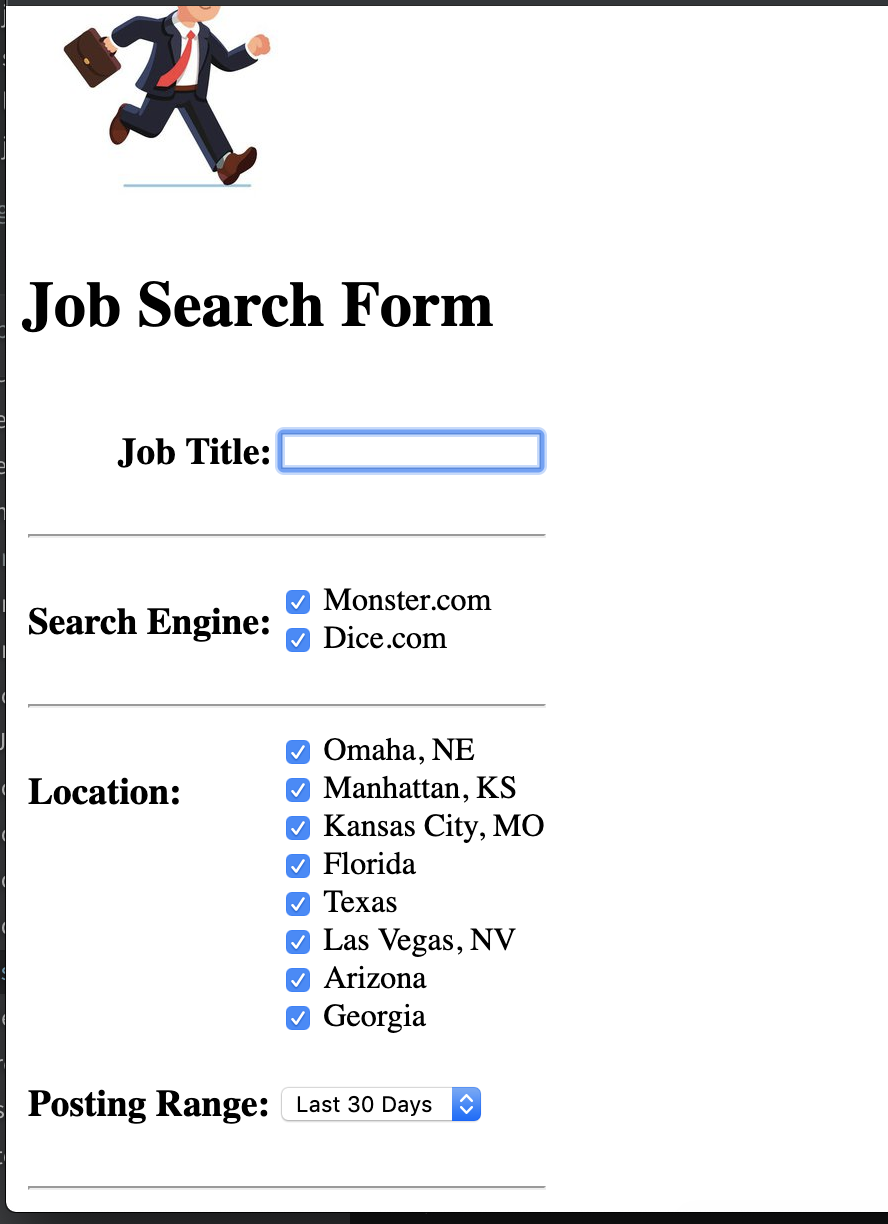
### Add/Edit Companies



1. Same Add/Edit functionality as previous areas.
2. This screen addresses the following stories:
   1. #5 Document Companies to watch for jobs – Acceptance Criteria:
      1. Navigate to a form
      2. fill out the form with company information
      3. Validate that data is in correct format
      4. Submit the form and empty out the fields so a new company can be entered.

Blocker Issues:

1. One of the issues popped up when I wanted to consolidate my job search by filling out the form below and submitting. My plan was to call the APIs on Monster, Dice, Indeed, etc. one by one to get a list of job postings back in JSON format, I could then store those JSON files, open them up and keep appending them to display all search results in one list. However, some of those sites no longer offer APIs and others require a certain amount of web traffic on your site in order to gain access to the postings, so with a traffic level of 1 user, I didn’t qualify. That may be an insurmountable problem for this system. Here is what I started before I hit the wall:



1. Another blocker, which was Time, prevented me from exporting the search results (filtering, sorting, etc) from each search page and exporting them to an excel spreadsheet to download. I found some code that would allow me to do this, but I haven’t had the time and I wanted to get on with the QA and DevOps setup so I punted for now and I’ll come back to it later.

Issues and Enhancements:

I’m sure there are plenty of holes to be poked in my system so far, but as I mentioned throughout the document, I have a pretty healthy list of enhancements and issues stored in Taiga.io, and they’re all assigned to me. I’ll have to decide the value of each and if its worth my time to fix or add.

## QA

I haven’t put on my QA hat yet other than planning. My plan is to start with a full regression test, writing test cases in Gherkin Format, running them through Cucumber to get a structure built and complete the tests using Selenium Webdriver in Java 8. True Agile would have dictated that I do this during each sprint but I was having such fun programming, I didn’t want to stop so I made the executive decision to say Screw it, I’m going move QA to post sprint work. Same for Implementation. Ongoing enhancements will try to use a more true agile process with functional test suites customized to each enhancement during the sprint.

# What’s Next?

I mentioned earlier that the idea is to get a full SDLC pipeline going before I move on, so now that development is mostly done, I’ll work on some test cases, get Jenkins configured and get the implementation process fully automated, then I’ll come back and start working my way through the outstanding issues and stories, and see how my little workflow works.

After that, I want to get some Java experience so I’m still thinking about how I can build onto this using Java, web services, etc. At some point, hopefully I’ll get a job as a programmer again and I can get my programming fix during work, then I’ll have to use this as an R&D platform to expand my knowledge on whatever I’m working on and surrounding environments. It will never stop.

# Who Is This Larry Ruh Person anyway?

# Where can I get this awesome system?

# Disclaimer

Not an artist

Frames