Take-Home Screening Exercise

Overview:
You will write a small, fully functional web application that models the business of a fictional competitor to TicketMaster (let's call it AdmitOne). We will judge the submission primarily on code quality and system design. If you are asked in for an interview, we will refer to your submission and ask you to speak to choices that you made in design and implementation.

Scenario:
A REST service on AdmitOne's servers receives three types of messages: Purchases, Cancellations, and Exchanges.

A Purchase consists of:
* The username of the ordering customer (we assume they already have an account)
* The number of tickets wanted
* The id of the show for which the tickets are being purchased (just an integer is fine here)

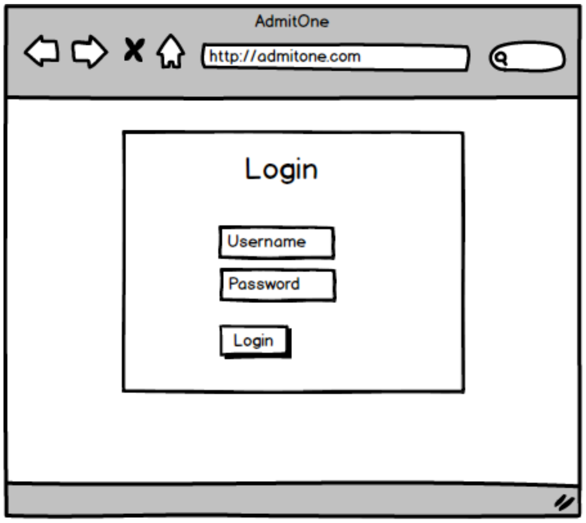
A Cancellation consists of:
* The username of the ordering customer
* The number of tickets canceled
* The id of the show for which the tickets are being canceled

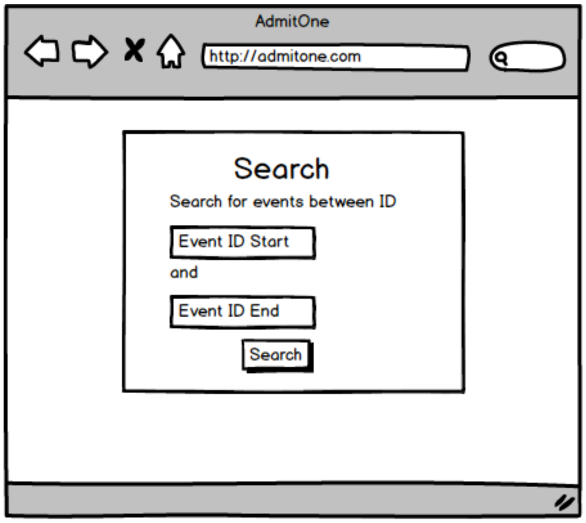
An Exchange consists of:
* The username of the ordering customer
* The number of tickets being exchanged
* The id of the show being exchanged from
* The id of the show being exchanged to

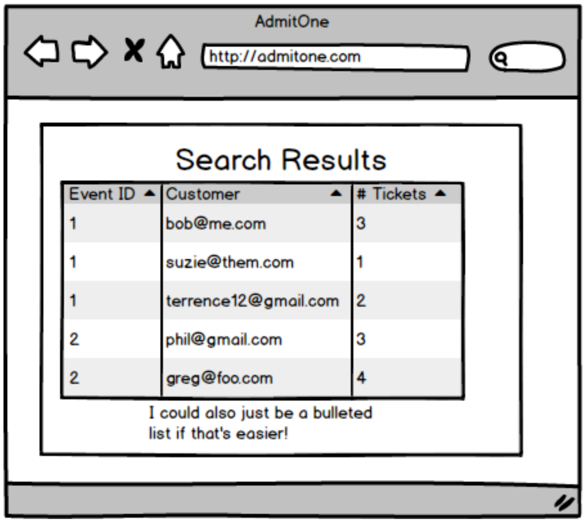
Let's assume, for the sake of simplicity, that every show has an infinite capacity. You cannot, however, exchange or cancel more tickets than you have ordered.

The web front-end for AdmitOne is only for administrators: all customer interaction occurs via the REST endpoints. The web app has three screens:

1) Login (a single hardcoded username/password is fine, but the site should be protected)
2) Search form (search for orders by event ID range)
3) Results page (lists each show, with the customers attending and the current number of uncanceled tickets)







Search Results

Event ID ▲	Customer ▲	# Tickets ▲
1	bob@me.com	3
1	suzie@them.com	1
1	terrence12@gmail.com	2
2	phil@gmail.com	3
2	greg@foo.com	4

I could also just be a bulleted list if that's easier!

There is no need to display order history for a given customer, but you should have a means of retrieving it on demand.

Process and Submission

The submissions should have very simple instructions – and ideally a single script – that will allow us to start the web server, database, and any supporting servers. The submission can either be uploaded to GitHub or provided to PatientPing as a compressed archive. Please feel free to route any questions about the requirements or process through your recruiter. Don't forget your test!

Thanks and good luck!