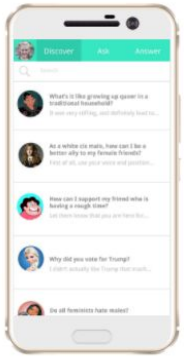
THOUGHTBUBBLE

FINAL REPORT

Bonnie N. | Grace H. | Jenny K. | Po T.

PROBLEM & SOLUTION OVERVIEW:



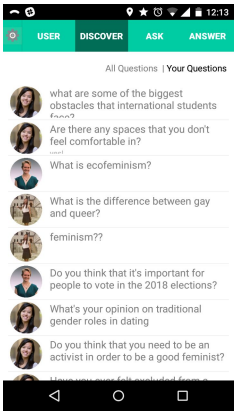
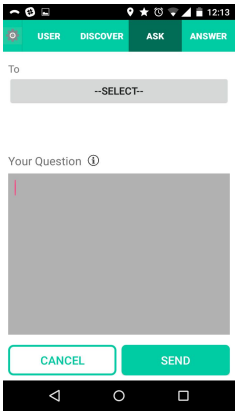
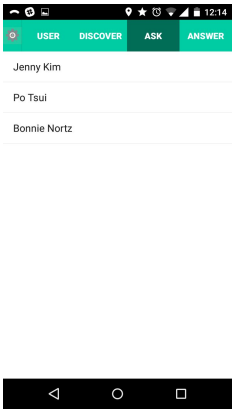
The story of Thought Bubble begins with the story of its users.

Our needfinding revealed the crucial insight that everyone has questions that they don't know how to ask. And the polarization, insularity, and lack of empathy we see in society is fundamentally an issue of communication. Thought Bubble is an Android application designed to facilitate these difficult conversations. We give users the ability to ask their friends anonymous questions, share their experiences, and build collections of questions and answers that contribute to learning about a new perspective.

TASKS & FINAL INTERFACE SCENARIOS:

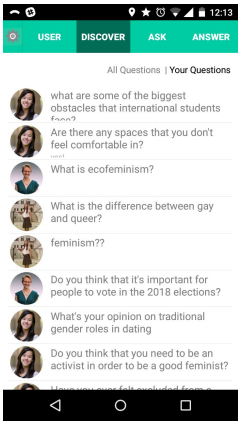
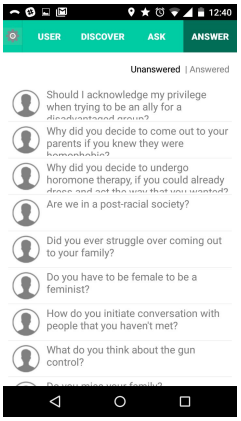
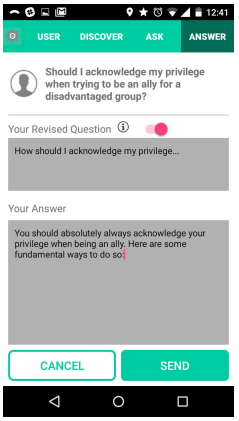
TASK 1: ASK A QUESTION

Since the primary goal of our app is to give people help in asking difficult questions, we chose to make asking the most frequent task. Asking someone a question in real life is already pretty hard, so our task flow is as simple and straightforward as possible:

		
Initial page (discover) that the app opens to.	Clicking on ask in the toolbar a page where you can write your question	Select which friend to send to and then hit send.

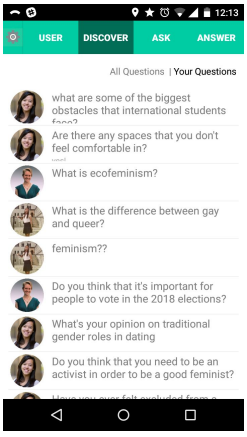
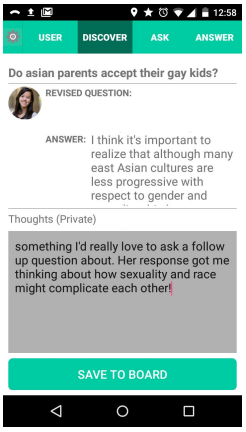
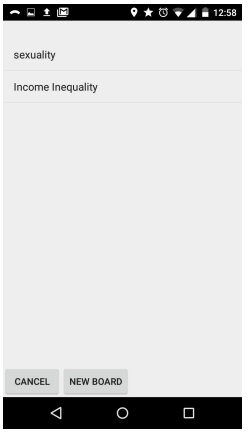
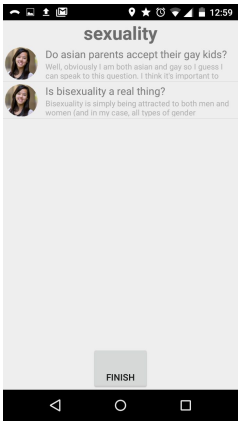
TASK 2: ANSWER A QUESTION

Because we wanted to help people ask tough questions, not only by streamlining the process but letting their peers teach them how to ask these kinds of questions, we also gave answerers the ability to critique a question. Thus, answering represents a slightly more-time intensive (and thus less frequent) task as it may involve critiquing a question in addition answering it.

		
Initial page (discover) that the app opens to.	Clicking answer takes you to your unanswered questions list	Clicking a question takes you to a page where you can provide critique and write about your experience

TASK 3: REFLECT UPON AN ANSWER

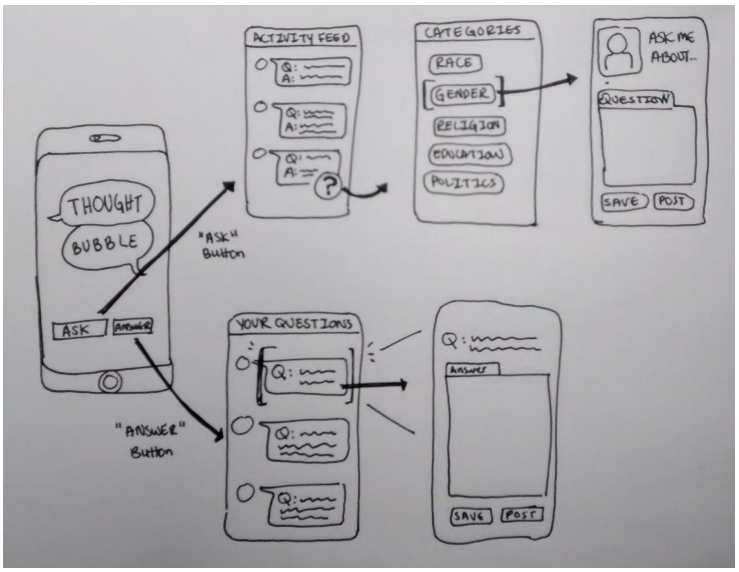
The long-term goal of asking questions is to build cross-community empathy and understanding. Thus, we gave users the ability to reflect upon different questions and answers by making collection that they can build over time. As in real life, reflecting and building upon new perspectives is often more difficult than asking or answering, so this task represented the most complex one of all.

			
Initial (discover) page the app opens to	Clicking on a question/answer in your discover feed allows you to write a private thought.	Hitting save to board allows the user to select which collection they would like to add it to.	After adding it to a board, the user will be able to come back and continue thinking about this response in context of other responses they have collected.

DESIGN EVOLUTION

INITIAL SKETCHES

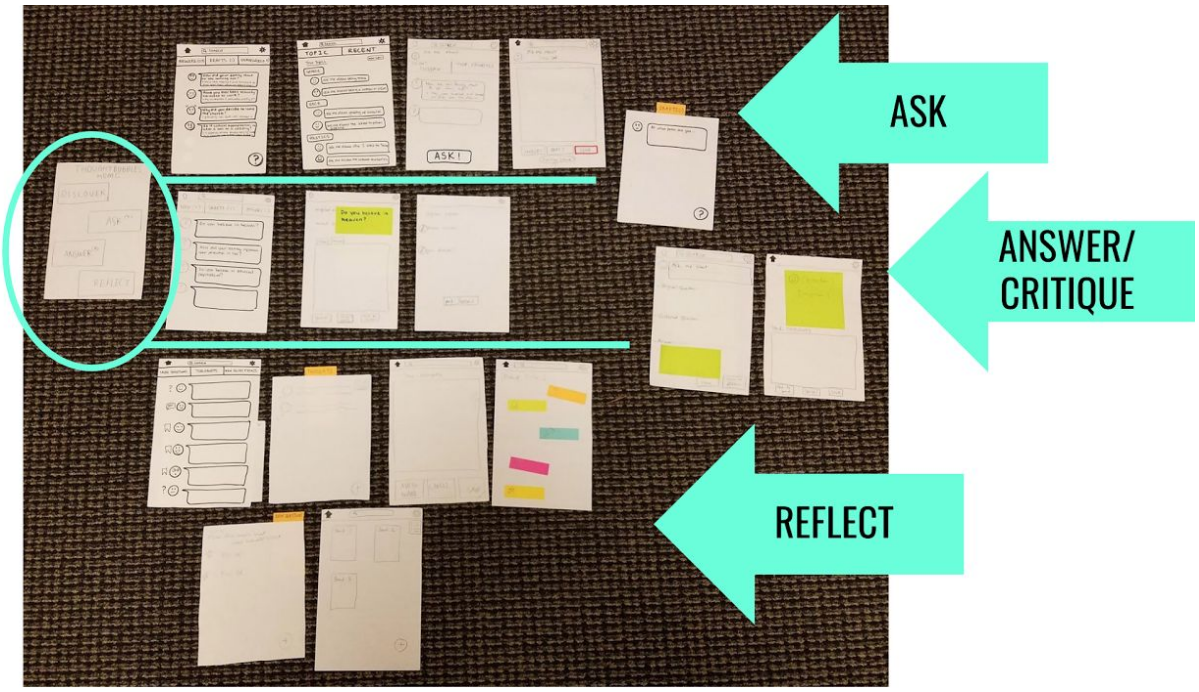
Our first step of the design process was to brainstorm multiple realizations across different devices of our idea. Of those sketches, we chose the following mobile design:



Through this medium, we wanted our users to incorporate our proposed form and content of communication amongst friends into their regular routine and normal devices.

LOW FI PROTOTYPE

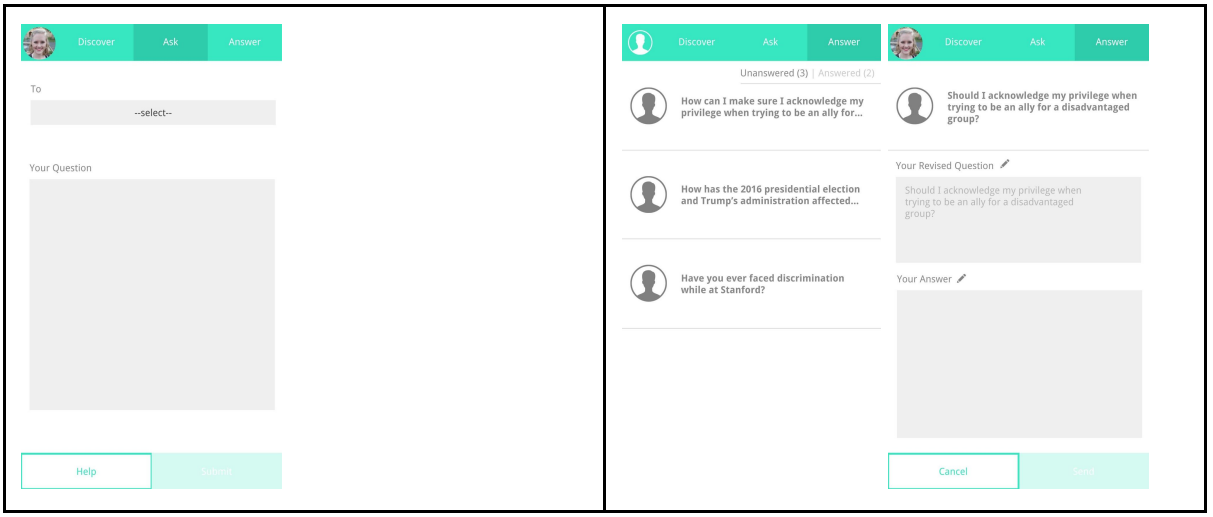
Next, we developed a low fi prototype to conduct user testing. We retained the home screen from our initial sketches and expanded each task to span multiple screens originating from the home screen. We also added a screen flow for reflection as we felt like reflecting on questions was a crucial part in addressing our user need. With the added reflection functionality, we added the concept of boards to collect those questions, answers, and reflections into a centralized location.

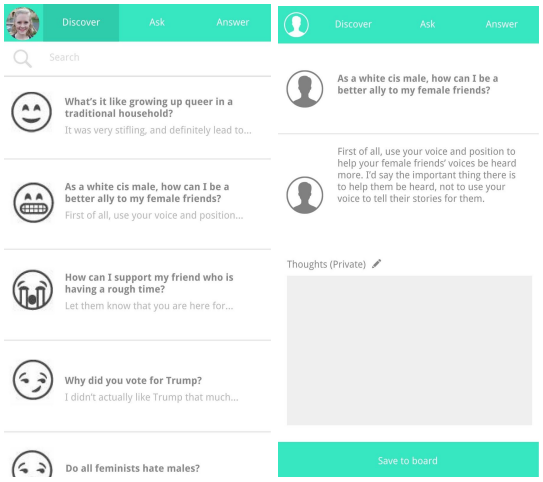
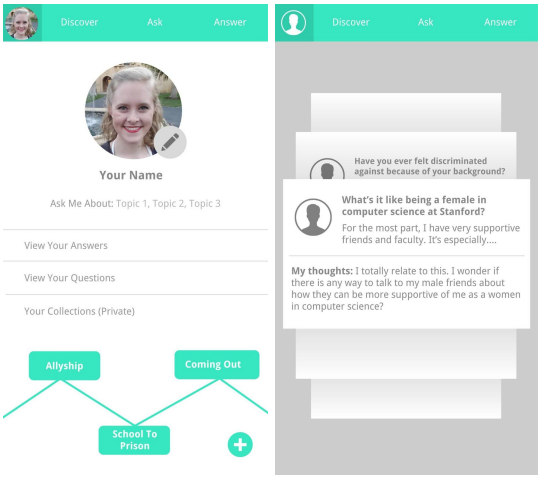


Our greatest insight from our user testing on this prototype was that users had trouble navigating to the right screen. They consistently hit the phone screen's back button multiple times, did not utilize our home button, and would on occasion get lost. The majority of our participants also expressed concern with anonymity and privacy.

MEDIUM FI PROTOTYPE

Moving forward with those realizations, we designed a navigational toolbar that denoted the three tasks in their respective tabs (as well as a user tab). We also reduced the number of screens required to complete a task, and removed content that was unrelated to the completion of the task. Another issue raised was privacy, which we ensured was apparent moving forward with a standard blank user profile image to denote anonymous users where appropriate.

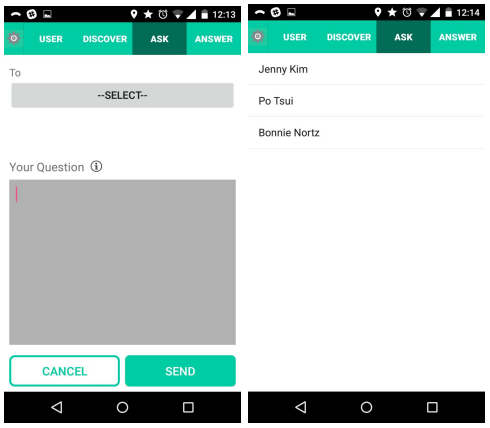
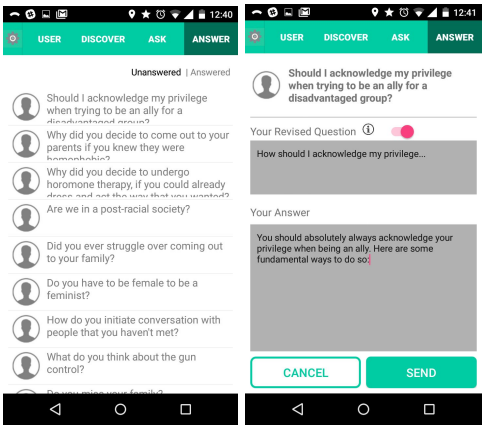


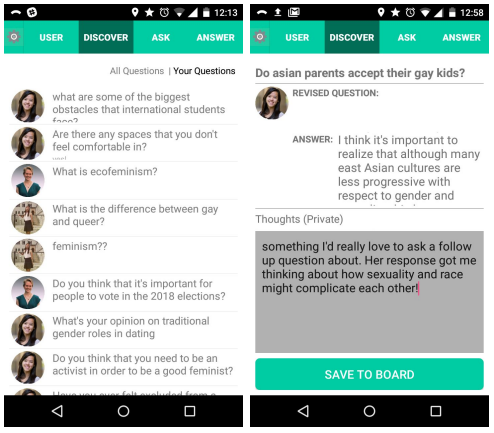
Task: Ask a question	Task: Answer a question
	
Task: Reflect on a question and answer	User profile and collections/boards

We received heuristic evaluations from our classmates on this medium fi prototype. Our evaluators discovered a total of 27 violations in our prototype, with the majority of them focused on consistency and user control.

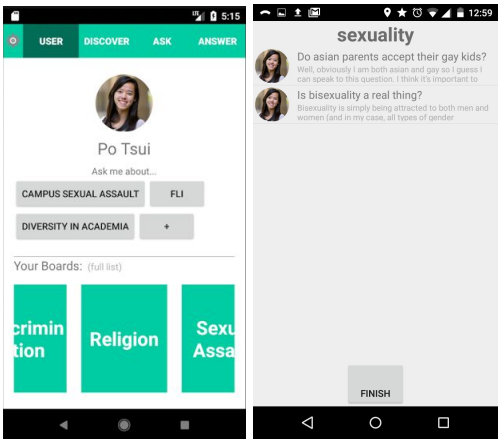
FINAL DESIGN

After incorporating the feedback we received in our heuristic evaluations (see below: Major Usability Problems Addressed), we implemented our final design in Android Studio.

	
Task: Ask a question	Task: Answer a question



Task: Reflect on a question and answer

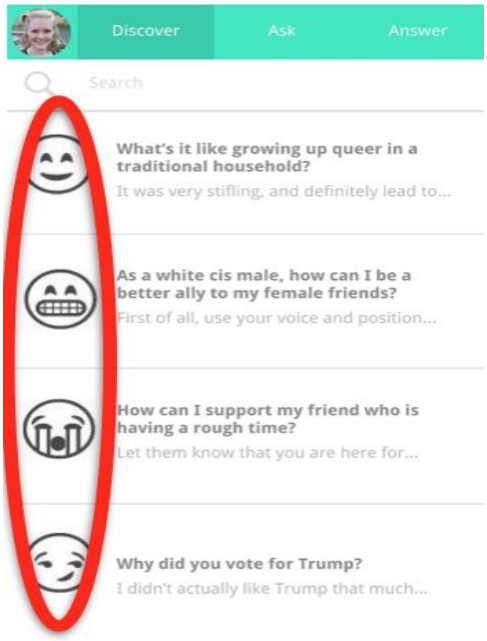
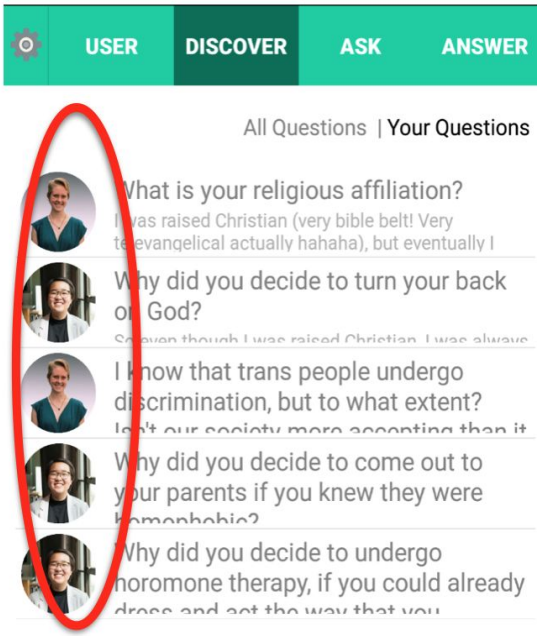


User profile and boards

MAJOR USABILITY PROBLEM ADDRESSED

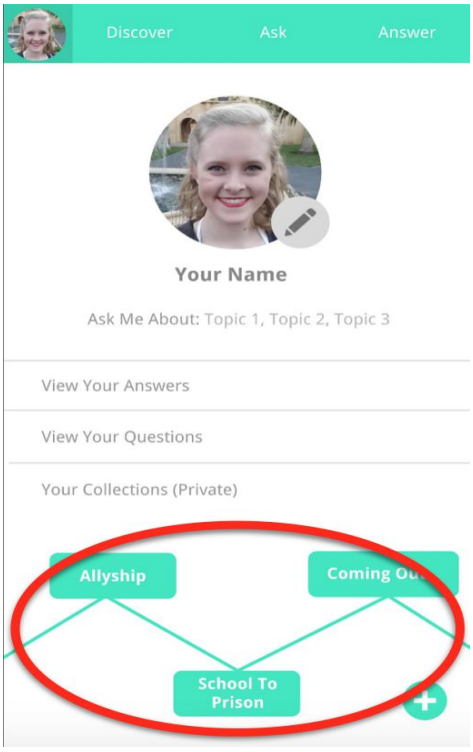
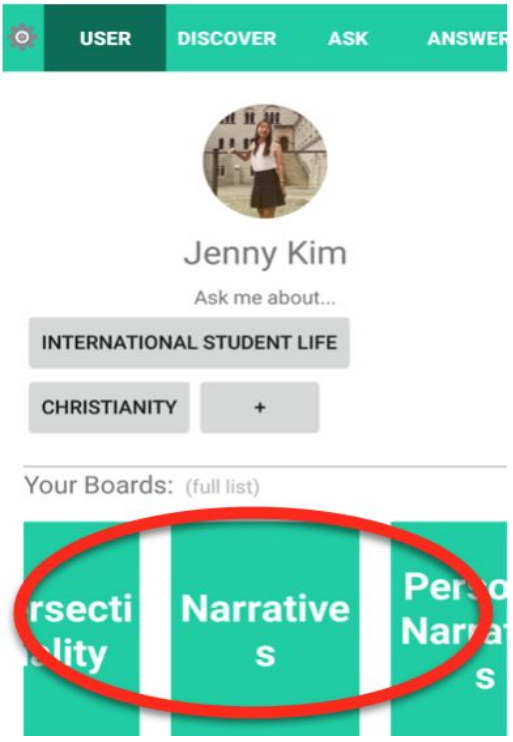
SEVERITY 3&4 VIOLATIONS

1. MISLEADING EMOJI ICONS (H10: HELP AND DOCUMENTATION, SEVERITY 3)

	
BEFORE The meaning of the emojis were unclear. The user would be confused what those emoji means.	AFTER We are using the profile pictures of the answerers instead.

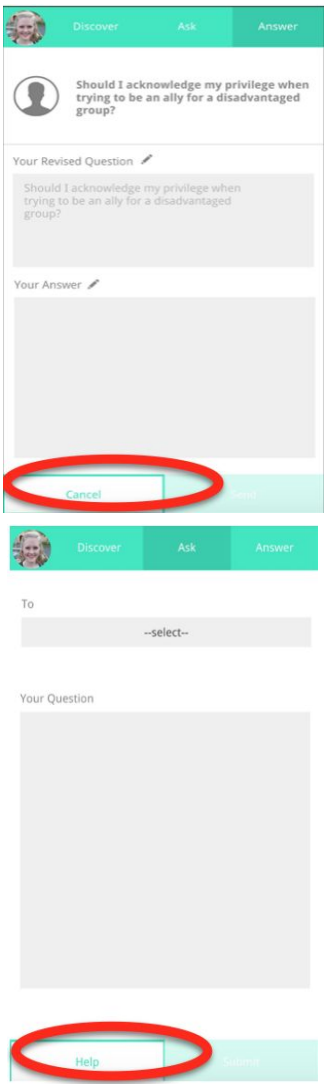
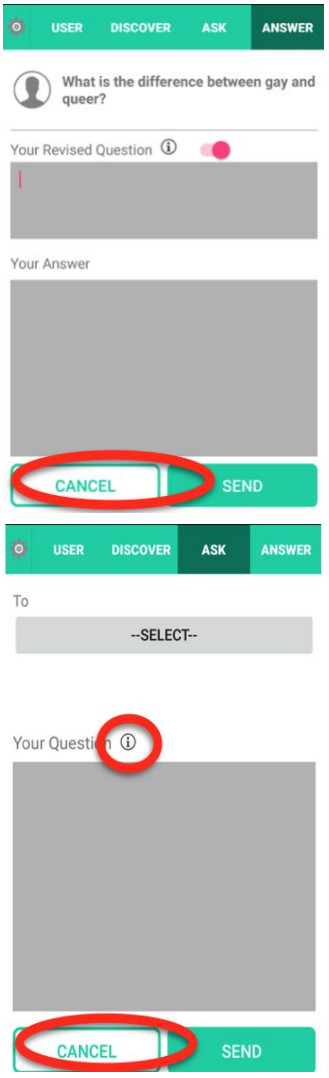
This was simply a mistake; we meant to use the profile pictures of the user in the mid-fi prototype too.

2. MISLEADING BOARD STRUCTURE (H4: CONSISTENCY AND STANDARDS, SEVERITY 4)

<table>
<tr>
<td> 
</td>
<td> 
</td>
</tr>
<tr>
<td>BEFORE
The board structure is misleading. Lines, valleys, and peaks may suggest hierarchies whereas there should not be any hierarchies between the boards.</td>
<td>AFTER
In order to remove hierarchies between the board, we implemented horizontally swipe-able board that helps navigate between different boards.</td>
</tr>
</table>

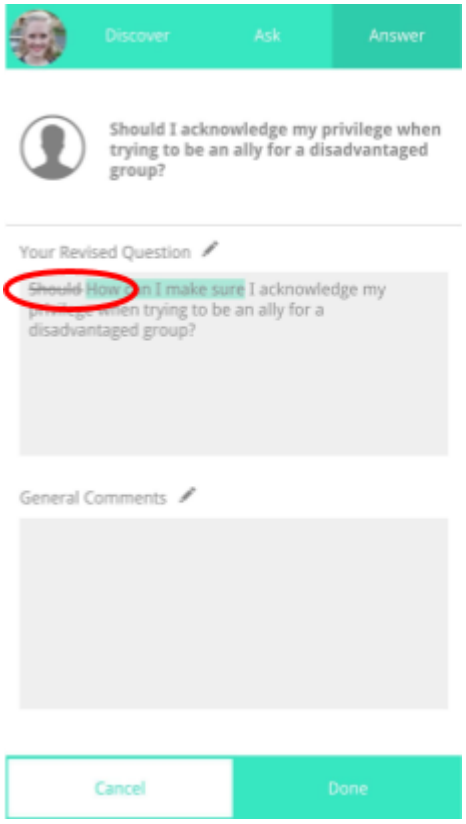
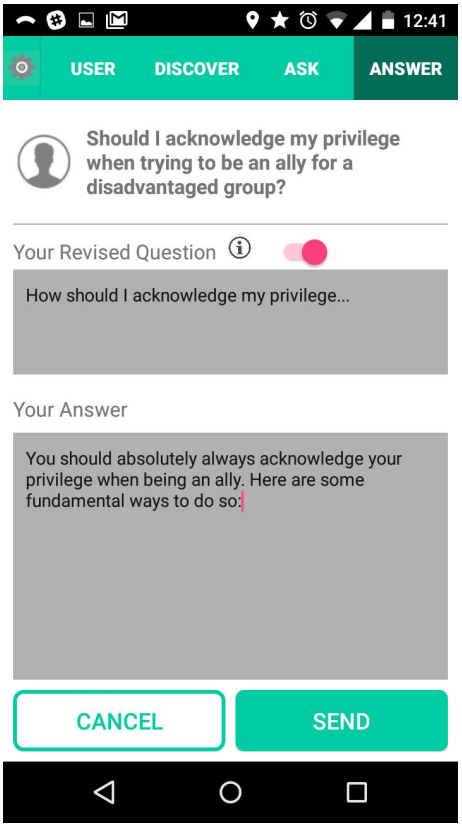
Rather than having a board structure that could be misleading, we implemented a horizontally swipe-able board. In addition to that, we removed the plus button which could also be misleading, and we allowed the user to see the full list of the board by clicking “full list.”

3. INCONSISTENT CANCEL BUTTON POSITION (H4: CONSISTENCY AND STANDARDS, SEVERITY 3)

<table>
<tr>
<td> 
</td>
<td> 
</td>
</tr>
<tr>
<td>BEFORE
In the “Answer” page, there is a cancel button in the left-bottom corner. However, in the “Ask” page, there is a help button in the same position. This can make users accidently hit the button that they are not supposed to.</td>
<td>AFTER
Both in the “Answer” and the “Ask” page, there is a cancel button in the same position. Help button got changed to an info icon -- once the users click the info icon, they can get help phrasing the question.</td>
</tr>
</table>

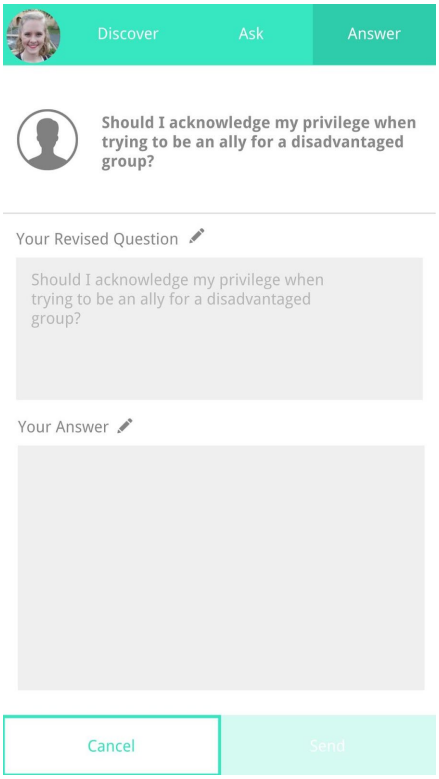
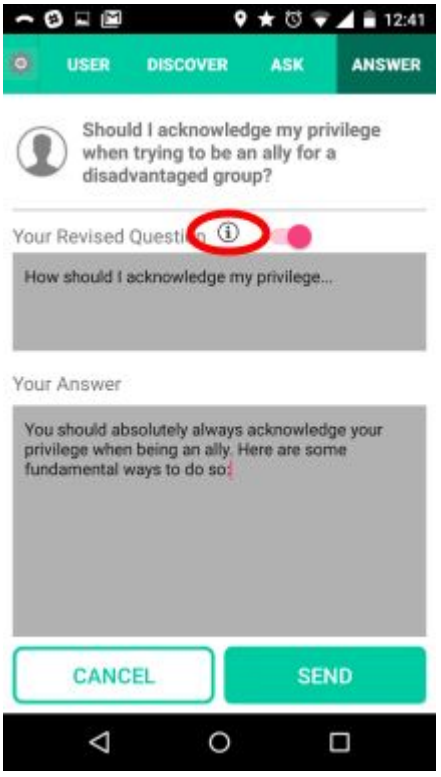
We acknowledged that having different buttons in the same position can be misleading. We replaced the help button in the “Ask” page with a cancel button. As we think it is important for the users to get help phrasing meaningful questions, we did not entirely get rid of the help button -- we implemented an info icon where the user can click and read about how to phrase a good question.

4. CONFUSION ON HOW TO ACCESS TEXT FORMAT OPTIONS (H3: USER CONTROL AND FREEDOM, SEVERITY 3)

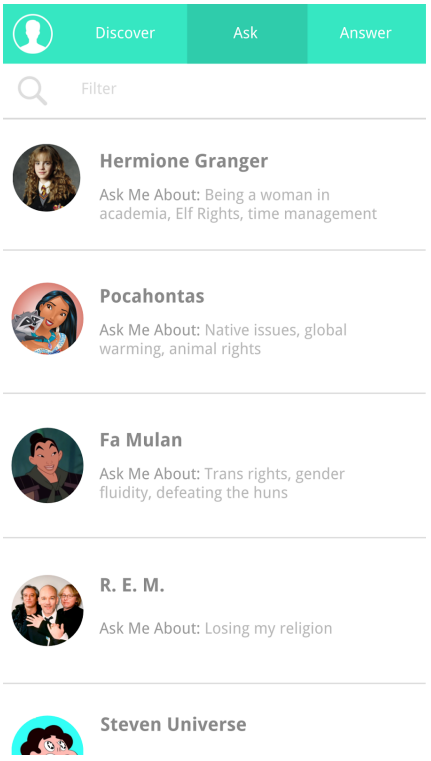
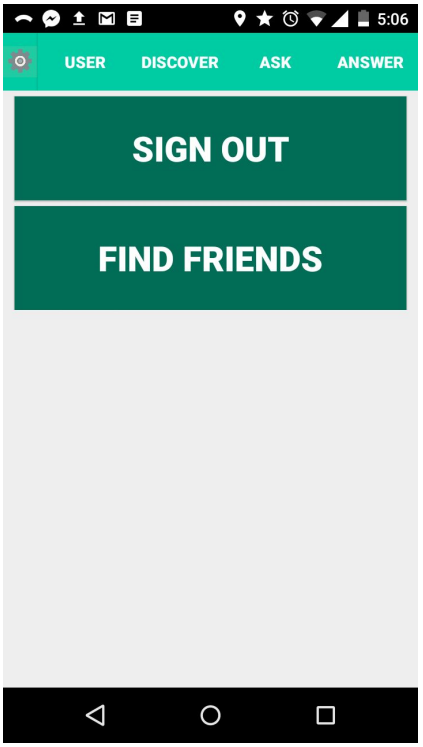
<table>
<tr>
<td> 
</td>
<td> 
</td>
</tr>
<tr>
<td>BEFORE
For the Answer task we had a complicated revision section that allowed for strikethroughs, highlighting, and commenting, but did not show the user controls that made these happen.</td>
<td>AFTER
We vastly simplified the Answer task so that only plain text was allowed in the revision section, removing the need for such complicated controls.</td>
</tr>
</table>

We chose to simplify the Answer task for a number of reasons, the first of which was simply to make the flow less convoluted for the user. We made the decision to simplify the revision portion in particular because we thought users would care a lot more about answering the question than revising it.

5. NO EXPLANATION OF QUESTION REVISION (H10: HELP AND DOCUMENTATION, SEVERITY 3)

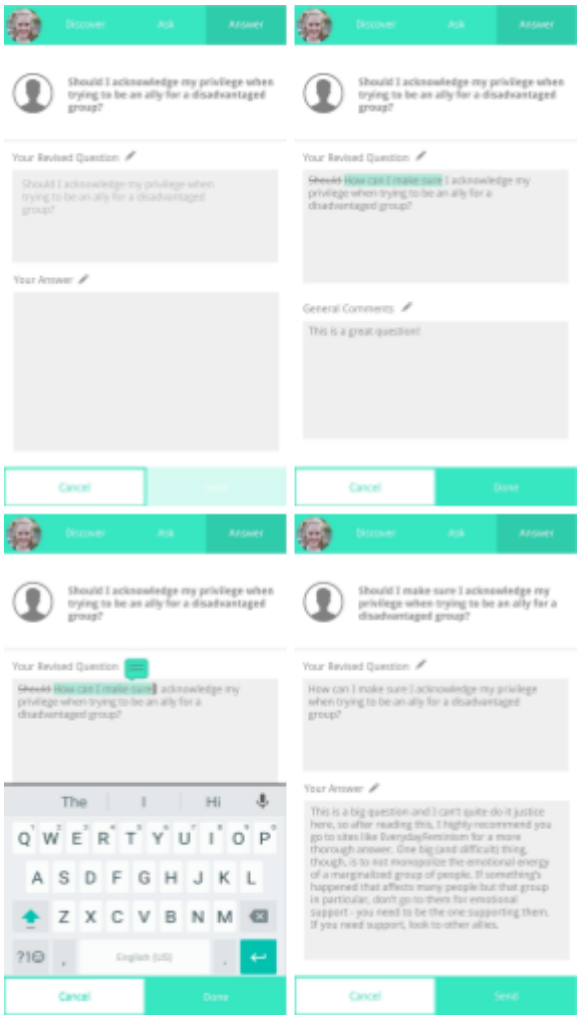
<table>
<tr>
<td> 
</td>
<td> 
</td>
</tr>
<tr>
<td>BEFORE
It may have been unituitive to new users what the motivation behind revising questions even was.</td>
<td>AFTER
We included an information icon that, if clicked, would display information about the revision task and the educational motivation behind it.</td>
</tr>
</table>

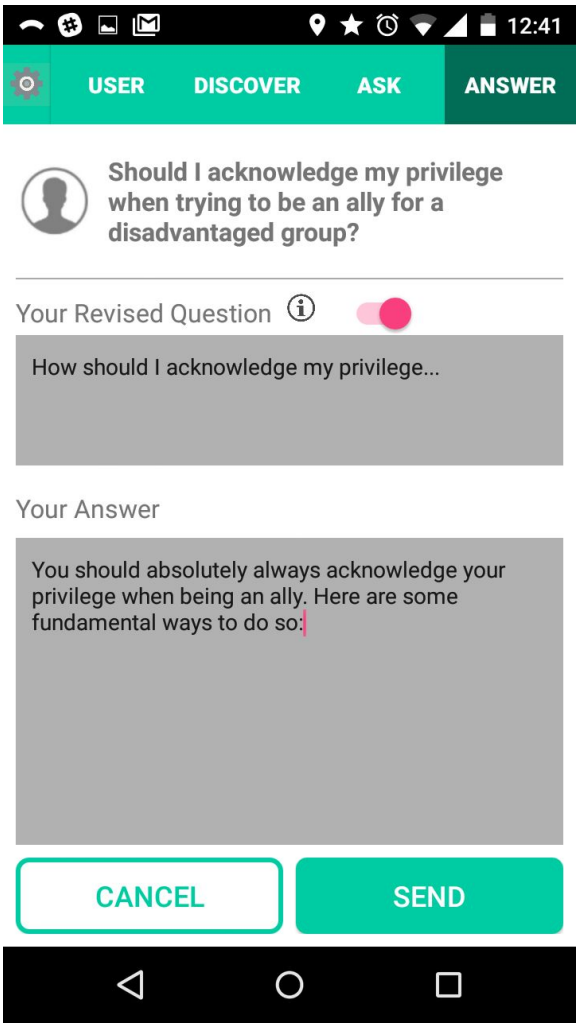
6. NO WAY TO FIND OR ADD FRIENDS IN THE PROTOTYPE (H2: MATCH BETWEEN SYSTEM AND REAL WORLD, SEVERITY 3)

<table>
<tr>
<td> 
</td>
<td> 
</td>
</tr>
<tr>
<td>BEFORE
For our evaluators, it was unclear where the list of users to which questions could be sent actually came from, since there was no place in the app where a user could find or add new friends.</td>
<td>AFTER
We did not yet implement these features, but we included placeholders in the app where users could find and add friends.</td>
</tr>
</table>

In the Settings menu, there is a button to “Find Friends”, which would eventually lead to a search functionality where a user could type in a name and search for people they knew. If a user navigated to another user’s profile page by clicking on a result in that search list, or by clicking on their icon next to any question they answered in the app (a function which is implemented), they would see a button on the profile page that would say “Add Friend” or something similar.. If they hit that, that user would be sent a notification and would have the option to accept the request or deny it.

7. USER CAN'T SEE FULL ANSWER/REVISION/COMMENTS BEFORE SENDING (H5: ERROR PREVENTION, SEVERITY 3)





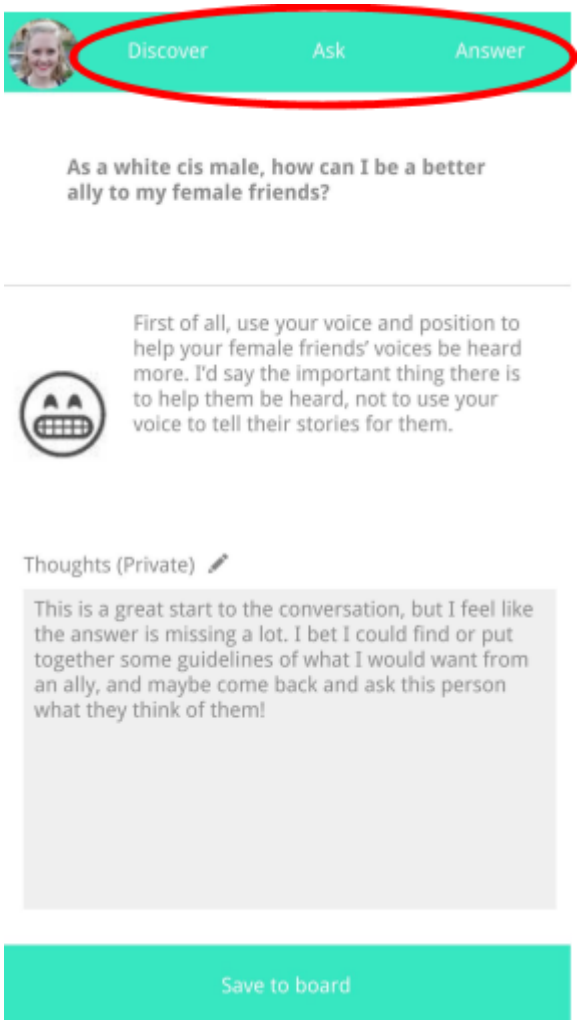
BEFORE

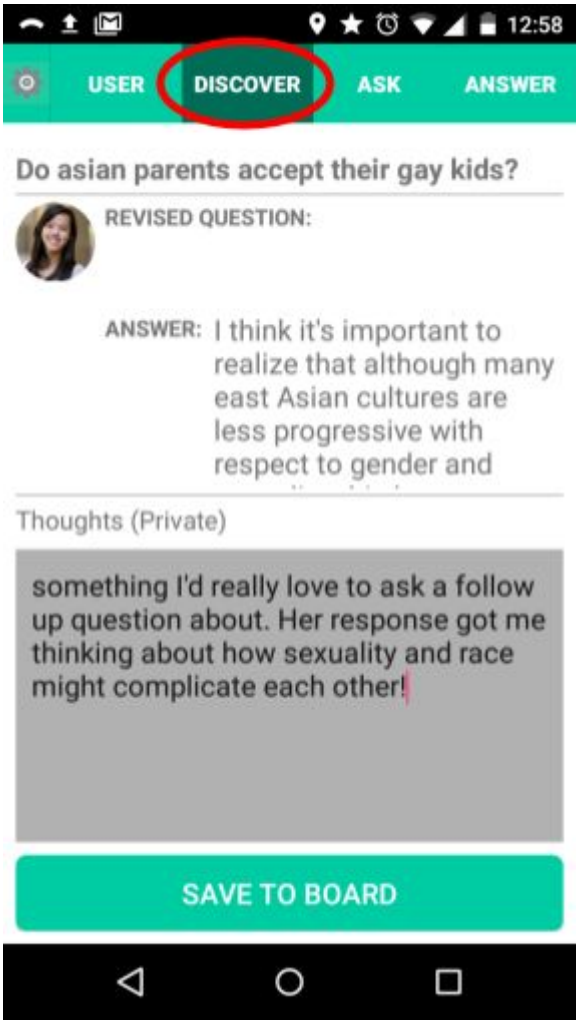
For the Answer task we had a complicated nested task flow that allowed for strikethroughs, highlighting, and two different types of commenting, and at the final page before the user sent the answer, there was no way to see all of the content the user had entered.

AFTER

We vastly simplified the Answer task so that only plain text was allowed in the revision section, so the user can see all of the content at once before they hit "Send".

8. NAVIGATION BAR DOESN'T ALWAYS HIGHLIGHT THE CORRECT TAB (H4: CONSISTENCY AND STANDARDS, SEVERITY 3)





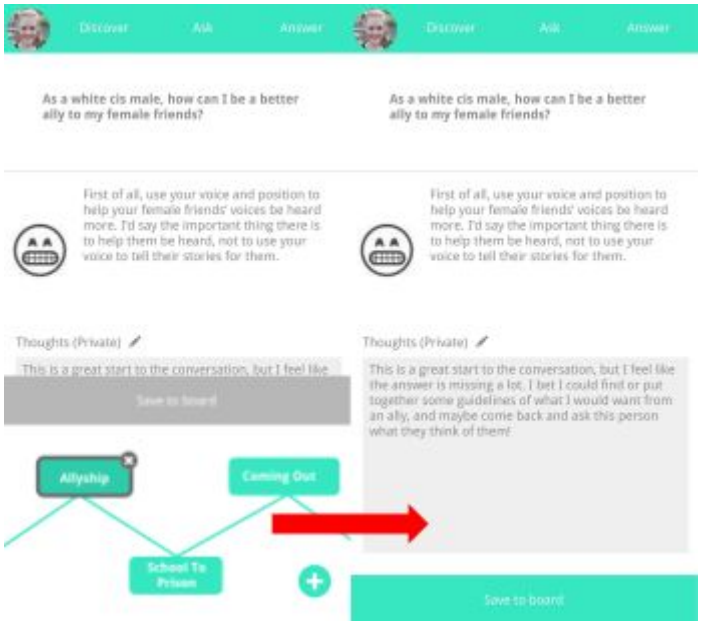
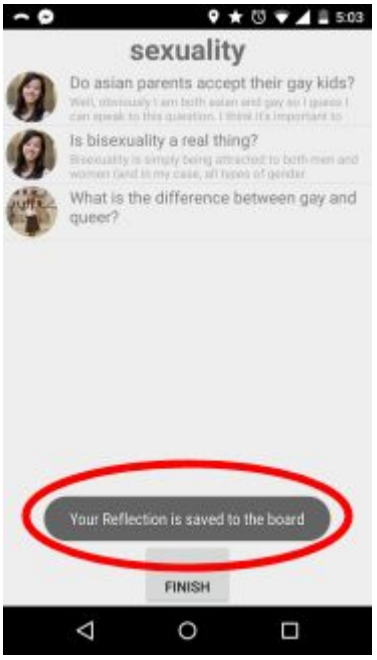
BEFORE

When the user selected a question/answer pair they wanted to view in more detail, they could navigate to that page from either the Discover tab or the Answer tab. However, because the med-fi prototype was implemented with Marvel, we decided not to go through the hassle of uploading multiple different screenshots with the different corresponding navigation tab highlighted.

AFTER

In the hi-fi prototype we were happy to have access to variables that could be passed from screen to screen so that we could highlight the correct navigation tab based on where the user had just come from, so that the user still had some sense of where they were spatially in the app and did not feel lost.

9. NO SYSTEM FEEDBACK WHEN SAVING A REFLECTION TO A BOARD (H1: VISIBILITY OF SYSTEM STATUS, SEVERITY 3)

	
BEFORE When a user selected a board and then tapped away, there was no system feedback confirming that they were successful.	AFTER We gave the user a Toast message to confirm that they were successful.

OTHER CHANGES

- In general, the font size was too small for users to read. Thus, we ended up enlarging the font size in the Hi-Fi prototype.
- We added a tab to the Discover page where a user can view the questions that they have asked other users.
- We redesigned the layout and look of the user's profile in order to make it more intuitive. And, in case the user wants to change their info (profile picture, "ask me about..."s, or name), we enable the users to do so. All the changes users make are persistent in the database, except for the profile picture.
- We added a public profile page such that if a user clicks on another user's profile picture next to a question they've answered, they are directed to a public profile for that user where they can browse questions they've answered and immediately ask that user another question. The function to ask this user a question is only partially implemented because it was not one of our core tasks, but the place to do so on the profile is there.
- If users don't want a huge text box for revising a question when they think the question is worded fine, they can toggle a switch on or off to either see the revise section or not.

PROTOTYPE IMPLEMENTATION

We implemented the hi-fi prototype as a native Android app coded in Java, using Android Studio. As we wanted a fully functioning back-end, we used Google's mobile platform Firebase, which let us have a dynamic real-time database, user authentication, and storage for images (for users' profile pictures). Some of our team had previous experience developing in Android Studio, and all of us had experience coding in Java, so with these tools it was fairly easy for us to ramp up and create a prototype that looked and felt the way we wanted it to. That being said, some of the features in Android Studio were frustrating. The Visual Layout tools do not give a useful preview, especially if we do not have any hard-coded list items in the listview or hard-coded text in the textview. The Gradle build tool also sometimes took a lot of time and and was a little clunky.

As for Firebase, some advantages were that we didn't have to be experts in SQL, which none of us were, and that it had authentication and even an authentication UI built-in. We did run into some frustrating problems initially, particularly with the authentication UI, which were made all the more difficult because online forums like StackExchange are not as robust for FirebaseUI as they would be for something like MySQL. Furthermore, it was slightly difficult reframing our thinking around databases from a SQL/querying format to Firebase's NoSQL/real-time framework. However, it was a positive experience overall using Firebase, as the ability to learn and build the technology quickly outweighed any challenges.

We did not need to use any Wizard of Oz techniques -- asking questions, answering questions, and saving questions to boards are fully implemented and are persistent in the database. However, there are a couple pieces of hard-coded data. Namely, since we have not yet implemented a feature where a user can search for or add friends, for grading purposes we generated accounts and manually made them friends with each other in the database. This is an important detail because a key feature of our product is that users should have a lot of control over who sees their personal content, so only users that are friends with the answerers are able to view those particular questions., which is enforced by the code. The other piece of hard-coded data is a user's profile picture. We dynamically load the images from Firebase storage. However, to change a picture, we implemented the activity but did not yet implement the ability to actually make that change persistent in our database.

As mentioned, functionalities that are currently missing from the product are the ability to search for and add friends, and the ability to actually change one's profile picture. Furthermore, while we support signing in through Facebook, we did not yet implement importing a Facebook user's profile information and friends, which we would want to do. In the future, we would also want to implement some kind of notification system where users would easily be able to see if they've been asked a new question, or if they've received an answer. Additionally, we might also want to give question answerers more control about who could ask them questions and see their content - for instance, a user might want only their friends to be able to ask them questions, but would be comfortable with friends of their friends seeing their answers. Finally, at the beginning of this project we did some research into what factors of anonymous online interfaces or situations could induce benign versus toxic disinhibition, but since then we first had to channel our energy into designing a usable, consistent, and aesthetically pleasing

interface. So, if we were to develop this project further, we would want to go back to that topic and fine-tune our design even more so that the anonymity of our platform encouraged self-disclosure and prosocial behaviors rather than trolling and abuse.

SUMMARY:

During the process of creating Thought Bubble, we were inspired and sustained by the people around us and all the insights, expertise, and heart they brought to this project. Though our vision changed many times during this process, many crucial design points of this application called back to some of our very first needfinding interviews. The realization that personal connections can motivate people to learn outside their comfort zones, that personal narratives can be a key to generating empathy, and of course, that everyone has questions they are too afraid to ask, defined the way we approached our problem space and ultimately formed the core principles behind the design and implementation of ThoughtBubble. Just as our design process would have been hollow without the people who motivated it, our final Android application would be nothing without the empathy, patience, and desire to make the world a better place that we hope our users bring when they decide to engage with it. We were honored to receive the “Best Social Impact” award this quarter, and look forward to giving this project a life of its own outside the scope of this classroom.