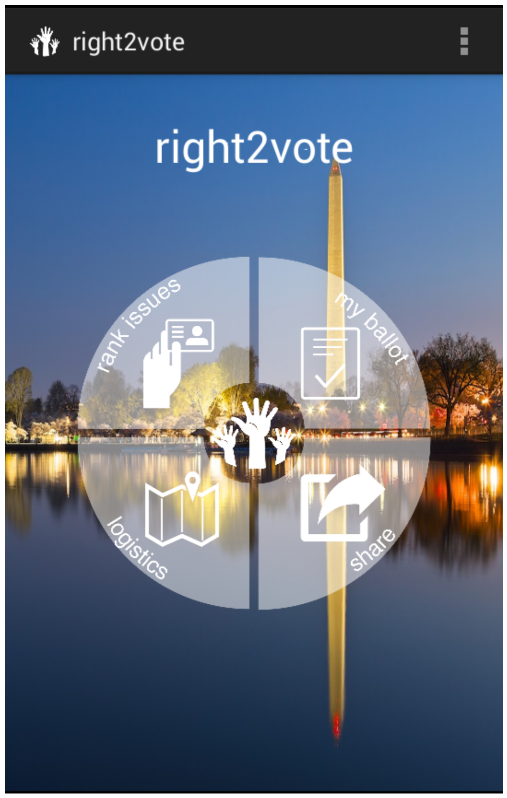
right2vote

[Design] Maya Israni
[Development] Devon Hinton
[Documentation & Manager] Marina Elmore
[User Testing] Christina Gilbert

Incentivizing young voters to make informed decisions and vote.

PROBLEM SOLUTION AND OVERVIEW

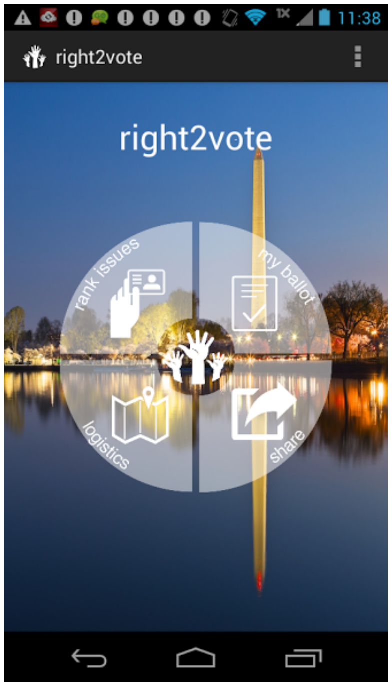
In 2012, only thirty-eight percent of 18 to 24 year olds in the United States voted in the presidential election. Young voters have consistently voted at lower rates than other age groups, a discrepancy harmful to both the young voters and the country. Contrary to some beliefs, younger voters' votes do affect elections (as seen in Barack Obama's reelection in 2012). Furthermore, the younger generations should play an active role in voting because many of these issues will define their future. However, raising voting rates of the youngest sector should be coupled with education of current events and issues. Ultimately, increasing awareness about events and issues will lead to higher voting rates and more educated votes, especially among the youngest voting generation. We will address this problem with our mobile application that will incentivize young adults to vote by providing support during the research and execution stages of voting.

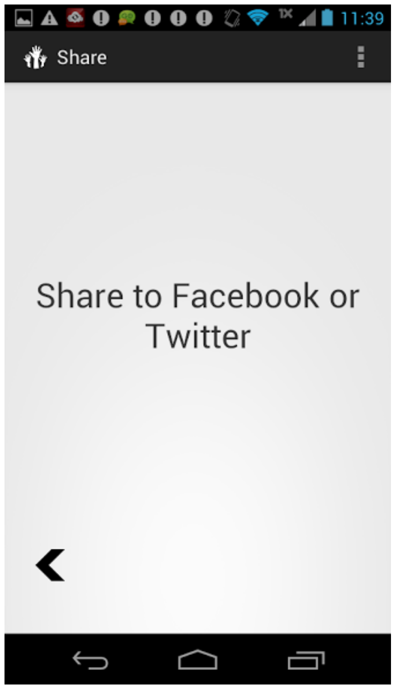


TASKS AND FINAL INTERFACE SCENARIOS

Task 1 - Share - Simple
A user wants to share with his friends that he voted. Suppose a user has just cast his ballot. After casting his ballot, the user wants his friends to know that he voted in an effort to encourage his friends to vote. The user wants to be able to do this task efficiently and effectively.

We chose this task because we felt that having a social aspect to our application would help users feel as if they could share their experience of civic participation. We thought this would incentivize people to vote so they could "share" it with their friends, and also encourage the friends of the original user to download our app.



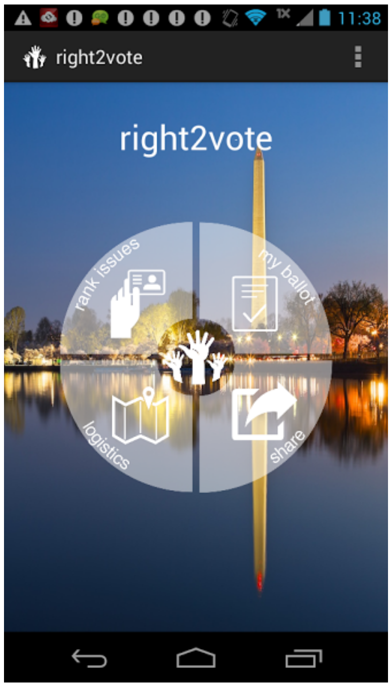


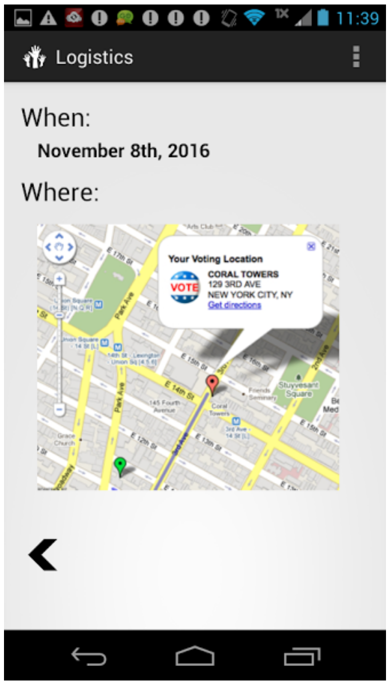
Share to Facebook or Twitter

Task 2 - Support - Medium
A user wants to vote on election day. First, she wants to find the closest polling locations to her current location and/or home address. She will then want navigation directions on how to get to the polling location. Second, she wants to know the date of the election, and the times that the polling location will be open. She wants to add an

event to her calendar with the date and time of the election. Finally, she wants to be reminded the day of the election to vote.

We chose this task because some of the major challenges with voting are finding out when the election is, where the polls are located, and remembering to vote. By assisting the user with these tasks, they he/she is more likely to vote.



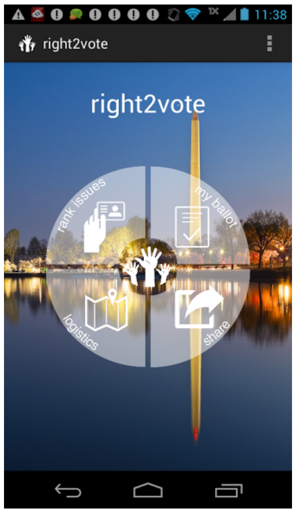


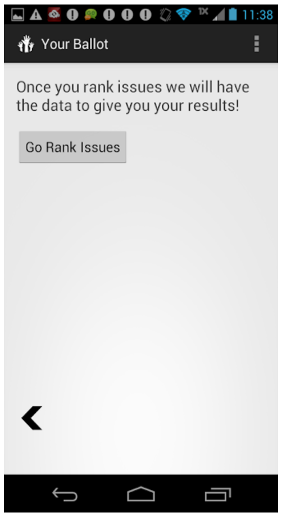
When:
November 8th, 2016
Where:
Your Voting Location
CORAL TOWERS
129 3RD AVE
NEW YORK CITY, NY
Get directions

Task 3 - Rank Issues - Complex

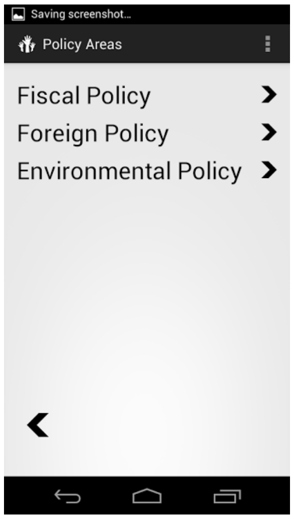
A user wants to find a candidate that matches his opinions. He wants to know how his views on specific areas of policy match up with candidates' so he can make judgements about which candidates to vote for based on how important these issues are to him. He wants to read impartial overviews of candidates' positions on different areas of policy.

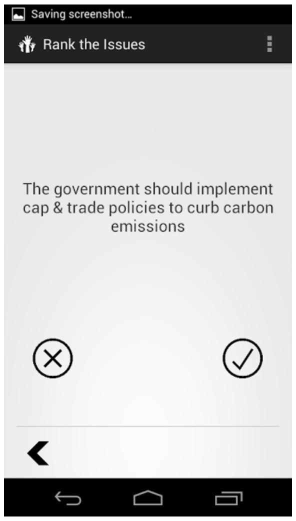
We chose this task because many voters feel as if they do not know enough about candidates to make informed decisions. By using policy statements, we can help users understand how well their views align with those of of different candidates.



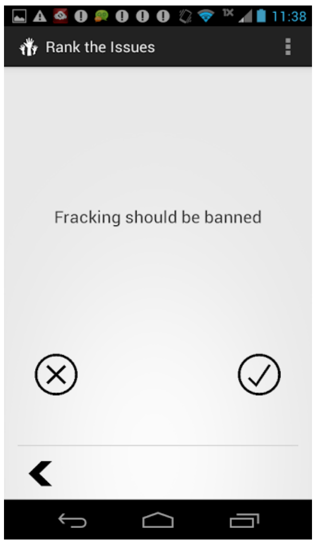


Once you rank issues we will have the data to give you your results!

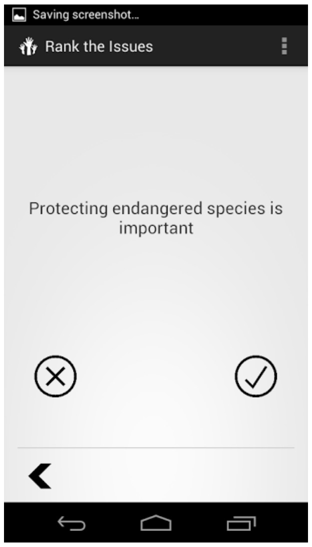




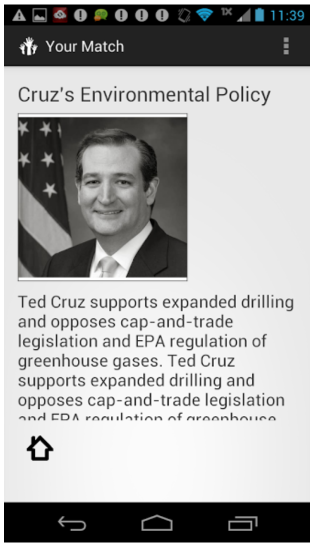
The government should implement cap & trade policies to curb carbon emissions



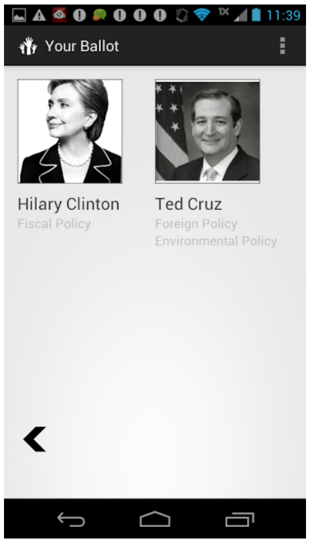
Fracking should be banned



Protecting endangered species is important



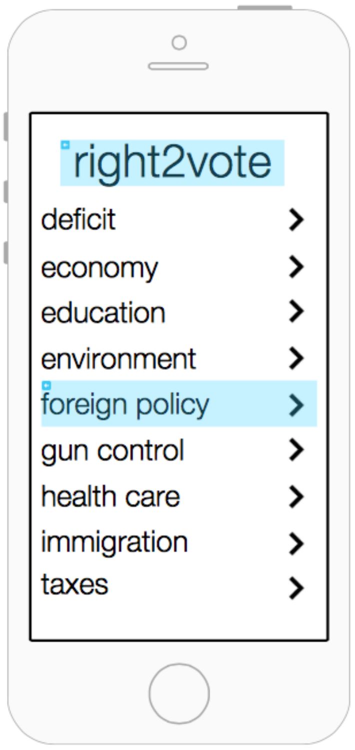
Ted Cruz supports expanded drilling and opposes cap-and-trade legislation and EPA regulation of greenhouse gases. Ted Cruz supports expanded drilling and opposes cap-and-trade legislation and EPA regulation of greenhouse

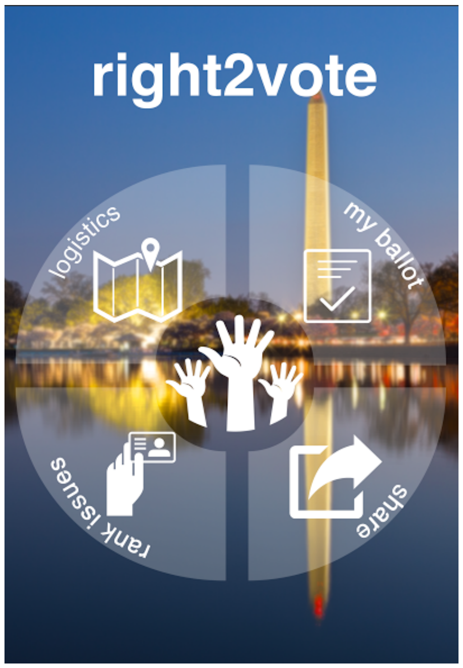


MAJOR USABILITY PROBLEMS ADDRESSED

1. [H2-10 Help and Documentation] [Severity 3] [Found by: A, B, C]
The home screen is not self-explanatory; without prior knowledge, there is no documentation to tell the user to click on the 'policies' to find a candidate, or to click on the title to get to the ballot. The user needs some help to be able to figure these out, especially the first time. A tutorial should be added, or some subtitles/headers to tell the user how to use it.

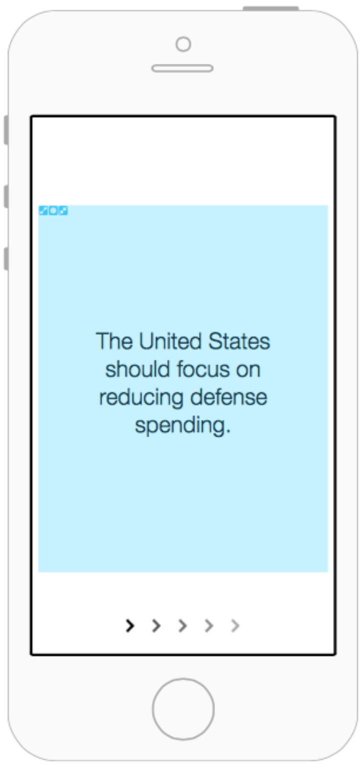
We entirely redesigned our home screen page to include four simple icons, along with a quick subtitle for each. Before, the home screen only included the “issues” task, and the other three icons were embedded within the right2vote title. Now, our homescreen delineates all four tasks through simple icons.

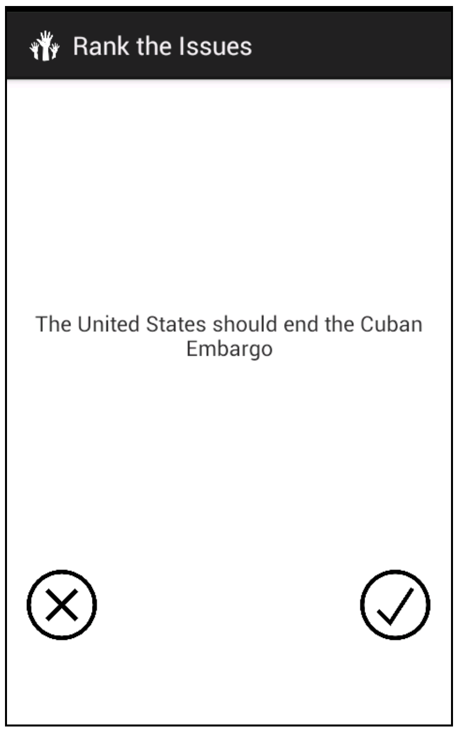




2. [H2-4 Consistency and Standards] [Severity 4] [Found by: A, B]
The pinching in and out to agree/disagree with a statement is not intuitive. In the real world, one agrees generally by checking a box, not by pinching – even in other apps, pinching is not used to agree or disagree - it is used to zoom in and out. To fix this, a checkbox yes/no or a vote yes/no button should be added instead of the pinching motion.

We decided to include two icons (a cross and a check) on either side of the screen to indicate which side to swipe on to agree and disagree. The user can click this icons to agree/disagree. This increases the visibility of the agreeing and disagreeing motions.


The United States should focus on reducing defense spending.

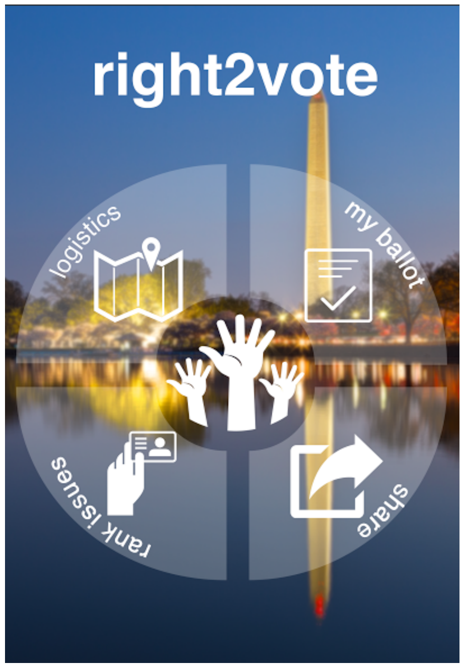


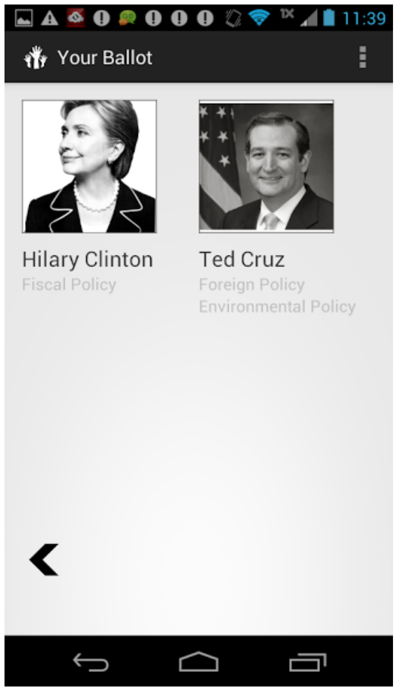
6. [H2-6 Recognition Rather than Recall] [Severity 3] [Found by: A, B, C]
The way to get from the home screen to the ballot is to click on the right2vote title. In the real world, someone picks up a ballot by taking it from a box, or something specifically marked for ballots. Generally, clicking on titles of apps takes users to a home page, not to a specific sub-page in the app. The app should have a button labeled 'ballot box' or something similar, to allow the user to not be confused.

As mentioned above in the heuristic violation #1, we redesigned the home screen to increase visibility of the system. We no longer have any "embedded" features; all transitions are labelled with subtitles.

8. [H2-10 Help and Documentation] [Severity 4] [Found by: A, B, C]
The ballot just shows the candidate, with categories listed underneath them. It is not immediately obvious that these are the categories which the candidate agrees most with the user on. In fact, some evaluators thought that those categories were the primary platforms the candidates were running on. It should be made clearer, perhaps through explanation at the bottom of the screen, that the categories under the candidate are those which the candidate agrees most with the user on.

In this feature, we did quick extra user testing on our hi-fi prototype and found that this was not an issue with the redesign of our prototype. With the new home screen, after ranking an issue and then selecting the ballot, it was clear to the user what our “ballot” feature was supposed to do.





9. [H2-4 Consistency and Standards] [Severity 3] [Found by: A, B, C]
The check mark used to get to the ‘voting logistics’ page is not immediately obvious. In most apps and software, people schedule things with a calendar, not using check marks. The icon to reach the voting logistics page should be a calendar icon or something similar.

As mentioned above in heuristic violations #1 and #6, we redesigned the home screen to increase visibility and consistency of the app. The logistics transition includes an icon along with a subtitle. Please see heuristic violation #1 for further description.

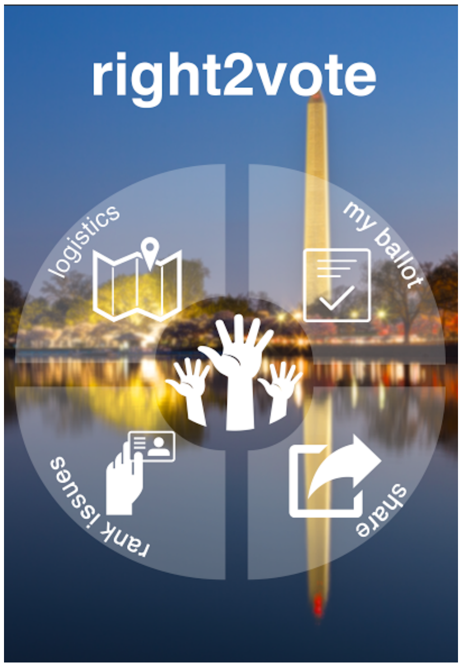
13. [H2-7 Flexibility and Efficiency] [Severity 3] [Found by: B, C]
If a user is interested in looking at a list of candidates instead of a list of voting topics, it seems like he or she would have to put all the candidates on his or her ballot to get a bird’s eye view. While the topics funnel is useful for people who only care about a few

topics, some users might want to start with candidates and narrow their choices down using topics. Add a pathway that lets users look at candidates instead of topics.

This suggestion does not align with our application's mission. We are trying to align people with candidates based on issues, not party alignment, parental views, or previous beliefs. We believe that it is important to provide an approach that moves from issue to candidate. This was evident early in the design process when a team member was talking to a participant who said she always votes on the same party lines as her parents. After demoing our app, she was matched with a candidate from the opposing party and was shocked and impressed that her personal beliefs actually aligned more closely with another candidate.

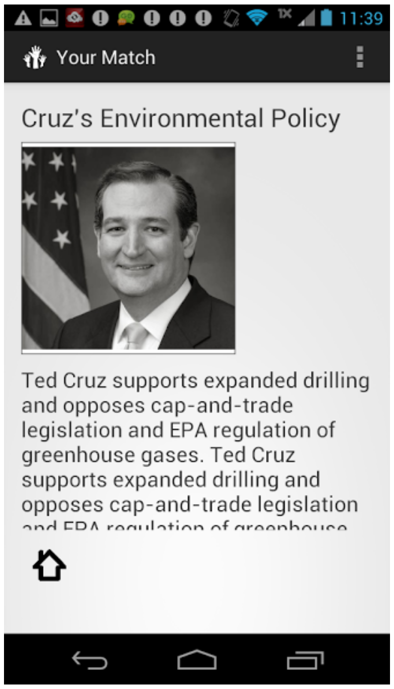
14. [H2-7 Flexibility and Efficiency] [Severity 3] [Found by: B, C]
Because the ballot is not actually cast from within the app and because the voting center is the same regardless of ballot choices, it doesn't make sense to have to build up a ballot before going to voting logistics. It would be more efficient for the voting logistics page to be accessible before having to complete the ballot.

We redesigned the homepage to be able to access both voting logistics and the ballot through the home screen of our application.



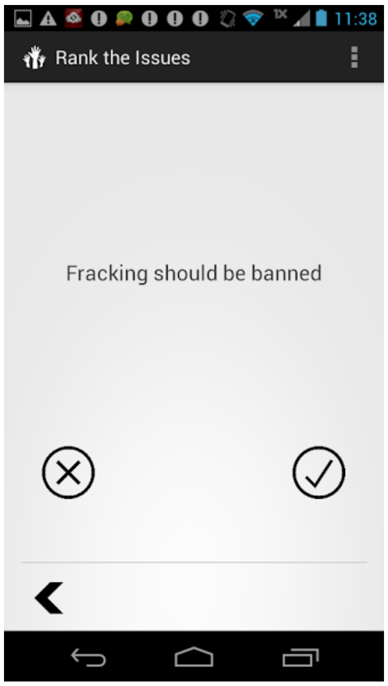
16. [H2-4 Consistency and Standards] [Severity 3] [Found by: B]
Depending on the path that the user has already taken, selecting the right2vote logo does different things. On most platforms clicking the logo gets the user back to home, but this isn't the case, so it is confusing. Make the logo be a pathway to home or at least consistent.

This was an error on our part, and we are more consistent in our final design. The "home" icon now consistently goes home, and the back arrow icon consistently goes "back" (see below, bottom lefthand corner)



Cruz's Environmental Policy

Ted Cruz supports expanded drilling and opposes cap-and-trade legislation and EPA regulation of greenhouse gases. Ted Cruz supports expanded drilling and opposes cap-and-trade legislation and EPA regulation of greenhouse



Fracking should be banned

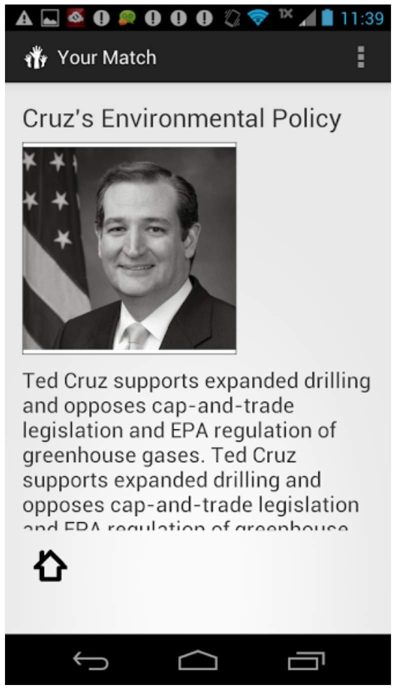
Other Changes:
4. [H2-3 User Control and Freedom] [Severity 2] [Found by: A]
There is no way for a user to go back and change their opinion on a policy statement. Peoples' opinions on policy change, so they should be able to update their opinions. There should be an option to scroll backwards to change previously formed opinions.

Users can now run through the policy statements on a particular issue as many times as they want, and this will update if the have different preferences.

5. [H2-1 Visibility of System Status] [Severity 1] [Found by: A]
There is no display in the app that shows the candidate that was selected for a voted on policy other than on the final candidate screen and the ballot. It would be good for

the user to be able to see the results of their choices immediately, perhaps in a subtitle under the policy name on the home screen.

After each set of policy statements, the user is shown which candidate matches with this set of views. This screen also has an overview of the candidates positions on this issue. Below is an example of this screen, this one is for when the user gets Cruz' on environmental policy.

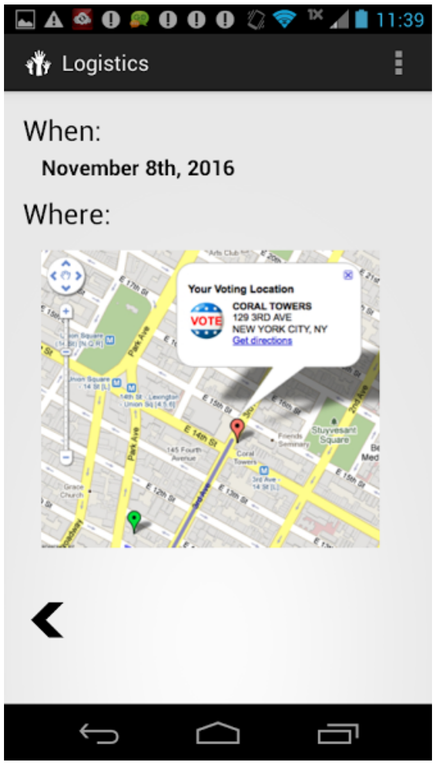


Cruz's Environmental Policy

Ted Cruz supports expanded drilling and opposes cap-and-trade legislation and EPA regulation of greenhouse gases. Ted Cruz supports expanded drilling and opposes cap-and-trade legislation and EPA regulation of greenhouse

17. [H2-4 Consistency and Standards] [Severity 2] [Found by: C]
When users enter the map to view voting locations, there are certain locations marked with pins, and one marked with a circle. The user may be confused as to what the different symbols mean. There should be a legend on the map to indicate what the various symbols are.

The voting location now is labeled as such with a pop up window, to make it clear which is which. The window also has more information about the polling location. Because there are only two pins, we didn't want to clutter the interface by labeling the second one. Because it is a different color from the polling location pin, it is clear that it does not represent another polling location and represents your current location.



DESIGN EVOLUTION

Initial Sketches

The initial application idea that we built off of was "Issue Priority Ranking." This three stage process would guide votes through understanding their political views and matching those views with candidates *(see Figure A)*. During the first stage, the user would be presented with cards with quotations or ideas written on them stating some political view. The user would agree or disagree with the idea or statement. Each card would also have an option to learn more about that subject with a menu of news articles on the topic Once the user completed this deck of cards, he or she would be guided the second stage, which would have a menu of issues, and would display where the user would fall on the spectrum of this issue. He or she would then be asked to rank the level of importance of each issue. Once complete, the user would be guided to the third stage, where there would be a menu of candidates. When clicked on, the user would be able to see a visualization of the candidate's political positions overlaid with his or her own. They would also be able to click on news articles relating to this candidate.

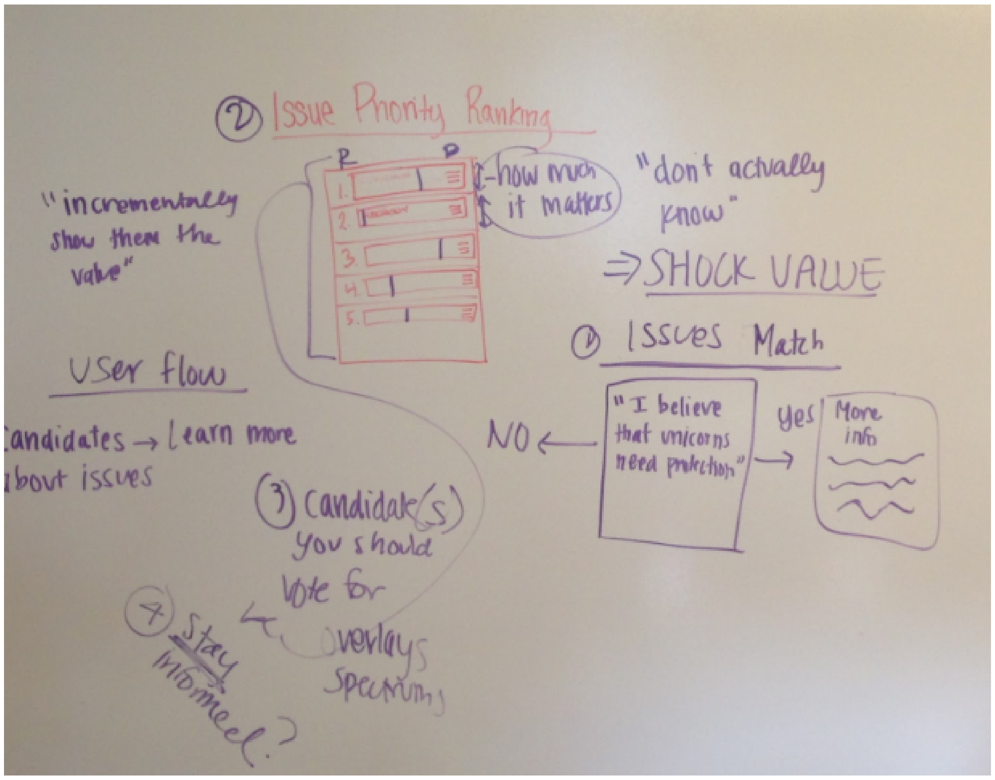


Figure A: Brainstorming Issue Priority Ranking idea

We developed this idea as a result of our contextual inquiry and task analysis where we interviewed three distinct users about their voting techniques. We chose to focus on the 'issue priority ranking' idea for two reasons. Firstly, this idea met all conditions required: feasibility, interest, and significance. The multi-stage progression of this app proves both interesting and significant. The implementation of this idea's design and user interface is also feasible. Secondly, this idea addresses multiple tasks and issues associated with our problem. The ranking system allows for young voters to learn more about their own views in comparison to candidates', and incorporation of news articles allows for more informed votes among the youngest sector, our ultimate goal.

Rough Sketches

Building upon our "issue priority ranking" idea, our group sketched numerous designs *(see Figure B)*. We focused on the overall flow of the app, user interactions (especially when agreeing/disagreeing with a policy statement), and order in which the user completed the tasks.

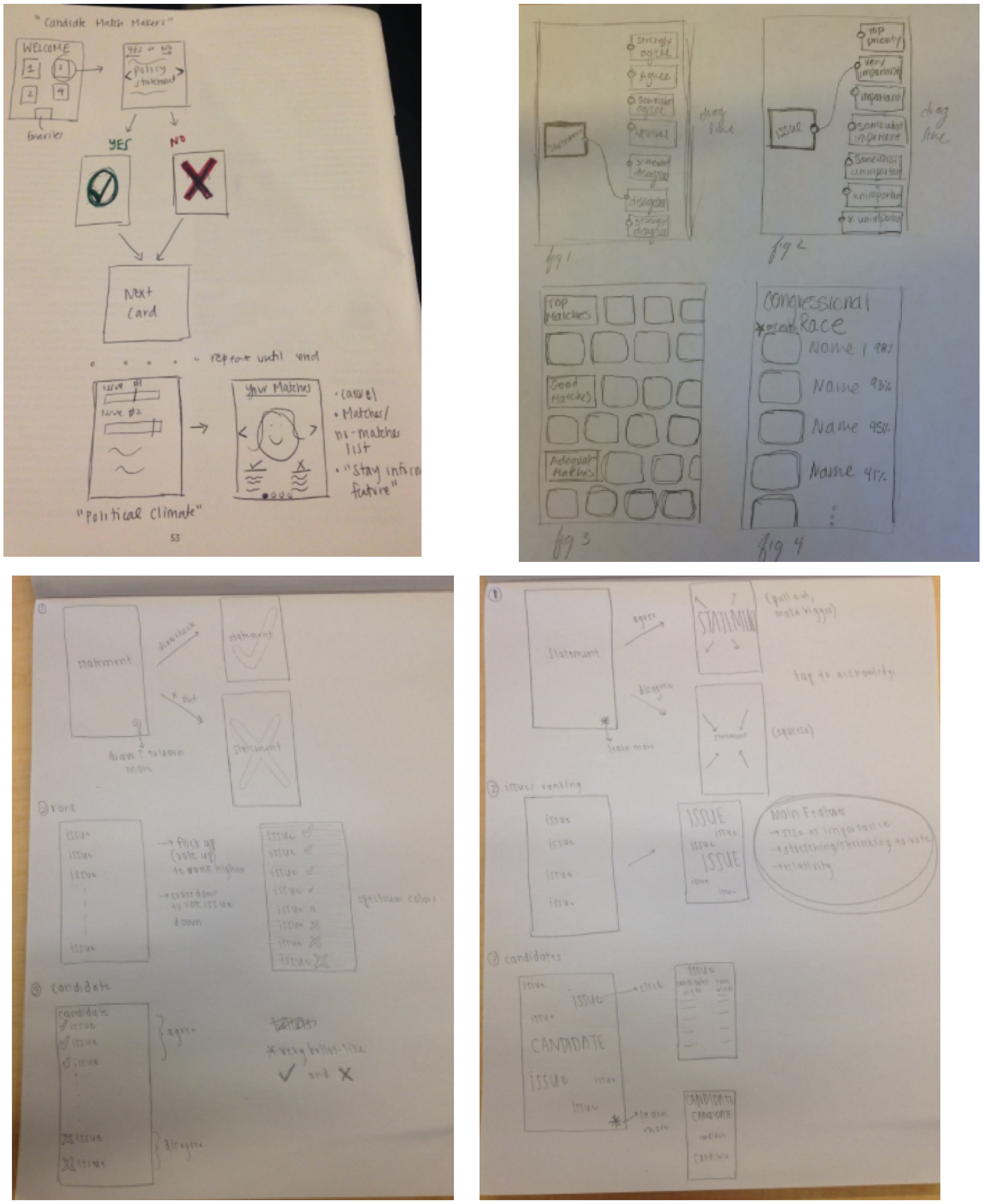


Figure B: Rough Sketches of "Issue Priority Ranking" Application Idea

Low-Fi Prototype

Our low-fi prototype included major interface screens imposed on top of an iPhone 5 *(see Figure C)*. These screens included the main home screen, policy statement screens, candidate screens, ballot screens, and a voting screen. It guided the user to rate his/her opinion on specific issues, formed the users' ballot based on the user's stance on these issues, and provided logistical information on how to cast one's ballot. All screens were drawn in only black ink.

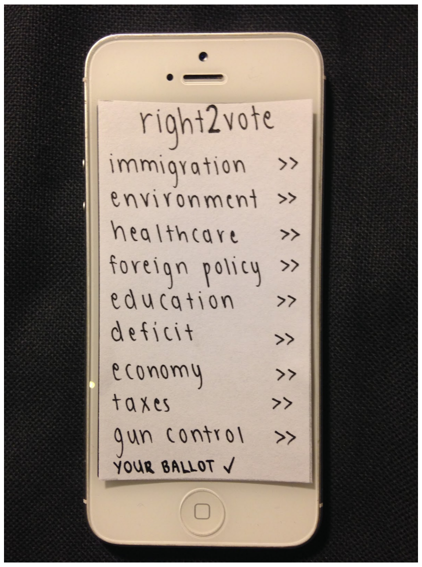


Figure C: Low-Fi Prototype home screen

To rank an issue, the user swiped across the issue title on the home screen. When the user swiped across an issue, a policy statement about the issue appeared. The user continued to swipe in agreement (left) or disagreement (right) with each statement. In this prototype, we had three screens of policy statements regarding foreign policy. At the bottom of each policy statement screen, the page indicator (three dots) indicate how many more policy statements the user needs to answer within the issue category. After the user finished answering all policy statements within an issue category, a screen popped up indicating which candidate's platform was more aligned with the user's on that specific issue. See *Figure D* for full layout of the prototype.

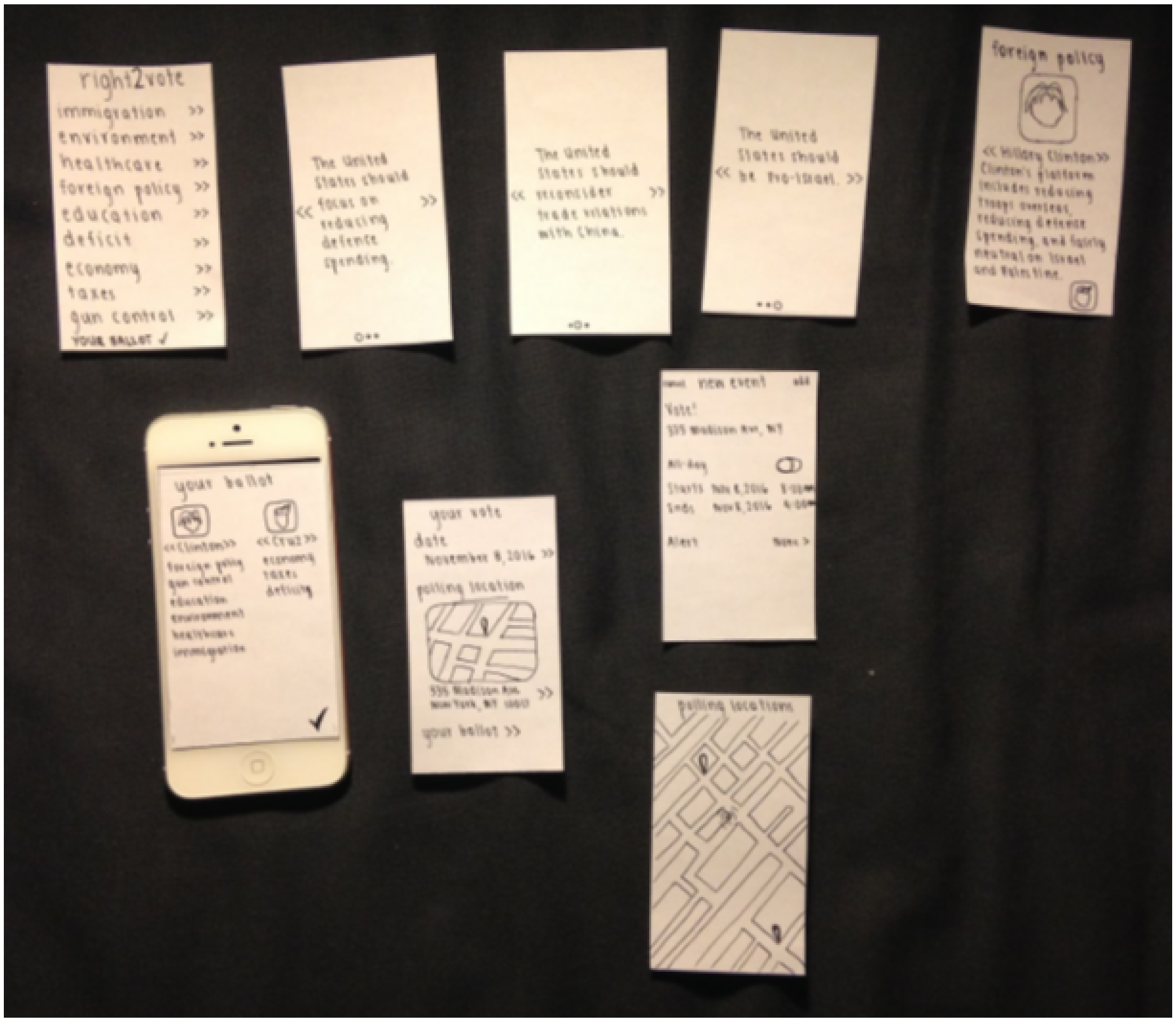


Figure D: Full layout of Low-Fi Prototype

We tested this prototype through usability testing on three distinct participants. We used the paper prototype imposed on an iPhone 5 screen and one of our team members switched the screens manually as the user went through the app. From this experiment, we took away two categories of improvement — UI and structural. The first UI change fixed the confusion of 'which swipe direction is agree?' We overestimated the affordance of three 'arrows'. Tinder's interface is not so common as to become as second natured as Apple's mouse or the iPhone home button. We needed to give users clearer indications of which direction to swipe to agree or, alternatively, use a different mechanism than swiping. The second major UI change fixed confusion on how to utilize the ballot page. We want to make this page more valuable than the visual presentation of your choice of candidates. We needed to brainstorm but, tentatively, wanted to add links to more resources as well as a more granular breakdown. The third major UI change will be to fixed confusion with the arrows around the names of candidates. Because we used arrows elsewhere on the interface to indicate swiping, he thought they also indicated swiping in places that they did not.

Finally, our structural take-away: the users felt the transition from swiping on issues to picking a candidate was too abrupt. The candidate choice was not supported by enough data. Whether this will be solved when the user has to go through every policy area (Foreign Affairs, the economy) and answer more policy questions, is yet unclear. We wanted to test this more, especially as we developed our medium-fi prototype.

Medium-Fi Prototype

In this new design we made several changes to the design interface from our low-fi prototype. We decided to start with our model in grayscale in order to provide the most clear medium-fi prototype and help the user understand the hierarchy of what is important in our app.

For the first change, we added different interactions from selecting the “right2vote” logo *(see Figure E*). This interaction allowed the user view his or her current ballot. The improved ballot organizes the previously ranked issues into an easy-to view format. It makes the candidate-issue breakdown very obvious for the client.

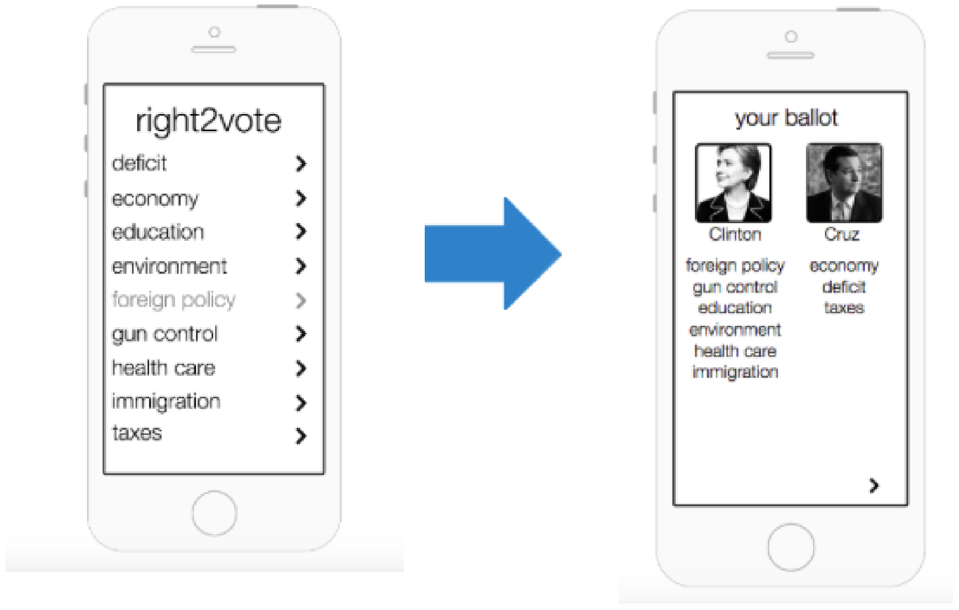
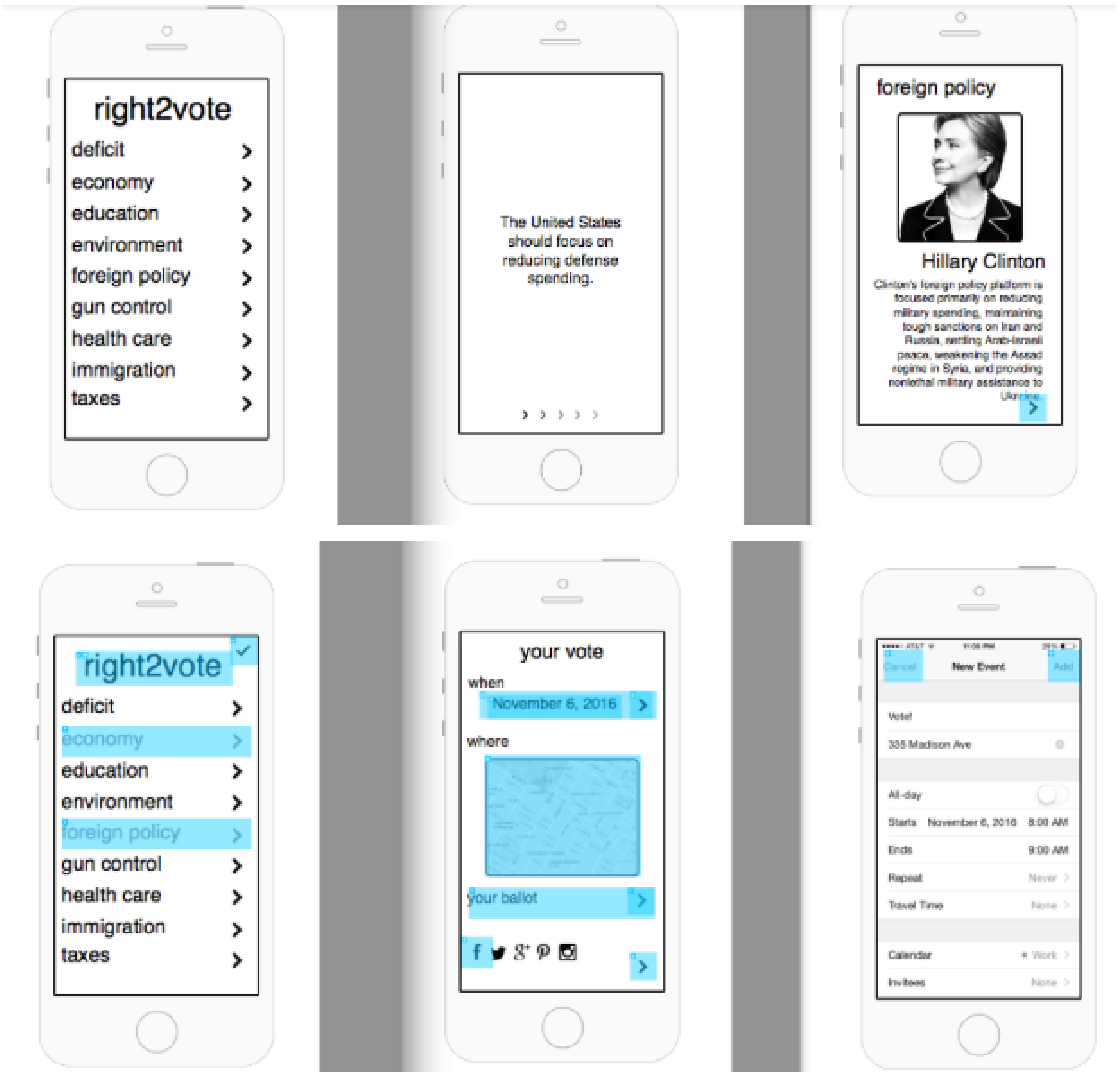


Figure E: Transition to ballot screen on medium-fi prototype

The second interaction from the homescreen is a combination of new design and design features that had positive feedback from low-fi user testing. In the the low-fi prototype, the user swipes left and right through a series of policy statements. However, in this new design the user shrinks or expands the statement based on if they agree with the policy statement or not. This task is accomplished with either pinching or stretching the policy statements based on the users opinion. This also means that we got rid of the confusing arrows on either side of the policy statement. We also wanted to address the complaint from one of our participants that transition from policy statement to candidate was too abrupt. For this, we included more obvious arrows at the bottom of the screen to track the progress through the different policy statements.

The third interaction from the homescreen is to get logistical support for casting the users ballot on election day. This can be accessed by tapping on the right2vote icon and then selected the checkmark in the upper right corner. Tapping the arrow takes the user to the screen for logistical support on election day. Overall, subjects in our low-fi user testings conveyed to the team that this layout was both effective and visually appealing so only minor aesthetic changes were made. See Figure F below for layout of this medium-fi prototype.



The United States should focus on reducing defense spending.

Clinton's foreign policy platform is focused primarily on reducing military spending, maintaining tough sanctions on Iran and Russia, settling Arab-Israeli peace, weakening the Assad regime in Syria, and providing nonlethal military assistance to Ukraine.

Figure F: Layout of medium-fi prototype

The medium-fi prototype was evaluated using a heuristic evaluation completed by another design group in our CS147 class. This group was in another studio, with the same focus on Behavioral Change. The results from this heuristic evaluation all centered around two main issues: current app lacks consistency and is therefore needs a simplified overall flow, and the general concept is too complicated for first-time users. We addressed both of these issues in our final high-fi prototype below.

High-Fi Prototype

Our final prototype included a completely redesign home screen *(see Figure G)* that increased overall visibility for the user. Separating the four main tasks allows for efficient access through recognition of icons and confirmation through subtitles. The simple homepage embodies one of our group's main goals: simplifying the voting process.

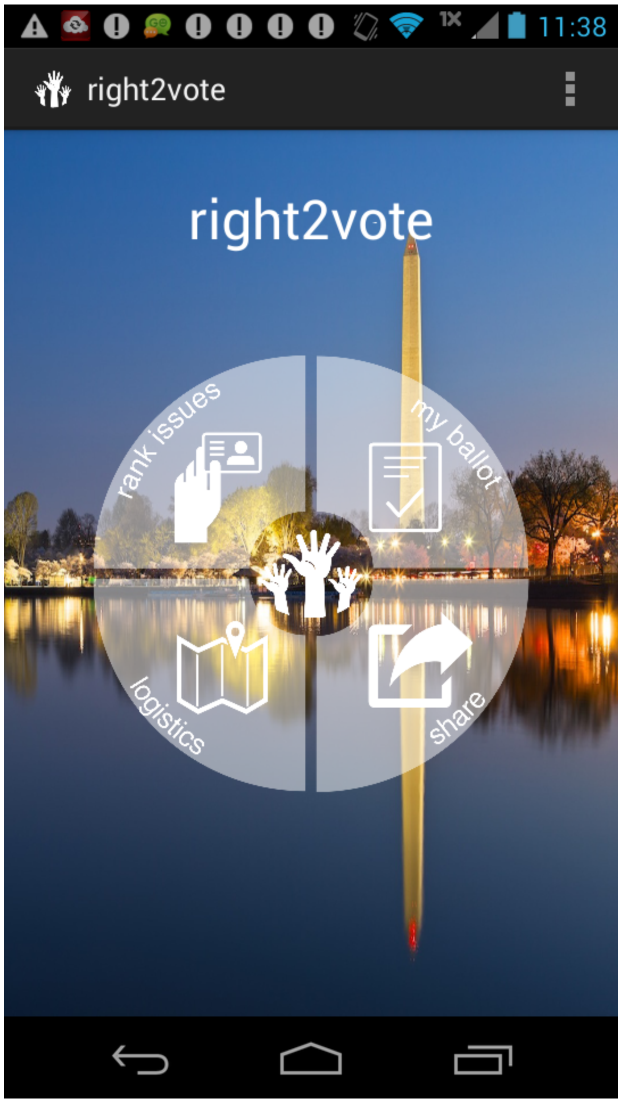
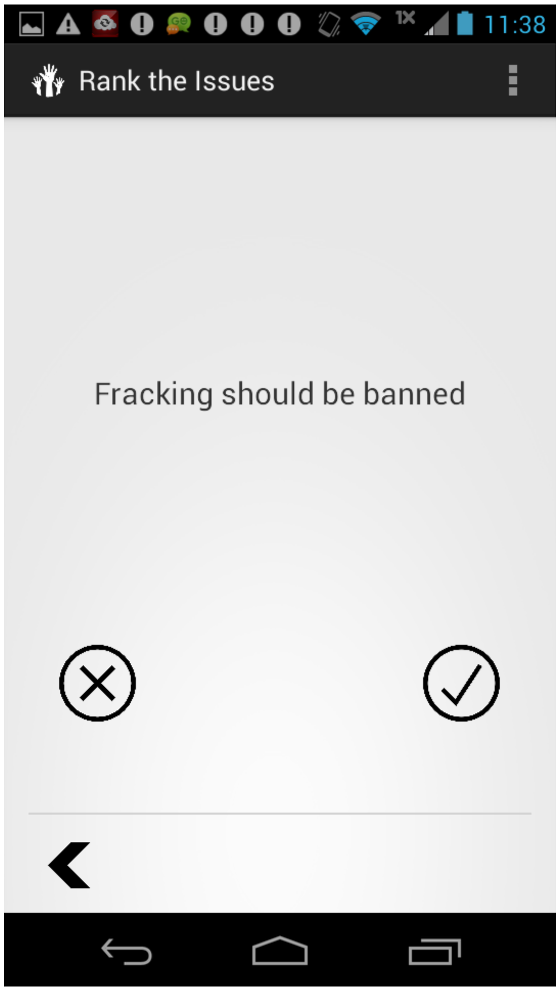


Figure G: Home screen of high-fi prototype

We included icons for users to use when ranking policy statements *(see Figure H)*. Our goals to simplify the app, increase user visibility, provide more frequent and constant feedback, and encourage consistent and efficient use ultimately led us to this final high-fi prototype design.



Fracking should be banned

Figure H: Voting on a policy statement regarding environmental policy

---

PROTOTYPE IMPLEMENTATION

Tools

We built the final hi-fi prototype for Android using Java on Eclipse. The tools helped because Java translates quickly into a full fledged application. The cost to make a brief application and get it working are not too high. In this way, it allowed us, an un-Android-experienced bunch, to quickly get up to speed and build our application. Eclipse itself was helpful in allowing easy testing. As any experienced developer knows, it is the time from “I changed code” to “I ran my program and tested it” that determines rate of code completion. Eclipse has a “one click option” that allows you to run your application either on the built-in emulator or on an actual device connected via the USB port.

These tools, while helpful for quickly getting a basic application working were not helpful, and in fact were harmful, in finishing the application. As we discovered the Android/Java/Eclipse has a few fatal flaws, two of which we will enumerate here. First, memory management. Though Java is meant to abstract the developer away from working with memory allocation (like one does in raw C), Android stores object so unintuitively behind the scenes that having just a few images can cause “out of memory” errors. This is a problem we ran into when moving from “make it work” to “make it look gorgeous”. We spent a lot of time trying to figure out the root of our issue but it appears easy memory management on Android/Java is an open problem. In the end, we opted to write our own image freeing code followed by calls to the Java Garbage Collector. Second, and last, Eclipse is unreliable. From time to time it fails for no reason (hence the existence of this http://www.ihateeclipse.com/). This caused both frustration and false positives for bugs: “I broke the app with this line of code. O wait, no, Eclipse is just messing up”.

Wizard of Oz

We did not use any Wizard of Oz techniques.

Hard-coded Data

The only hard-coded data we had is the map of polling location in the logistics section. We did this because it is hard to get accurate data for polling location for the 2016 elections. The rest of the application is dynamic. We are proud to say that our technical design is flexible enough that you can pick any number of policy statements or policy areas (Education, Fiscal Policy...) and our application will work correctly. This includes the dynamic scoring. When swiping through issues you can change which candidate you end up matching with (each statement has a candidate who supports the statement. If you agree it is a point for that candidate, if not, then vice-versa).

Future Add-ons

There are two things we wanted to build but did not have time to: touch gestures and finished notifications. We wanted to make swiping the action users take to indicate their stance on a policy issue. Unfortunately, we did not have time to make this happen. Additionally, on the policy areas screen (list of Fiscal Policy…) it would be nice to indicate some UI to show when the user has already gone through and rated the statements in one of the categories.