

# covalence

connect your world.

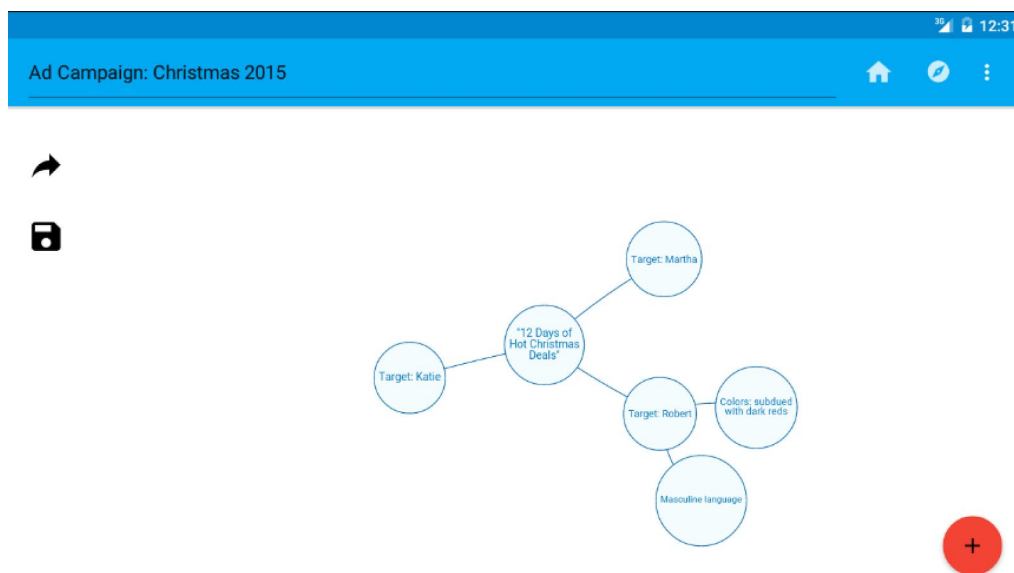
Courtney N. / Ted L. / Logan S. / Emma T.  
CS 147, Fall 2015

Covalence is a mobile workspace for concept maps,  
helping teams collaborate by connecting  
images and text in a visual, intuitive way.

# Problem and Solution

Covalence is a visual, mobile workspace that allows you to create connections between written concepts, images, and other resources like never before. Learning today is scattered across PDFs, course websites, handwritten notes, and Powerpoints -- it's hard for students and teachers to draw connections across many different mediums.

Covalence has a lightweight interface with very intuitive gestures for creating and connecting nodes, lowering the barriers for brainstorming and exploration in any topic. By focusing on the need for flexibility, Covalence allows users to pull together all the resources that are most important for learning and synthesis, whether that's in the classroom to study or in the boardroom to brainstorm.



## Tasks & Final Interface Scenarios

### Simple Task: Sharing

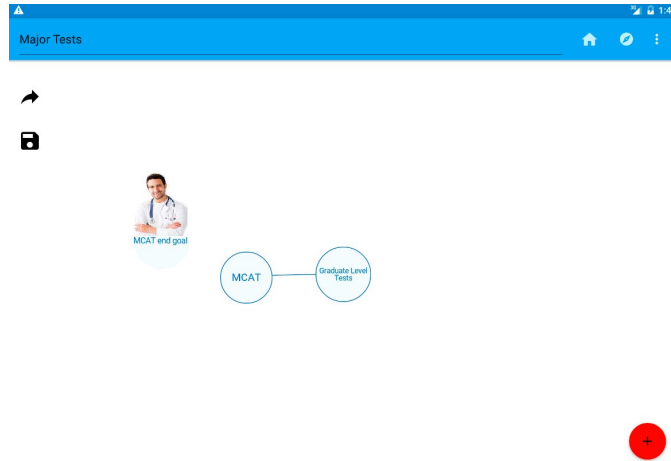
**User goal:** Share connected ideas with others.

**Interface:** Share a concept map you've created.

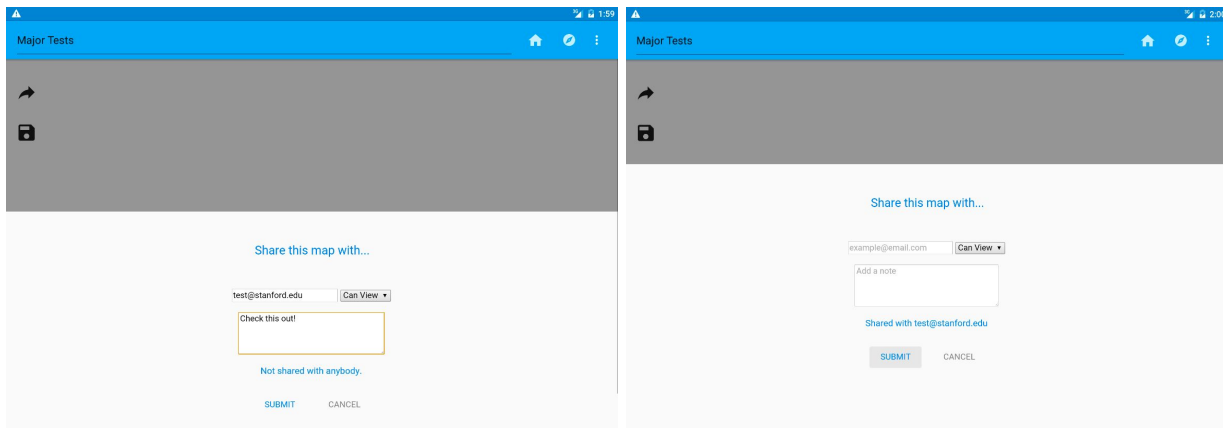
We chose this task because collaboration is central to Covalence's success -- brainstorming and learning sessions, even during our paper prototyping, were more successful when multiple people participated. In order to facilitate collaboration, Covalence needs to make it easy to invite people to share in your creative process. We made the sharing function available directly on

every map, and the sharing dialogue pops up over the map, keeping it part of the creation workflow.

*Click 'share' button in the top left*



*Fill out share pop-up dialogue*



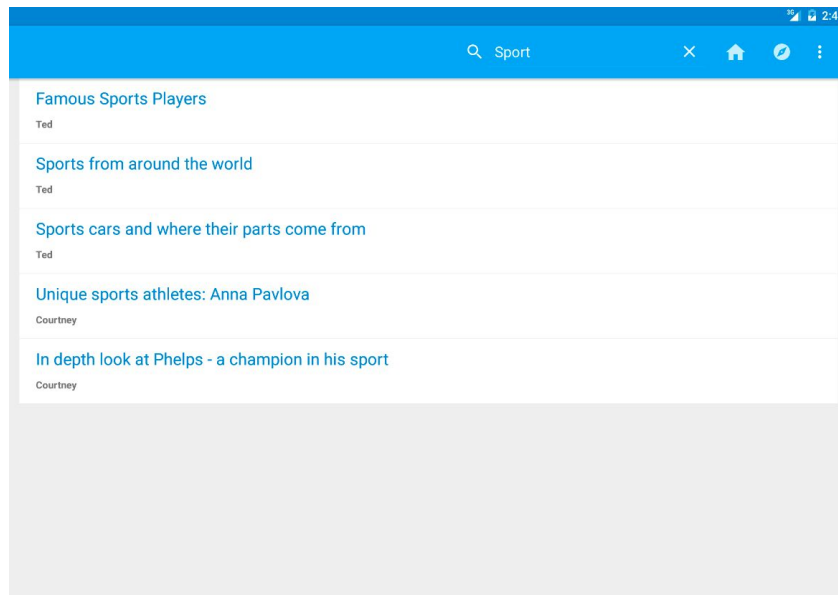
### Medium Task: Exploring

**User goal:** See how other people have connected ideas you're thinking about, and use the community to build/enhance your own understanding

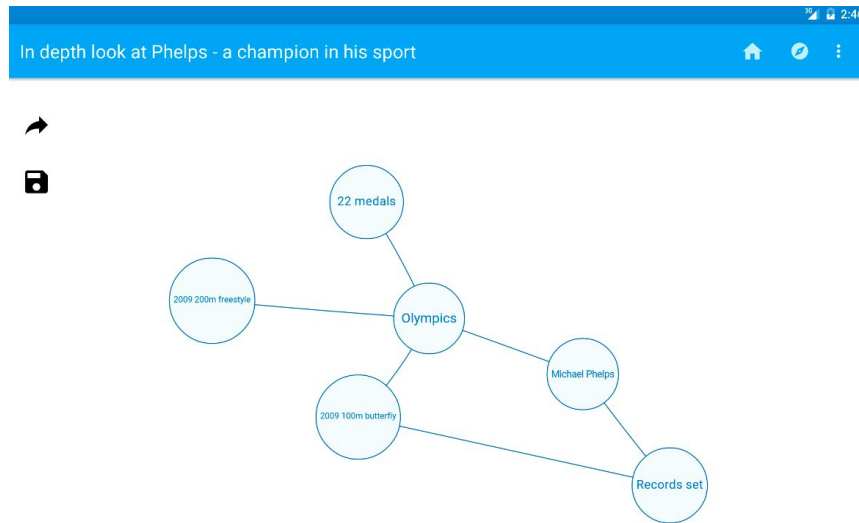
**Interface:** Use the explore menu and a search bar to search for related maps.

We chose this task because we think one advantage of Covalence will be in its creative community -- you don't always have to start from scratch, you can build off of the work of others. If you're studying for a class, see maps that have been generated by other students from previous years. If you're planning a holiday marketing campaign, check out last year's brainstorm to see the great ideas that weren't quite in your budget.

### Enter text into search bar



### Load and view desired map



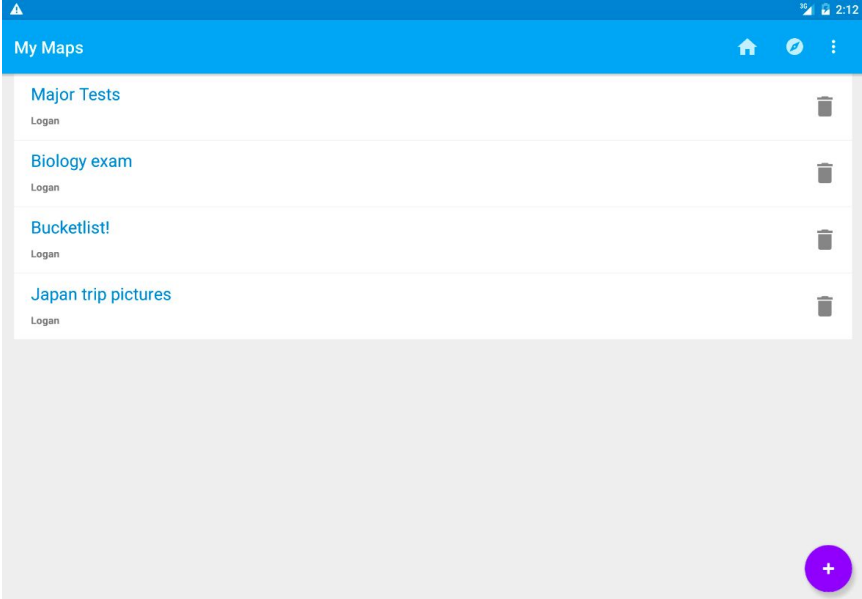
### Complex Task: Creating

**User goal:** Connect abstract ideas.

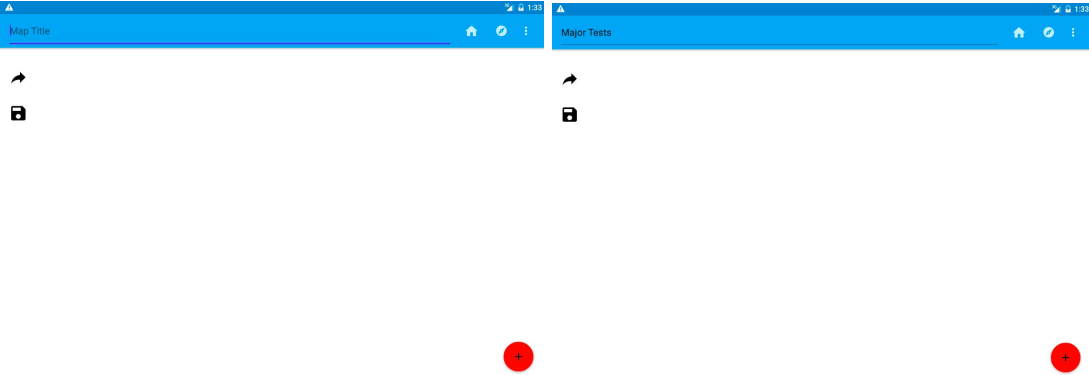
**Interface:** Create nodes of different types and connect them by dragging between different nodes. Edit the content of nodes to add more details. Move your nodes around as you desire. Delete useless nodes, and use nodes as “flashcards”, with extra content appearing when you double click on a node.

This complex task is the heart of Covalence -- we had to make it easy and intuitive to create and connect nodes. We spent a lot of our design time iterating on gestures, described below at each mini task flow.

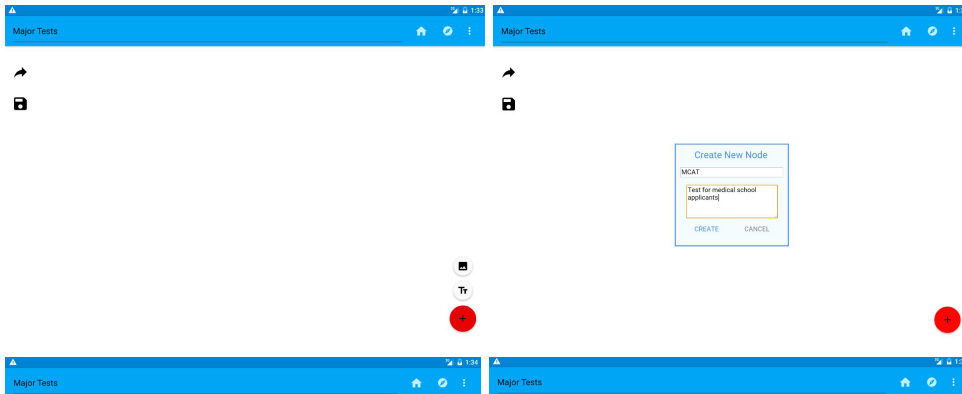
Home screen: My Maps



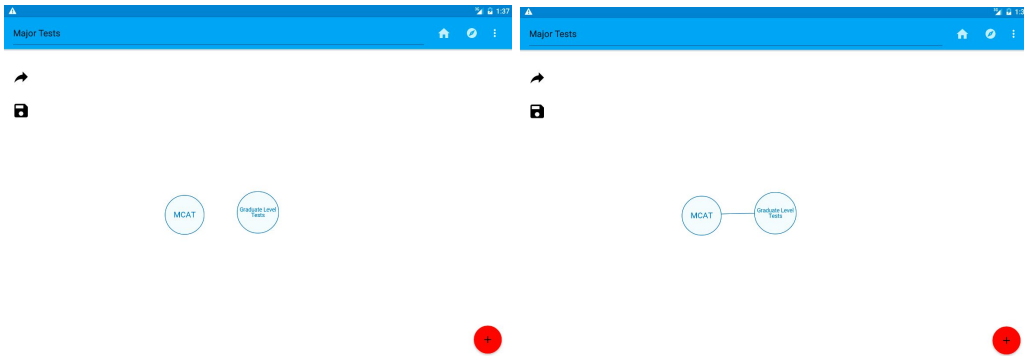
Creating and titling a new map



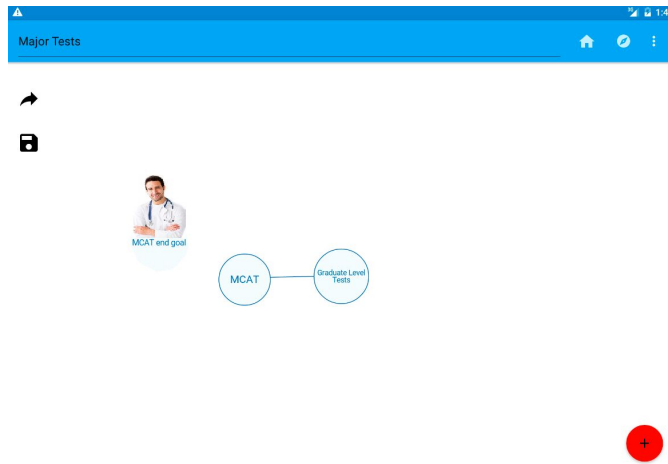
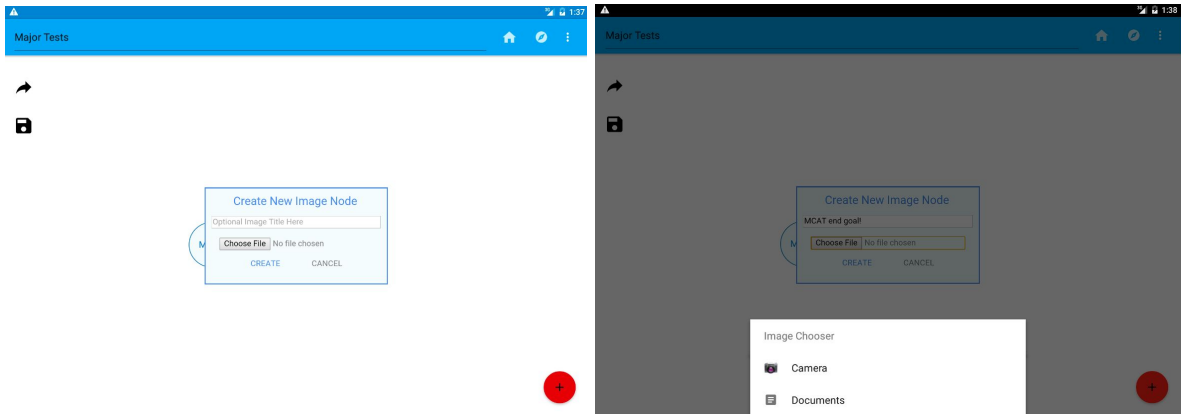
*Creating and moving a text node  
(Interaction: hold and drag to move a node)*



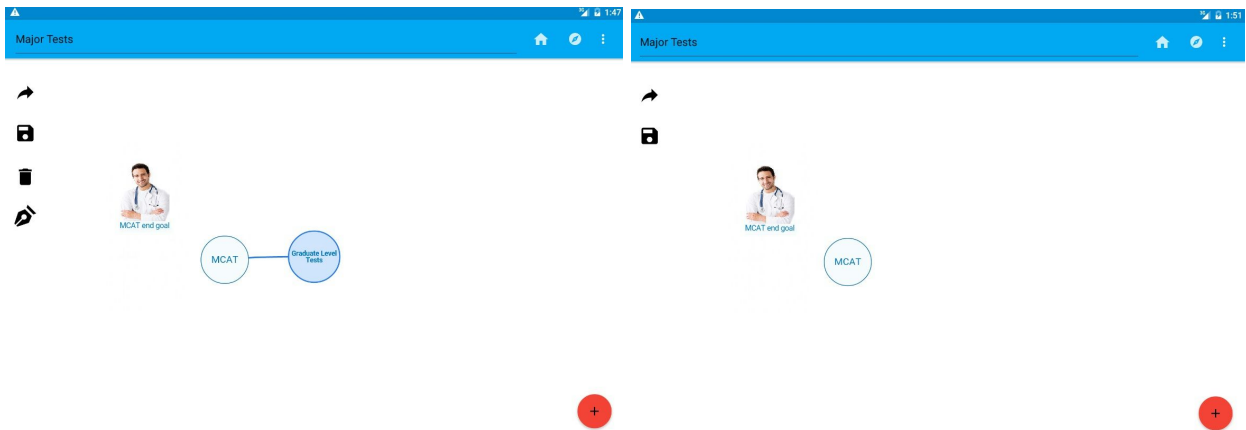
*Creating and connecting a second text node  
(Interaction: drag between the centers of two nodes to connect them)*



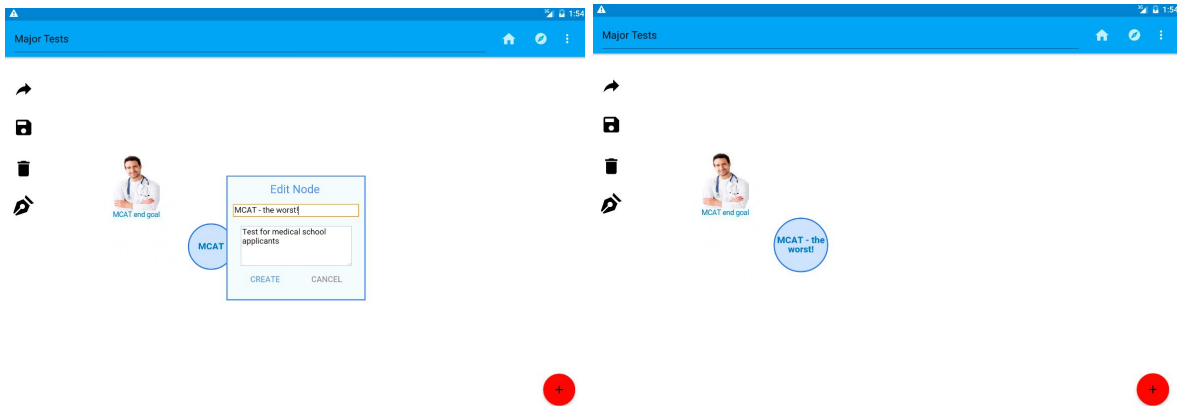
## Creating an image node



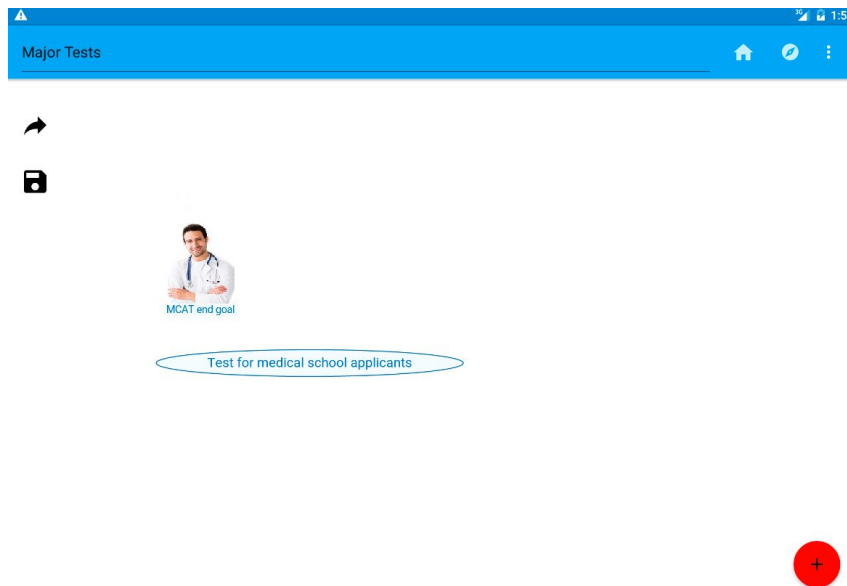
## Select and delete a node



## Editing the content of a node



View content on the other side of a node  
(Interaction: double tap on a node to 'flip it over' and view the content)

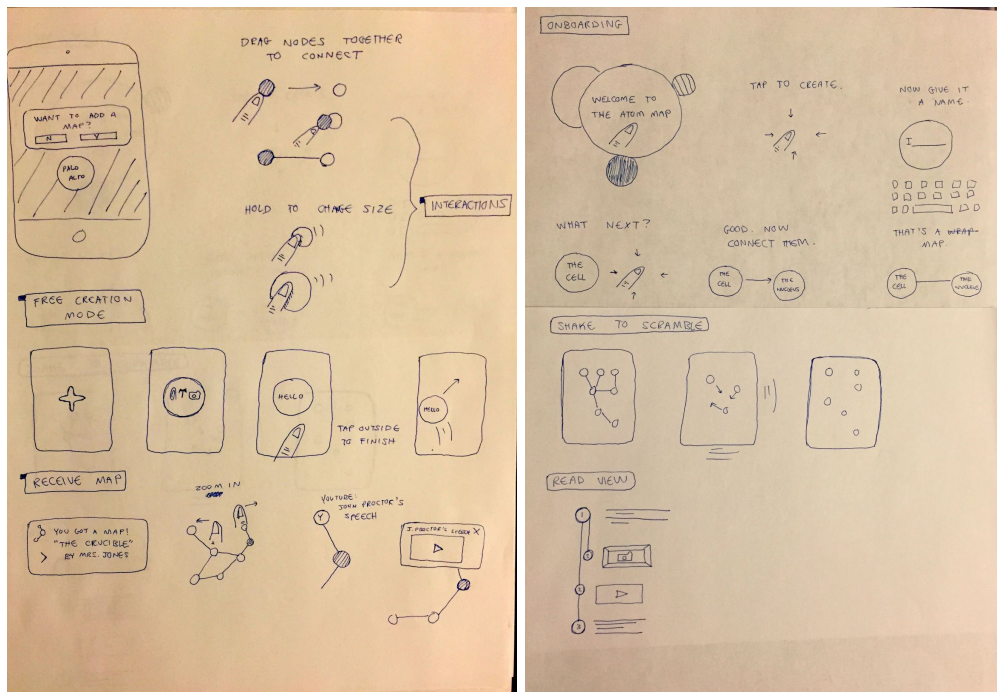


# Design Evolution

## (1) Initial Sketching

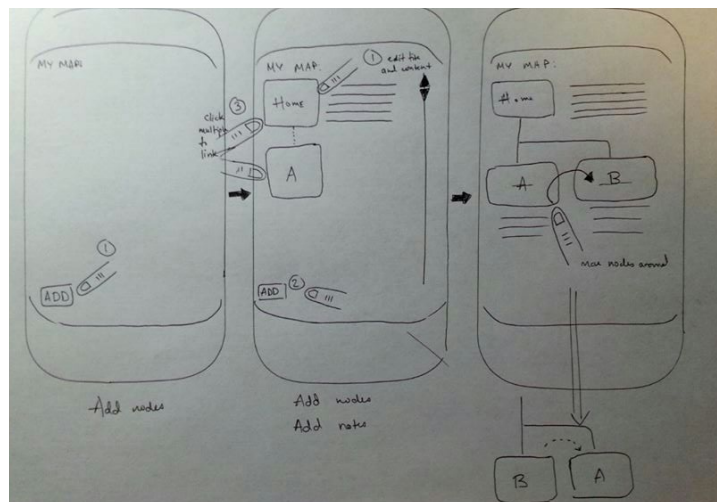
In our initial sketches, we tried to explore different gestures and a few different ways that people might view their concept maps. We decided ultimately that designing for a tablet or large phone would be best for the use case we intended -- watches were too small to effectively explore concept maps, and virtual reality didn't fit well into the education or business use cases we had envisioned.



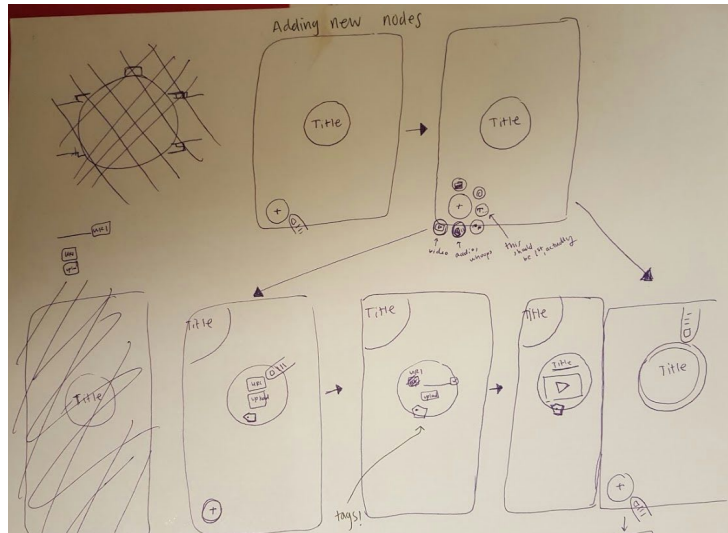


Our initial sketches came down to a decision between “read view” and “node view,” as you can see below -- read view was optimized for students studying and looking at detailed connections between discrete nodes, while node view was optimized for viewing a map overall and exploring connections on larger maps.

### Our initial sketches: read view



### Our initial sketches: node view



We decided to move forward with the “node view” for our interface design because it facilitated more intuitive creation than “read view.” It takes fewer taps to add and connect nodes, which is the fundamental interaction. In addition, “node view” seemed more tailored toward collaborative creation of maps. “Read View” felt very static - it would be difficult to rearrange and add new nodes and content (especially with multiple people) given the limitations of the structure of the map: it only has up/down scrolling and a lot of text. “Node View,” on the other hand, has a dynamic interface that allows users to scan around wherever they are interested - making changes is cleaner and easier in the larger space, and people can work on their own sub-categories freely and combine them quickly. Ultimately, we prioritized the areas that our users valued the most: flexibility, speed, and collaboration.

## (2) Paper Prototyping

The first task of our low fidelity paper prototype revolves around the creation of a concept map using the “Create” tab -- the core functionality of Covalence. We really wanted the creation process to be fun and intuitive, so this was a critical flow to test. The default “Create” tab is almost blank save for a large plus figure in the lower right corner (*Fig. 1*). Tapping this plus figure brings up another menu with tabs for creating text, image, or hyperlink based nodes. Tapping one of these buttons prompts a sticky note (representing a node) to be placed in the “Create” tab. The user can position this note however they choose and write information on the note.

This task showed us a few key UI elements we needed to change:

- Participants didn’t understand the “link” option coming off of the plus button. They thought that “link” was for connecting nodes rather than for entering hypertext. We later decided to remove this option for our MVP, because people were more interested in

inputting text and images than in linking to webpages. Webpages did not facilitate discovery and learning within our app.

- Participants found the gestures difficult to discover -- the gestures were intuitive once introduced by the facilitator, but they weren't obvious just from the interface itself. They especially struggled with the flow we had devised for connecting nodes. In our hi-fi prototype (described in more detail below), we changed the interactions for creating and connecting nodes to be more intuitive.



*Fig 1: Paper prototype flow for creating nodes.*

In the second task, the “Explore” tab is used to simulate searching for and examining a map made by another user. We wanted to play with generating a community feel, and see how interested users were in exploring other people’s thought process. Tapping the “Explore” tab in the bar at the top of the interface brings the user to the “Explore” screen which features a search bar and suggested topics (*Fig. 2*). For the purposes of our prototype we assume that the user searches for the keyword “Biology”. This brings up a screen displaying relevant maps and their community based ratings. Selecting one of these maps brings the user to an interface displaying the map. At this point subjects were asked what kind of interactions they expected to be able to perform with the displayed map.

While walking through our ‘explore’ task flow, participants seemed to like and understand the upvoting and downvoting system. However, they wanted to be able to mark and save their favorite maps, and add to saved maps themselves. We made this workflow more explicit in our medium-fi prototype by adding a ‘star’ to help users save maps.

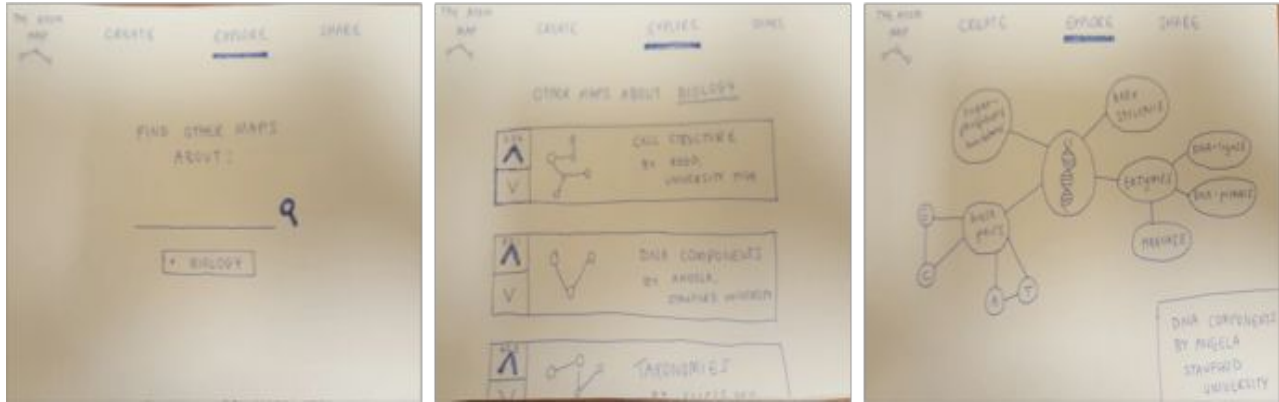


Fig 2: Paper prototype flow for exploring maps.

The final task of our paper prototype was to utilize the “Share” tab to share a map with another user. We wanted to see if users would gravitate toward the “Share” tab by making it so visible, since sharing and collaboration are key in having a good brainstorm or conversation about a concept map. Tapping the “Share” tab at the top of the interface generates a display containing a list of user created and recently viewed maps. This display also contains editable fields on the right that allow the user to specify whom to share a map with (Fig. 3). Tapping a map in the list highlights it, indicating selection. Tapping the same map again brings up a menu allowing a user to confirm the sharing of the map.

This share tab design was successful in highlighting the importance of sharing for users, but didn’t fit as well into the creation workflow as we had hoped. Users had to leave the map they had created in order to share it -- it felt like they had to be “done” before they could add other collaborators. In our medium-fi prototype, we converted the sharing function to a button on each individual map, so that other collaborators can be added at any time.

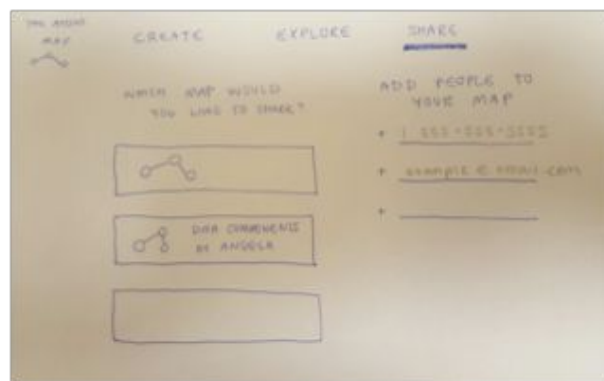


Fig 3: Paper prototype flow for sharing maps.

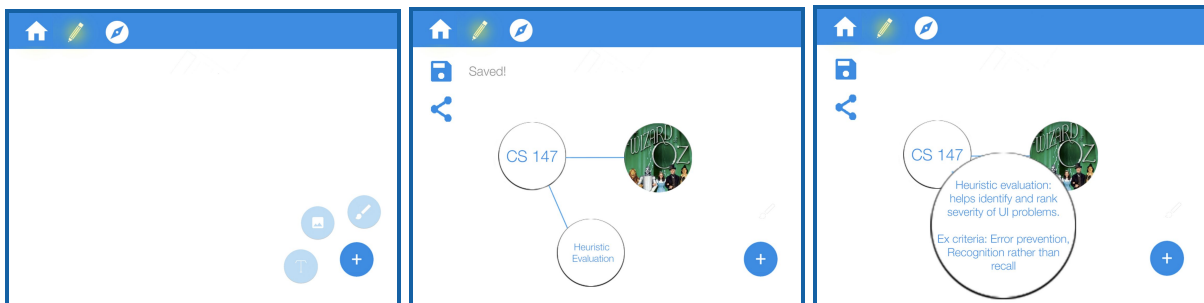
### (3) Med-Fi Prototyping

In our medium-fi prototype, we addressed some of the changes mentioned earlier:

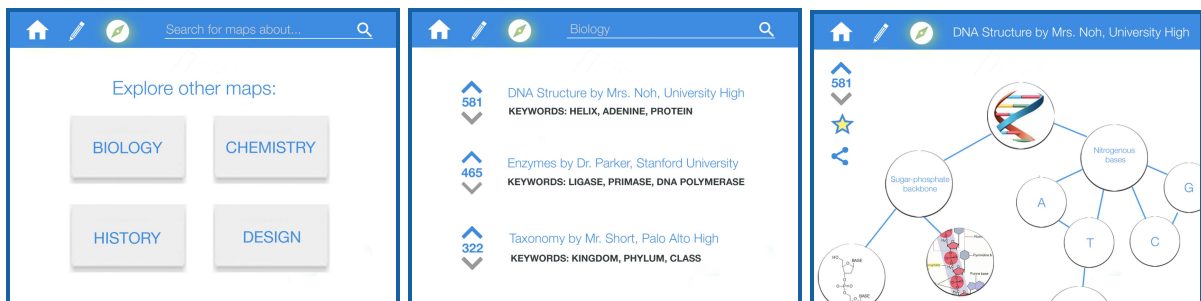
- We got rid of the hyperlink option for a node type, and replaced it with a 'drawing' node type, since participants in our paper prototyping session wanted to be able to see their own handwriting within the map.
- We moved "share" to a button on the map interface rather than its own tab, to better integrate sharing into the map creation process.
- We added a 'star' icon to the map interface when viewing community maps, to cue users to favorite / save the maps they viewed.

We tested our medium-fi prototype with a heuristic evaluation from one of our peer groups in studio, which we'll discuss in more detail below.

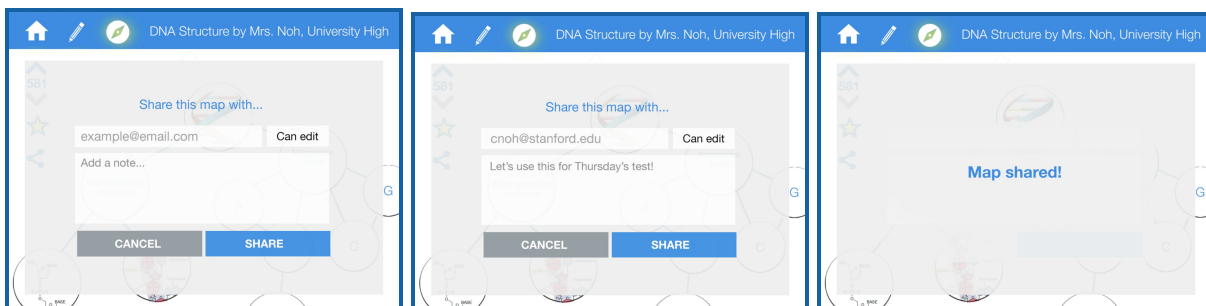
#### *Creating a map and viewing additional content*



#### *Exploring community maps*



#### *Sharing maps with others*



# Major Usability Problems Addressed

We addressed each of the severity 3 and 4 heuristic violations, and then added a few other changes to improve the user experience.

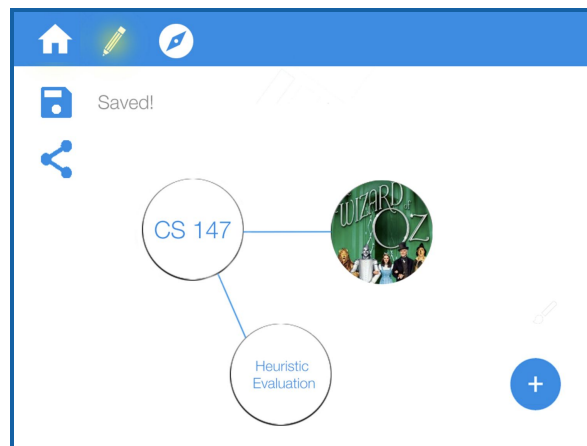
From our peer heuristic evaluation:

## [H2-1 Visibility of Status] (Severity 3)

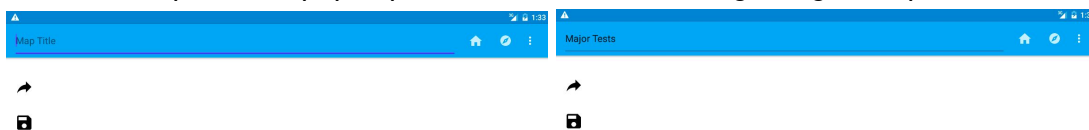
When saving a map, the only user-feedback is a “saved!” message. I did not name my map, so what was it saved as? Provide a popup “create map” form when users click the “create” tab, or when they press the save button.

This feedback helped us realize that we had not, in fact, allowed users to title maps anywhere in the interface. We added a “Map Title” bar to the top of each newly created map, which has to be filled out before map creation can begin.

*Before: No way to title a map -- not obvious what the map is saved as.*



*After: Map title bar pops up and is filled out at the beginning of map creation.*



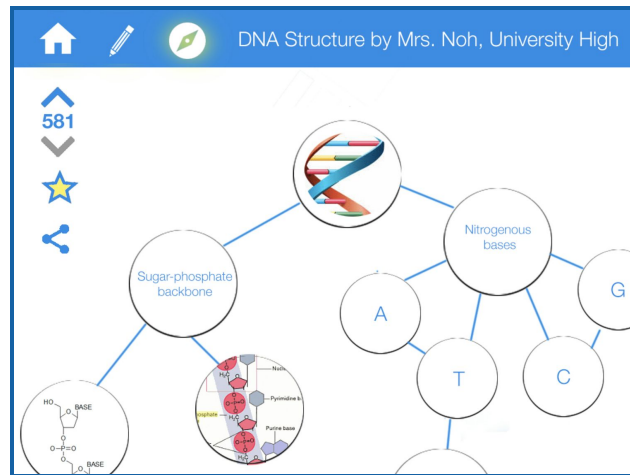


### [H2-3: User Control] (Severity 3)

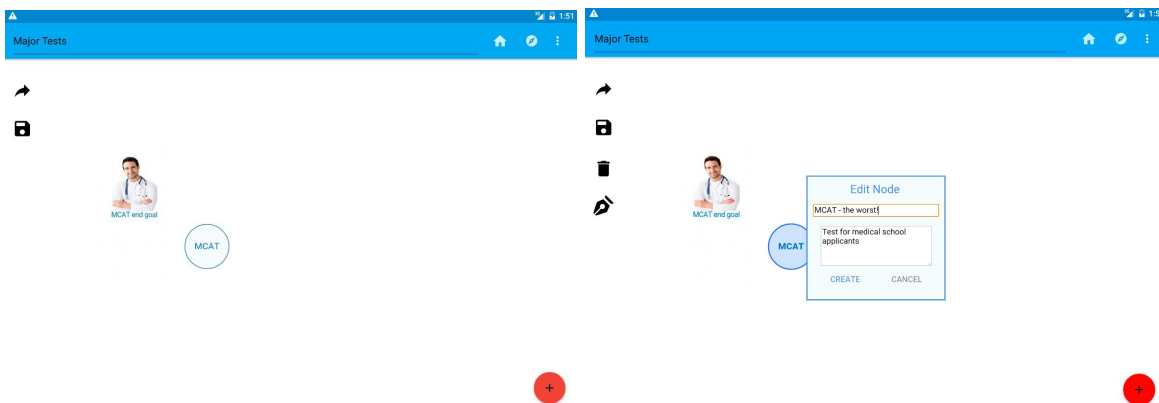
How do you edit an existing map? The editor only allows you to create new maps - I don't see any edit button when I'm viewing a finished map - this seems like an important oversight especially since editing/creating maps is one of your main tasks. There should at least be an edit button somewhere in your prototype to represent this.

Similar to the issue described above, there was no clear cue for the user that they could edit the map they were viewing. We added explicit 'edit' and 'delete' buttons that are visible when viewing maps and selecting particular nodes or edges.

*Before: No clear way to edit a map you're viewing.*



*After: 'edit' button and 'delete' button pop up when node is selected*

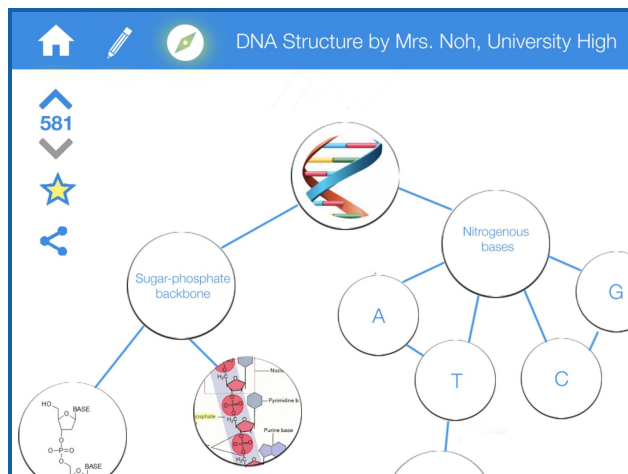


### [H2-3: User Control] (Severity 3)

There needs to be a save button when you view other people's maps, too. How do you import/save a public map into your local storage so that you can edit it, or see it later? (I see a "saved maps" section in the "Home" section, but there's no corresponding save button when you look at a specific map under the "Explore maps" section)

We decided not to implement saving someone else's map for the hi-fi prototype -- there is no 'community' in our hi-fi prototype, so it's not possible to save other maps. We decided that for the sake of the exploration task, it would be sufficient to view and click around in other maps without saving your own local copy.

*What our evaluators were referring to: no clear way to save a local copy of this community map.*



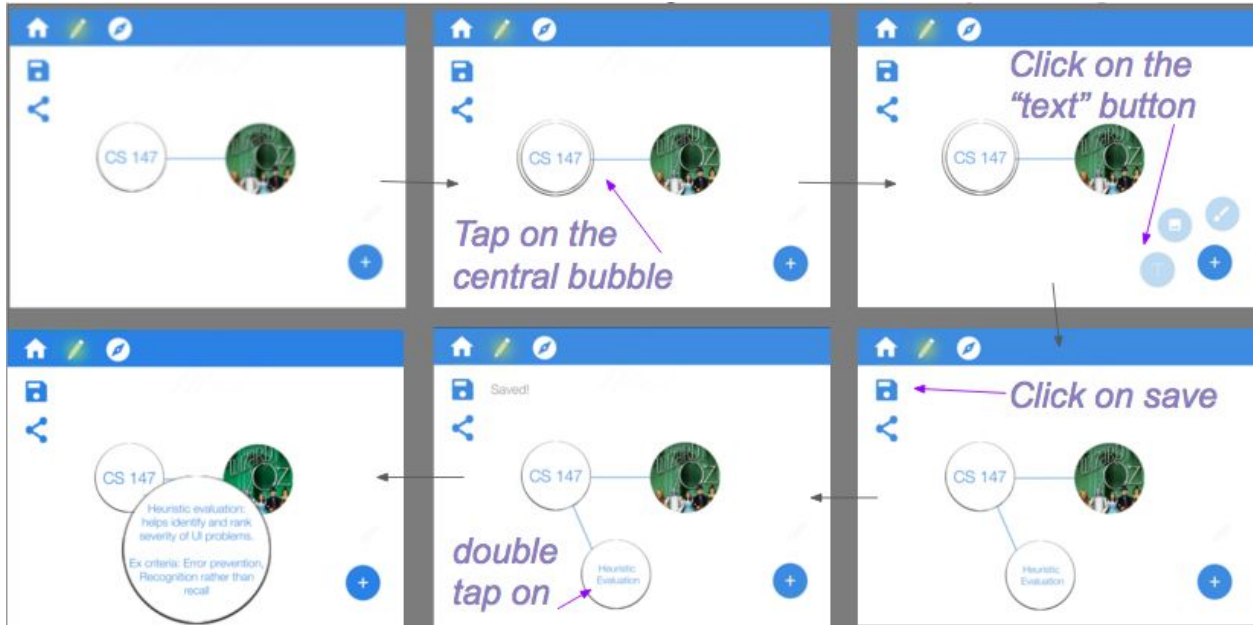
### [H2-7: Efficiency of Use] (Severity 3)

Two taps to add a node (first tap “+” and then tap the node type) seems like too many. Why not let the user single-tap on blank space to create a node, and give it a default type? This would make it one tap to create a node.

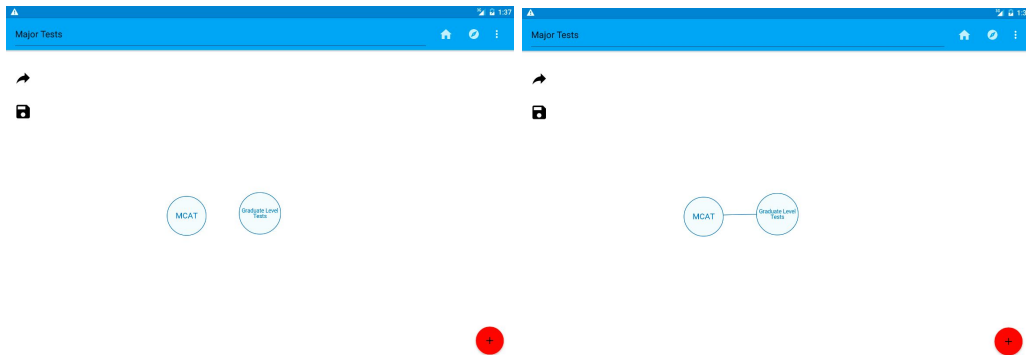
Although you still have to click “+” and then select the node type in our final prototype, because this flow is a classic and intuitive feature in apps with this floating action button, this was a critical piece of our application revision with regards to our workflow for connecting nodes previously, which had too many steps and was not intuitive. Each added node had to be connected first, which didn't necessarily fit with the user mindset of having unconnected ideas, and then drawing connections. We resolved this by having users add each node individually, and then draw between the centers of two nodes to create connections. These gestures were much more intuitive and delightful. We also decided against giving a node a default type, because it would add many more user steps down the line for users who wished to change the content type of the node.

*Before: Tap on a node that you want your node to be connected to, then hit 'plus' to create a node that is automatically connected to the selected node.*





After: drag between the centers of two nodes to connect them.

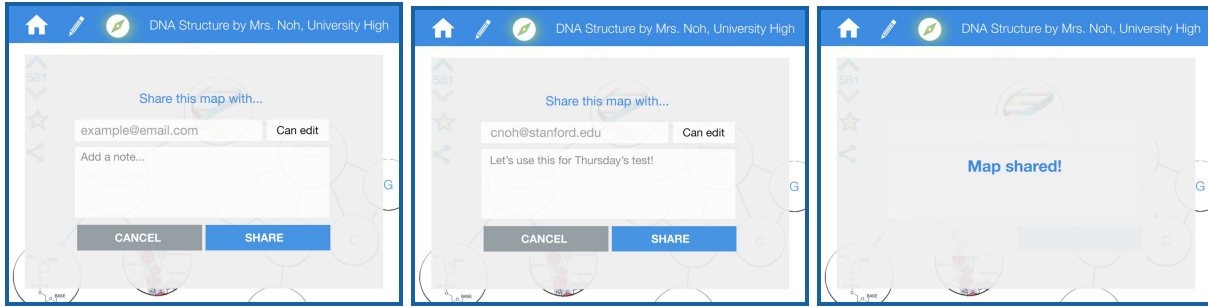


#### [H2-10: Documentation] (Severity 4)

The sharing screen should indicate how it will work. What if the person you want to share a map with doesn't have the app (which is a high probability, since you're sharing via email)? Do you send them a PNG? Do you ask them to download the app? If they have the app, does the map get saved to their account with or without their permission? The screen should inform the user how it works, since that may affect whether or not they want to share something, or whether they'll have to tell the share to do something in order to access the content, etc.

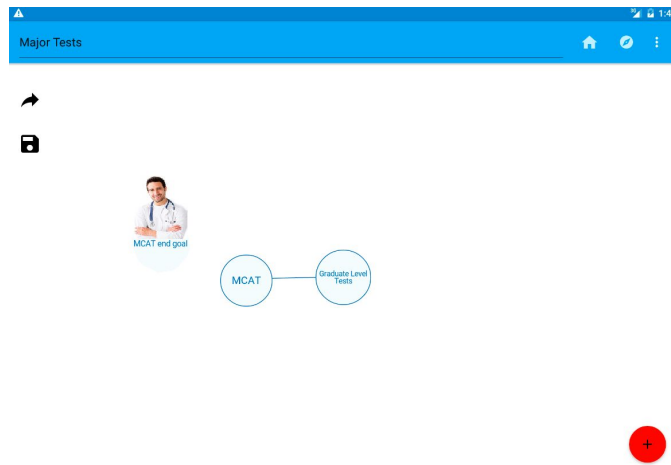
Since this is a common convention in sharing dialogues, such as those on Google Docs, we decided to leave the sharing dialogue relatively the same. In implementation, if the invited user had the application, they would get a notification inside the application. If the invited user did not have the application, they would get an email inviting them to join Covalence.

*Before: sharing maps with others*

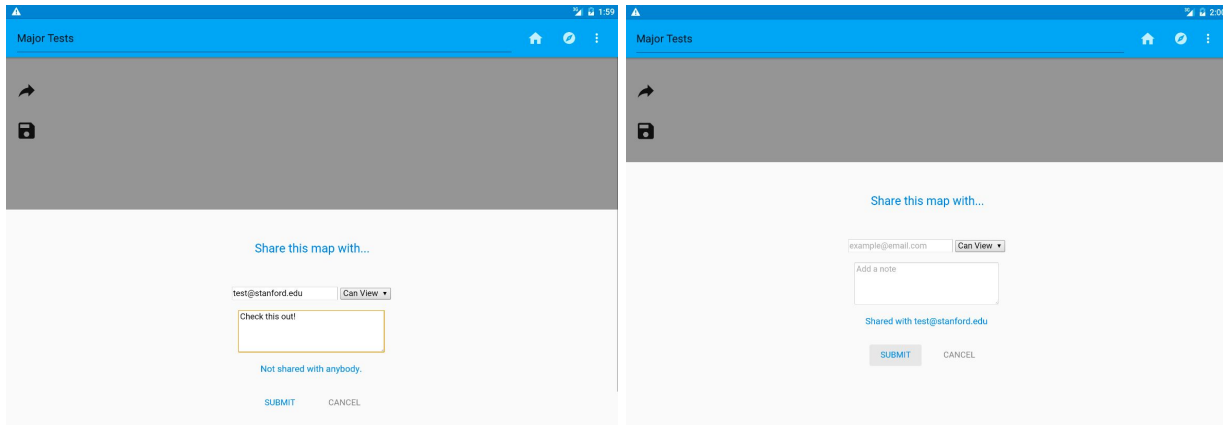


*After: sharing maps with others*

*Click 'share' button in the top left*



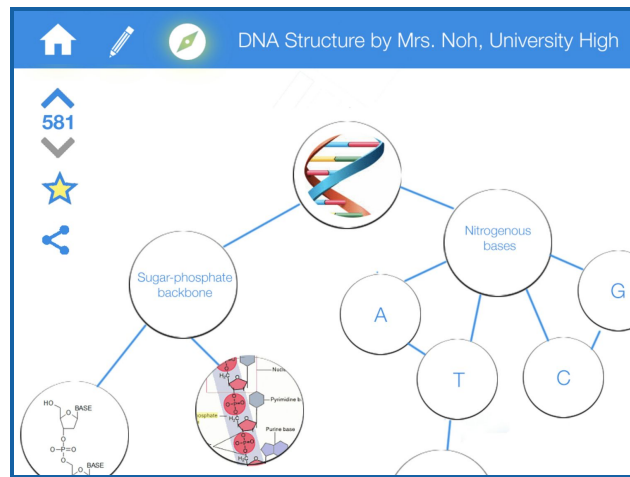
*Fill out share pop-up dialogue*



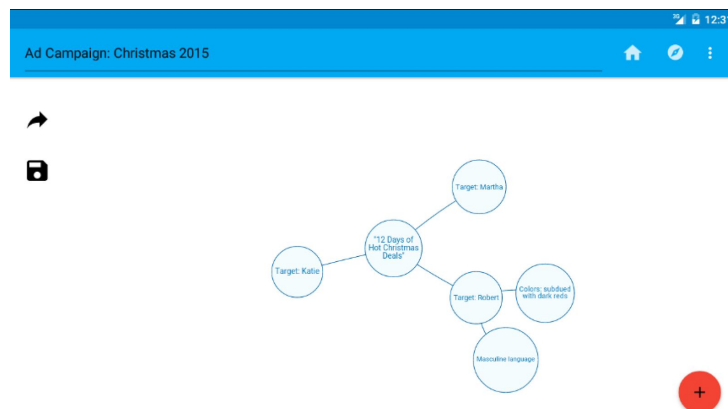
We also made a visual change in light of the heuristic evaluation that did not tie to specific level 3 or 4 critiques:

We created a more aesthetic map look by adding color to the node outline and background, helping the map stand out more against the white background.

*Before: White nodes with black outlines on a white background felt plain.*



*After: Blue nodes with a blue outline helped each node stand out, and made the map come alive on the canvas.*



## Prototype Implementation

### Tools

Covalence was built for Android using Android Studio. We optimized the app for the Lollipop mobile OS, because there was an existing bug in Marshmallow with the SearchActivity. We wanted to use a robust map-making library, but none existed for Android, and it is beyond the scope of this project to create that. We found a library called vis.js which was suitable to our visualization needs, so we built a web app to display the concept maps using the graph concepts of nodes and edges from the library.

The web app was created using Angular.js, which on further reflection was a touch overkill for what we needed the app to do, especially because there was no back end and the toolbar was

in the Android app. However, Angular.js is nice because it's very responsive and asynchronously grabs data and refreshes, and would be useful if we continued forth with building out the database for the project. We displayed this web app using a WebView in Android and passed the data back and forth through JSON. WebViews are never optimal for Android, but it was necessary to use because of vis.js and our lack of other available resources to create a 'concept map' look.

We chose Android because programming an Android app doesn't have too high of a learning curve -- it is based on Java, and CS 106A knowledge is sufficient to learn how to work with Android. This was important because only one team member had any prior mobile experience. We saved maps locally, because setting up a database would have been too difficult and time consuming, and because of this, we were unable to robustly implement users. Android was also useful because a lot of our earlier design was based on Material Design standards, and Android has most of the Material Design built into the system, unlike iOS. However, there were difficulties in changing the standard Material Design look when we wanted a different sort of style; we had to wrestle with changing the default conventionality to add some extra pops of delight and innovative design.

### **Wizard of Oz and Hard Coding**

We hard-coded a small set of community maps to simulate the search and discovery experience for users, since we don't currently have community content available. We also hard-coded users themselves, with a default user named "Ted" or "Logan" for the current user. For our sharing function, we used a Wizard of Oz technique and told users that their map had been shared. At this point in the implementation, Covalence does not actually send emails to the email addresses written in the dialogue.

### **The Future of Covalence**

Missing from our current implementation:

- As mentioned above, we don't actually share links to download Covalence to email -- in the future, we'd need to have this system setup to automatically send emails and push notifications where relevant.
- For our hi-fi prototype, we chose not to implement the community up and down-voting system that appeared in our medium-fi prototype. This would help build the Covalence community in the future.
- We're missing a database to store user and map information and build a community. This wasn't necessary for our hi-fi prototype, but it's a likely next step if we continued the project into CS 194H.
- We decided not to build copying a map or favoriting a map for this hi-fi prototype, but those are critical actions to allow collaboration in future iterations.

Additional features and fixes:

- Login: for the hi-fi prototype, it wasn't necessary to create a full login flow. In the future, we want users to login with their gmail credentials.

- For our hi-fi prototype, we only implemented text and image nodes. In the future, we'd like to add other dynamic content types: web links, files (like PDFs), hand drawing with a stylus, etc.
- We want to encourage people on Covalence to collaborate and build off of each other's work -- one element of this would be automatic crediting, so that any map made by multiple people would credit all of its creators with a small text line at the bottom (similar to how Medium currently credits all editors of a published piece).
- We'd like to add a way to easily view recent searches and recent maps -- if Covalence is meant to be used as a study tool, we want it to be easy for users to pick up where they left off.
- Again on search, right now the only things users see in a search function is the title of the map and its author. In the future, we want to make more detailed and informative search cards that give a small preview of the map's contents with a single tap.
- Our users frequently requested more visual options -- for example, more node sizes, and more node and line colors. In the future, this could help make Covalence maps more aesthetic and fun to explore.
- We need to add real-time collaboration and editing, so that two people in two different locations could edit the map simultaneously, as we originally depicted in our concept video.

## Summary

Building Covalence was a great opportunity for us across many dimensions. From a design perspective, we got to explore three key tasks that would be critical to seeing our app come alive in the classroom. Our focus on these three tasks helped us get to a higher fidelity core experience, while avoiding feature creep or focusing on unnecessary detail. Throughout the project, we were able to refer back to our low-fi prototype, where students from Bellarmine High got excited about brainstorming the narrative of the Crucible. The insights that grounded our project through the needfinding process helped us stay grounded about our goals: flexibility, ease of use, delight. Technically, this project gave our group members an opportunity to explore Android development, and play with the data visualization library that ultimately gave Covalence its cheerful look with the movement of nodes and connections on each map.

Thanks, CS 147!

--Ted, Logan, Emma, and Courtney