**Sharknado**

**Introduction:**

Our group decided to study fitness instructors as the users for this project. Our instructors consisted of: two pole instructors, a jazzercise instructor, an aerobics instructor, and a swim instructor. The major tasks that our users performed were planning for a class and executing that plan during class. We found that our users differ in how strictly they plan lessons. Some of the users took a more organic approach while others used a strict and rigid approach. Because every user’s execution phase was so varied we decided to only work on the planning phase for our users. From the planning phase we found two major breakdowns: storing content and recollecting content. These breakdowns made it difficult for the fitness instructors to find exercises they had previously saved which made the planning process long and tedious.

We addressed these areas by creating an application that centralized where the data was stored and how it can be sorted. We also streamlined how planning is done by having all the data centralized. Our application is composed of two sides. One is called Storage Locker which is where all the content is stored. Here users can save and edit content they find from third party sources and tag the data. The second part is called Plans. This is where the users can add content from the storage locker to create an exercise plan. We used usability testing and think aloud evaluations as techniques during the prototyping phase of the project.

We used several evaluation techniques to determine the quality of the application we created. The two techniques we used during our evaluation period were Wizard of Oz and Think Aloud. During the beginning of our evaluation phase, we took two of our most promising low fidelity prototypes from the previous phase and performed the Wizard of Oz demo with one of our users. At the same time, the user also performed a think aloud, explaining exactly what she was doing and thinking to the person performing the evaluation. The initial goal of these evaluations was to really help us narrow down the features we wanted to take from both designs, as well as what features we still needed to create. Following these evaluations, we had two experts then perform the same evaluation on our high fidelity prototype.

These two stages of evaluation provided many useful pieces of information that helped us create our final high fidelity prototype. The first issue we found had to do with our own biases: we had inadvertently designed our app for a specific subset of our users, rather than for all of them. Acknowledging this helped us modify our designs to create a high fidelity prototype that appealed to all of our users, instead of just that specific subset. The other major issue we found during the evaluation stage was that we were wrong about what interactions our users would assume were implemented. Our first designs both used press and hold interactions with pictures and videos, which none of our users discovered on their own. After that discovery, we were able to exchange those interactions for ones more easily understood by our users.

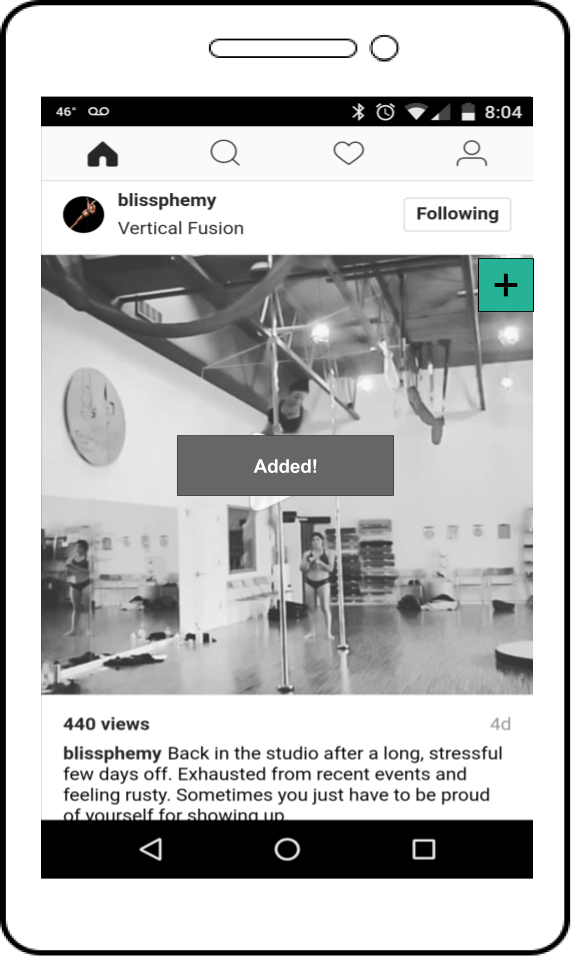
**Evaluation Plan and Discussion:**

For the evaluation plan, we decided to use quick and dirty, usability, and analytic evaluation techniques. We combined think aloud and usability testing with two of our end users. We came up with two sets of tasks to ask each user to do on two different prototypes, prototype A and prototype B. We had both our users do the same tasks on both prototypes. Each initial prototype had similar sorting functionality, but they differed in the way they created plans. Prototype B used a list format that had a note taking feature. Prototype A, on the other hand, used a video/sequence format with a music feature. During the evaluations, the first task was to create a new plan, add three items to it, and then delete one item. The second task was to edit an existing plan by adding an item with a specific tag before removing a previously saved item. The evaluations only took about 10 minutes each. Each evaluation went smoothly because the user gave us input throughout the entire evaluation. However there were a few things that could have been better during this phase of the project. After selecting icons for the project that the group thought were intuitive we found during the evaluation that they were not intuitive to the users. Another thing that could have gone better was the AB testing we did. The tasks we used we specified to prototype B more than prototype A. Also, we found it difficult to model certain behaviors like dragging on a paper model and that confused some of our users.

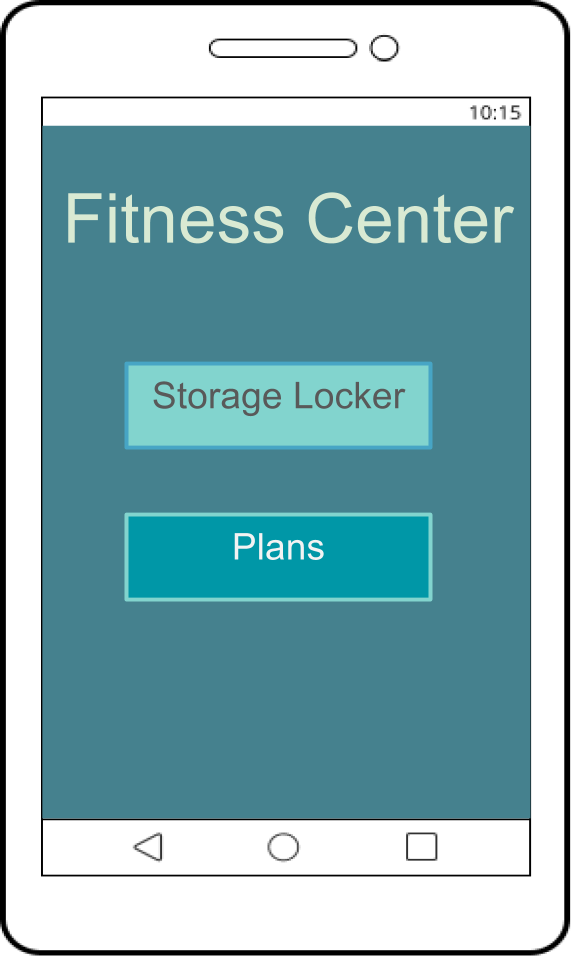
During the second evaluations that we did with our experts, we used the same techniques but performed them on the final high fidelity prototype. We had the experts create a new plan, edit an existing plan, and search for a specific image or video. These evaluations took roughly ten minutes to complete. There was nothing major that needed changing after these evaluations, as the changes we had made due to our previous evaluations proved adequate. Overall, these final two evaluations went very well and gave us great feedback from the perspective of other developers and designers.

**Evaluation Results:**

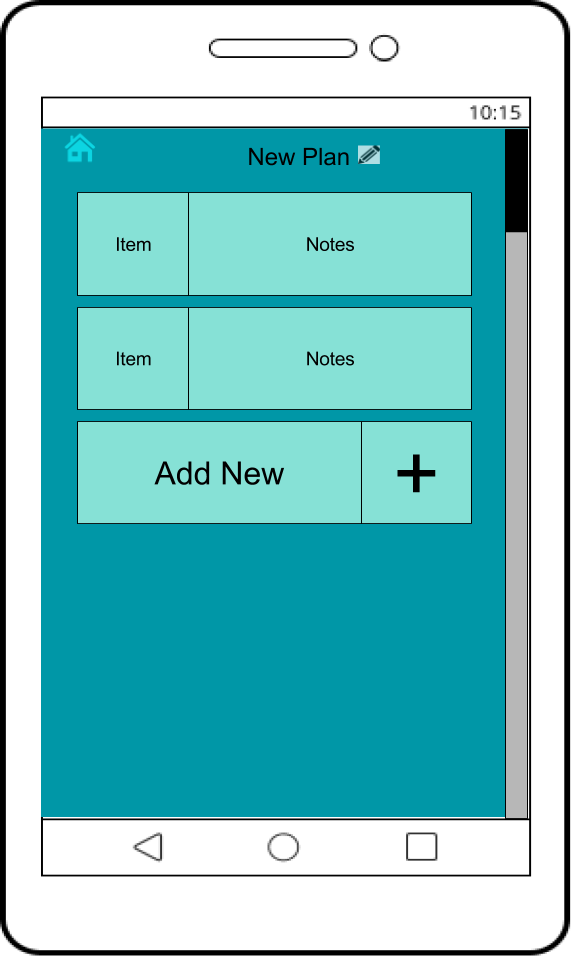
During the evaluations, we learned what functionality our end users preferred, and what functionality they didn’t. For instance, prototype A had a music feature that prototype B did not. After having our users evaluate both prototypes, we discovered that most of our users did not want and would not use that feature, since they already had solutions which they liked that performed the same function. We also found that our users really liked the note-taking feature from prototype B that wasn’t in prototype A. Another feature that we ended up removing was the use of interactions that, while supported by the operating system, were not visibly telegraphed to the end user, such as press-and-hold interactions. Our users ran into difficulty discovering these interactions, and we realised very quickly that we needed explicit objects for our users to interact with. After initial testing with our end users, we decided to consolidate the prototypes into one by cutting out the music feature and keep the note function to continue supporting some of our users’ natural behavior.

**Final Design:**

For the final design, we really wanted it to be as intuitive as possible. To that end, we created an application called Fitness Plan that users would be able to use to obtain content from other applications using basic mobile operating system sharing functions, manage their content, and create plans from that content.

The user will interact with the default sharing features of the app they’re getting content from to share that content with Fitness Plan. During this process, the user will add a title and content description tags to aid in categorization. This aspect of our design is only used when a user finds an exercise they wish to store.

The Fitness Plan app was designed to meet the end user’s need for sorting and planning. Here we broke down the application into two parts: the Storage Locker and Plans portion. The Storage locker is where users will find all the content they’ve stored from other services. The plans portion of the app was created to allow some flexibility in planning styles. This allows for the most diverse amount of users being able to use this app.

The Storage locker is the centralized place where all the saved content can be viewed. Here you can search through all of the content in multiple ways. You can sort the exercises currently on screen by date added (new and old), relevance, popularity, and alphabetically (both forwards and backwards). The app also features a search bar that allows users to search for specific tags or titles. When users are using the storage locker they can select a specific exercise to get more information about it. This will generate a pop up with the selected item’s information as well as what actions can be done with this item. One action is to create a new plan using this item as the first exercise. Another is to edit the item they are looking at so they can change the title, add more tags, or remove it from the locker. Finally, a user can add the item to an active plan which will take the item and add it to an existing plan of the user’s choice.

The plans part of the application was created to streamline how the users planned their lessons. The goal of plans is to have all of the content accessible so they can pick and choose which items they want to use. The Plans process is subdivided into two parts: create a new plan or view saved plans. Creating new plans gives the users a blank template for them to add content to and build a new exercise routine. Here users can add an item from the storage locker and then add notes to better describe exactly what kind of exercise is being added. In saved plans a user can edit their pre-existing plans, add more content to them, and delete them if they are no longer being used.

Overall, our design is realistic in the way it looks but not very realistic in it’s functionality. We used Google Slides to create our prototype, and we believe that held us back from creating the exact functionality we wanted in our prototype.

**Future Work:**

At this point, the application we designed is a baseline for all of our different users to use as a standard when planning. It was a challenge for us to create something that could work for all of our users because of the wide variety of users we started with. Further work on this application would be customizing it to fit the individual needs of the different types of users. One example of extending the application for a specific users is the jazzercist. The way the jazzercist acquires content is much more structured and far different than the process used by all of our other users. It would be great if we could take the DVD’s they receive and translate them to the digital content they use for the lessons. All of our users had different needs and it was hard to create a application that could work for all of them.

In the future, the prototype needs to be greatly improved. The current way of using slides was a major limiting factor in what we could show our users. It would have been better to use a tool that would have allowed us to mock the real behavior of the system.

The design aspect of this application was challenging, but the significant engineering challenges that are required to make this system a reality also need to be addressed. One of the main challenges is getting access to the third party sources and being able to use the content. This is not only an engineering challenge but it could also be a legal challenge. Another challenge is creating a better way to store the data so as to avoid needing users to save it in multiple locations. If the data already exists in one location, it would be redundant to copy it and save it to another location. This application would need a way of pointing to the original source in order to save space. These are just a few of the challenges that would need to be addressed if we were to continue working on this project.