[Release Canvas I:](https://www.notion.so/Release-L22-1-Update-Functionality-d1fda2ccaec04612916f77642ea1f155)

**Product Vision: Week 2**

### Prompt 1: Decision Defense

* **For data scientists and engineers:** Describe the features in your first release canvas. How do the features in your first release canvas support one another? How will the completion of the first release canvas help drive the technological decisions moving forward?

We are doing what we would call updating the functionality. We figured a good way to jump in and get to know the code was to do major user-facing bugs first, or the ones we thought were noticeable by users. Again, our users aren’t the most tech-savvy user base you could have, so the app being user friendly is very important. We also had a discussion about how fixing bugs would be what happens at the majority of jobs; probably won’t get to do flashy new things as a newbie to a company. We imagine more senior people get to do cool stuff they want to do, so this also influenced our decision to fix bugs first.

So our first release canvas consists of fixing some of the more major user-facing bugs that we narrowed down from a list we made. In order to decide what priority we should give them, we agreed as a team that user-facing bugs/fixes were the most important and each initialed the ones we thought could be done in one to three weeks.

Next, we then further narrowed the list down to our first release canvas by putting the list of bugs in order of quantity of people’s initials. Then we decided as a group to aim for stuff we could turn around in a week to a week and a half, and polished the canvas after coming up with the final list of bugs to fix for that particular canvas.

We did this for several canvasses and decided as a team that we would leave the ‘new’ feature for last so we’d have whatever time is left to work on it AFTER accomplishing the MVP goals set out for us. All of the things we fixed are more user-facing, and we gave that concept a fairly high priority. That drove most of our decision making frankly, what we thought would most affect the user, and then secondarily or along with that time, the amount of time we thought it would take.

We actually started the first canvas Thursday and have three draft PRs done already for three of these features.

### Prompt 2: Breakdown Entry

* **For non-UX students:** Summarize what you included in your first release canvas and why the features in that release canvas create a complete feature set.

Our first release canvas includes several bug fixes. When a user deletes a contact right now, it doesn’t confirm the delete or prompt ‘are you sure’, which we felt was very user unfriendly.

There were also several pieces of information which do not display consistently across iOS and web apps; birthdate, job title, tagline, bio section, and location; again, pretty user unfriendly. The profile picture alignment for both the home/profile pages was not consistent to the iOS app, so we plan on fixing that also; again, user-facing issue, annoying that users might go fully fill out their profile but later find out it’s totally pointless to do so because some users can’t see any of the information.

Also, as a team we felt the link update/creation area and process may be confusing for the user, so we are going to redesign that a little bit. For example, the ‘world’ icon designating a regular dot-com site doesn’t show up in the list after a certain point and there's no prompt anywhere for how to save the link once you type it into the box; not a problem for savvy tech users, but our users largely won’t be tech savvy, so this is a problem for them and we are reworking that. Also, the token expiration was way too short, so we also decided to fix that as well; again, user-facing fixes being the theme here. A short token expiration means the user has to log in more frequently, annoying for a lot of our users.

Why did we select all these fixes/bugs? This will make the app a lot more polished looking and user friendly in general. There are other fixes and bugs to be made, but these were what we determined were the most user facing and could be done within the time constraints.

* How will you test those features to know that they actually solve the problem?
  + 90% of features are fully functional
    - number of interactions for happy path -define happy path (user flow) -record clicks + bugs in happy path
  + 25% or more increased user base

We’ll use Cypress where appropriate, and test the results against the above metrics to see if we met the above metrics.