Lifespark Technical Evaluation

For Data Engineering Candidates

Introduction

Congrats on making it to this step in the interview process! We are looking forward to seeing what you can do.

This evaluation is designed to see how you'd handle situations that you might encounter on the job at Lifespark. Since this is a remote team, a lot of the actual work will be taking requests like the ones below and seeing them through to completion. Since real work doesn't have too many true/false or multiple-choice questions, we've avoided them on this test too. Instead, you'll find that you could solve these things in a variety of ways, and we hope to see your creativity shine through in the decisions you make.

Good Luck! We look forward to seeing what you create and learning more about how you work.

Tips

1. While we'd normally love for you to clarify requirements, in the spirit of fairness we hope that you try and complete the evaluation with the information we've given you. But if you get completely stuck, let us know.
2. This test is designed to take 2-3 hours to complete. Keep this in mind as you think about the solutions – try not to get too carried away.
3. When you're creating test data, assume you'll demo this to someone – have fun with it, but keep it safe for work. And please don't just put "test" in every field.

Pre-Requisites

Since Lifespark is a Health Care company, we have created some fake data to support this exercise. Don't worry this data is completely made up, they aren't real people it is not real PHI. You can find these data sets attached in this email

Please submit your code via Github. It can be a public repo or you can be prepared share it with one of our teammates as part of submission.

You can complete this exercise in any language of your choosing. Currently we have expertise on the team in Ruby, Python, and Javascript. It will be fun if it is in a language that we are unfamiliar with, just make sure we have the setup steps.

Scenario

We regularly get data from our partners for each population we are serving. We are currently processing these data dumps (these files come as excel, csv, pipe delimited, etc) into our data warehouse to support various forms of analysis. We have many questions that we ask on a daily, monthly, or quarterly basis and our data pipeline is being built to support these analytical questions.

We would like you to build a mini version of a data pipeline and be able to produce some reports that will help us make better decisions around care of our members.

These reports are often at the request of our CFO, our VP of Population Head or our VP of Life Management.

While your application is running, a user should be able to load new files into the system. This could be as simple as dropping a file into a directory or running a command or a file upload. Just make sure it is well documented.

User Stories

Here's what we've gathered from conversations with our CFO and VP of Pop Health. We've taken a stab at ranking them in order of importance, so if you don't have time to get to all of them, just start at the top.

1. As the VP of Population Health, I'd like to be able to see a list of 10 members that have had the highest cost of care from admissions.
2. As the CFO I would like to be able to see the total count of and cost of care for admissions broken down by week Sunday to Saturday.
3. As the VP of Life Management I would like to see a report of members that have had a readmittance within the last 30 days.
4. As the VP of Life Management I would like to see a report of the top 10 longest admissions.
5. As a data engineer I would like to be able to upload a new list of admissions.

Submission

Your proof of concept will be evaluated by Tim Park, Engineering Manager at Lifespark.

To submit your proof of concept, please do the following:

1. Don't forget a Readme!
2. Record a quick screenshare of your work highlighting the user stories. Upload this video to the git repo as well
3. Share your git repo with Tim via email, tpark@lifespark.com - Or if this is on Github you can share it to https://www.github.com/timsjpark