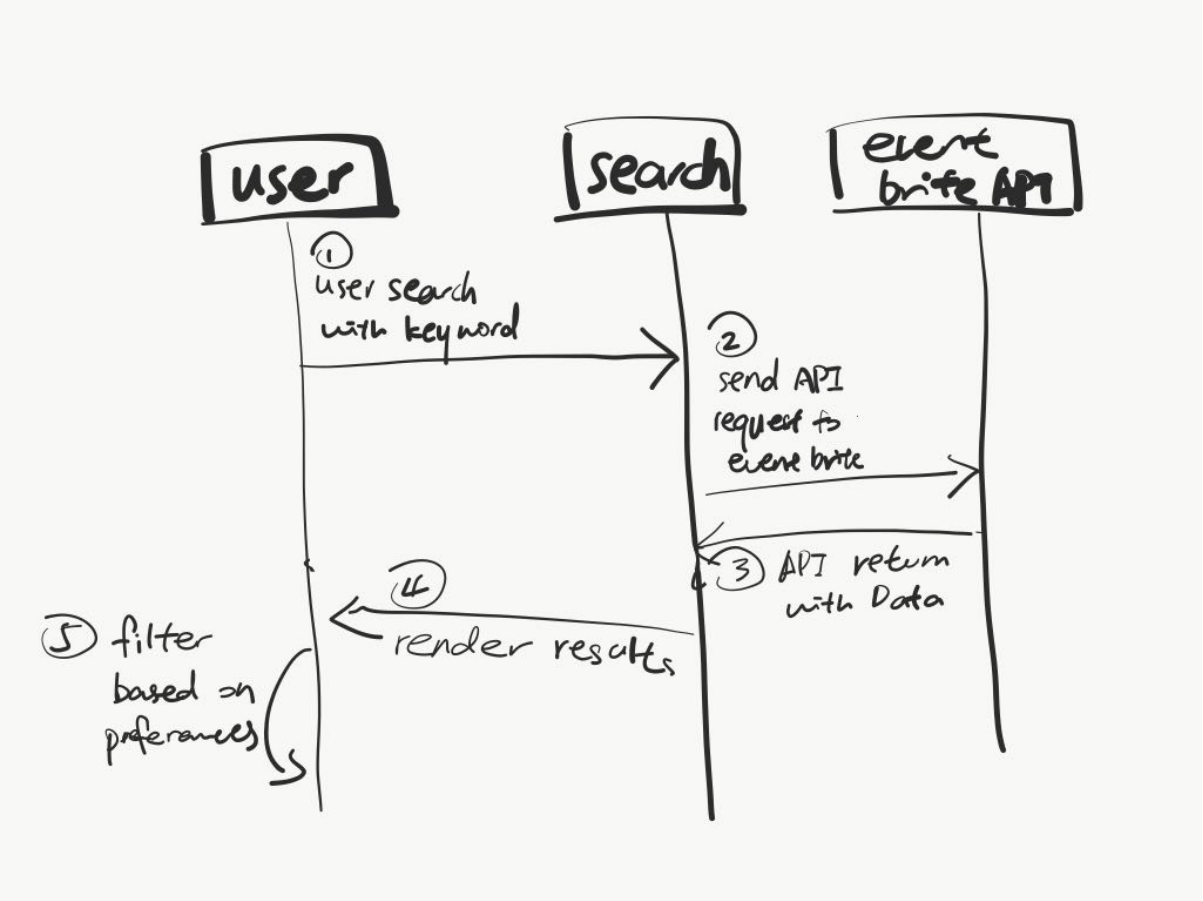
FunTime (Group 5 Section A4)
Team Assignment 3

UML Diagram showing searching case:



Analysis and Architecture Decision

User Story: User searches for lists of events around themselves, and filters them by their preference

Happy Path: User searches events using keywords. App server sends get request to Eventbrite, and Ticketmaster, using each API credentials. App server receives data from each API, and renders search results on client side. Users can sort/filter by their preferences that are saved on client side when they are logged in. If user is interested in a certain event, they can click “Add to schedule” to save to calendar.

What could go wrong?:

- User enters something in the search field that does not generate any results
    - In this case, we will render a simple view that states nothing matches the given input
- There are errors fetching the data for the given user input
    - Prompt a simple error page specifying that the user should try again shortly

4. Our project stack: ReactJs

We chose to use ReactJs because it is a very popular framework that is easier to learn than others. ReactJs is constantly growing and is usually the framework of choice among web based apps. React allows reusable UI components to make apps easier and scalable as well as multiple 3rd person libraries that are easy to implement. There is also a large support group for React and very well written documentation. There is also experience among our group members.

Angular, on the other hand, has a steeper learning curve and requires learning TypeScript, a language that is not too popular as of now. We chose Node over Python or Java because it allows us to save a lot of time by having to learn only one language: Javascript. We did not choose SQL because query can get complex very quickly and for our purposes, Mongo works better and saves time for MVP.