[Figma file](https://www.figma.com/file/CkvwWMzxcI4lI4udHm8QiH/Labs18_Conference-Contacts%2C-Emily%2CTylerr?node-id=1127%3A90103), which shows previous iterations of events-related design elements

Various Trello cards related to this post:

[1](https://trello.com/c/cmI7UdMV/54-user-flow-once-arriving-at-event)

[2](https://trello.com/c/pKgMvaGg/53-click-on-event-tags-to-display-list-of-users-connected-with-that-tag)

[3](https://trello.com/c/e6jUW1CY/72-events)

[4](https://trello.com/c/egNzjvRA/57-events-displayed-in-home-profile)

* Describe how the requirements of the features you have been building/designing have changed.
* We changed from a conference contacts app to marketing as more of a general networking app because we tried to work out implementing events into the app and, after coming at it from every possible solution we could think of, we figured out the details made it take too long to be able to implement.
* There were issues with location in reference to an event and how to pull that information together, where to pull it from, how to handle it once it was in state. For privacy reasons, the two major event aggregators, Facebook and Eventbrite, don’t allow you to do certain things on their API, so that pretty much settled it because the alternative would be setting up our own events database of some sort from scratch and that totally gets away from what the app is trying to do, because it means customers would have to input their own events.
* In general, there were many aspects of this issue that caused us to realize, despite the stakeholder justifiably really wanting this in the app, that we could not implement it in the time constraints.
* In the end, we removed the static events calendar entirely and scrapped the entire potential feature. It’s still in the Figma design files and in a release canvas if the next group wants to work on that as a new feature, if they figure out a way to get past the privacy issues and other problems that popped up. The static version which used to be shown on the live website is now also commented-out code, so they can always go back and use it.
* What caused these requirements to change?
* Working out the details of how and figuring out we didn’t have enough time to implement what was required to implement the features
* What types of architectural requirements changed? Did you have to make any trade-offs or add any new packages/libraries or dependencies to your application's technical stack in order to meet the new changes? **Please be as specific as you can. Provide details via screenshots to Trello cards or your release canvases to support your response.**
* The tech stack didn’t really change; what changed was our perception and marketing of the app after a robust team discussion within the team and among higher-ups and stakeholders.