4.2 DESIGN EXPLORATION

4.2.1 Design Decisions

Our team was given the task of implementing an existing design into both an iOS and Android application. To do this, our team made decisions to improve the existing design to better fit the client's and user's needs.

One design we made was to change what data result information the user can see. The current design displays the lift just performed as a line graph, previous lift improvement as a bar graph, and a bunch of other past data in a condensed, hard-to-read format. We will change the previous lift improvement graph to a line graph of the past month's lift readings with the option to change to a time frame. We also intend to lay out the other past data stats in a more readable layout. This will allow users who are not as familiar with weight conditioning metrics to still understand their data and be able to reach their goals without always talking with a coach.

Another design change we made was to change the app colors to be friendly to colorblind users. The previous iteration used red and green to indicate left and right force statistics, however, these can be hard to differentiate. This will allow users with sight imparities to have full use of all features of the app.

The most important design change we made was to have the app follow the orientation of the mobile device. The previous design had statically oriented screens. With this feature, users will not have to repeatedly orient their devices when navigating the app.

The last design change we made was to change the icons used in the navigation bar. The previous design did not use common symbols for settings and other features. Using icons the user expects to be associated with certain actions will improve app learning time.

4.2.2 Ideation

We identified the potential options for our graph display using a round-robin ideation technique as a group during a meeting with our client. We each listed off ideas for displaying user lift data and compared them to the existing design's solution for it.

Some options our team has considered for Graphs:

- Exploring enlarging the current bar chart possibly on selecting it on the screen, showing more detail on user history
- Adding labels to the graphs' axes to make the graph data clearer and easier to interpret
- Including interactivity to the graphs such as selecting data points to show more detailed data values
- Choosing colors that support color-blind accessibility when displaying different data metrics and values on graphs

- Making the graph dynamic so it zooms out if data exceeds its originally displayed limits, ensuring all data is visible to the user

4.2.3 Decision-Making and Trade-Off

Our team used a matrix to identify the pros and cons of both the past design and our new design. The matrix allowed us to understand what was currently going well for the initial design and where improvement was needed. Below are our lists of pros and cons of the two designs.

	Previous TrueForce Design	Our TrueForce Design
Pros	● Accurately measure user's force input fast ● Good graph visualization ● Logical screen navigation	● IOS and Android implementation ● Easier navigation ● Better security ● More capabilities to see strength in athlete ● More accessible to visual problems ● Screen size adjustment ● Easier to create users ● Can save quick start data
Cons	● Only works on specific tablet ● No security on the database or application ● No working back button ● User can't create their own account ● Can't save quick start data	● Will take time to create and debug, especially since the product is already out and we are developing two apps. ● Unable to see the backend, and takes time to connect with the backend team

4.3 PROPOSED DESIGN

4.3.1 Overview

The TrueForce Technology app is composed of 3 main systems: the mobile application, the data storage, and the weight rack (see Figure 4.3.1.1). The purpose of the weight rack is to measure the force exerted by a user and share the data with the application's interface. This communication is done through Bluetooth. The purpose of the application is to display current and past information to the user. The data stored from the application to the database includes player lift data, teams, athletes on specific teams, and different exercises. See figure 4.3.1.2

4.3.2 Detailed Design and Visual(s)

Our design is developed using the native languages of both Android and iOS, Java and Swift, so two different applications will be created for our solution. The application is responsible for connecting to the hardware rack through Bluetooth and authenticating a user, accounting for their permission levels. For a user, the application configures their lift, displays their live data, and makes API calls to the existing backend code for database operations such as saving new data or retrieving and displaying past data.

The hardware rack and the bankend are not being implemented or changed by our team. The backend and database were developed by a Romanian team and we do not have access to the code of either due to the contract our client has with them. At the moment, we still do not have information about what data is being stored or the API calls. The hardware rack was co-developed by our client and advisor. The rack consists of a metal frame and barbell that can be adjusted to different heights. The barbell is connected to isometric force sensors that measure the force exerted on both the left and right sides of the barbell.

4.3.3 Functionality

Our design would be intended to operate by a user on any personal device such as phones and tablets. A user, typically an athlete, would be able to enter a gym, turn on the existing hardware device connected to the weight rack, and from there, open their device. They will be able to launch our application, either from an iOS or Android device. Our design will allow the user the option to either log in or quickstart. Once logged in, or if they choose quickstart, the user can begin their lift. Once finished, a graph will be displayed with the results from their lift. If the user is logged in, a graph with the previous lifts will be displayed to see the user's history. A diagram with the navigation and visual look of the application is displayed in the Appendix. Some configuration options are available to users. If they are logged in, a user can select a specific exercise they will perform before beginning the lift.

In addition, a user who is a Coach and has Admin privileges will be able to set up configurations. A coach will be able to set up a "team" with multiple athletes so they can track each player. This is ideal for strength & conditioning coaches that might be looking at multiple athlete's improvement over the offseason. They will be able to select a specific team, exercise, and athlete before having the player begin the lift.

4.3.4 Areas of Concern and Development

Based on our current design the main area of concern is connecting the apps to the rack and database. We are using Bluetooth to connect to the rack. This is a concern for the group because, without a proper secure connection to the rack, the apps will not be functional. Connecting to the database is another concern for the group because we currently do not know anything about the database and the app will not be able to store athlete and coach data without it. Our immediate plan of action is to receive the database from the client so we are able to store data. We are also currently researching ways to securely connect to the rack through Bluetooth to keep personal information away from unwanted viewers.

4.4 TECHNOLOGY CONSIDERATIONS

One technology that we are looking at using is two-factor authentication for a more secure login process for our users. This would allow us to verify that the correct person is trying to access their data and also help prevent SQL injection. Whether we create this ourselves or use a company like Okta is still up for discussion. However, the latter would only require setup and would be more secure in the long run as it would require little to no upkeep.

Another distinct technology that we are using is API calls to the backend either to receive or to put data into the database. This is the best choice of action since the design of our design project does not give us access to the backend. However, we do have access to the API hooks to achieve our desired functionality as previously stated. This of course, has a drawback in that if the API hooks do not work we have no way to resolve the issue in the backend, thus if there is a bug we will have to wait for it to be fixed before continuing on.

The last distinct technology is security around the database. We are still in discussion about the best action plan, however, the two ways that we have researched lie in cloud computing database or simply using a company and setting our system/server to track interactions with our server and track the data as well. This is the best course of action in comparison to us trying to work to create the security ourselves, as one main issue that we are taking into consideration is information recovery if the system were to be tampered with in a malicious way. Data protection and recovery were some of the client's main priorities.

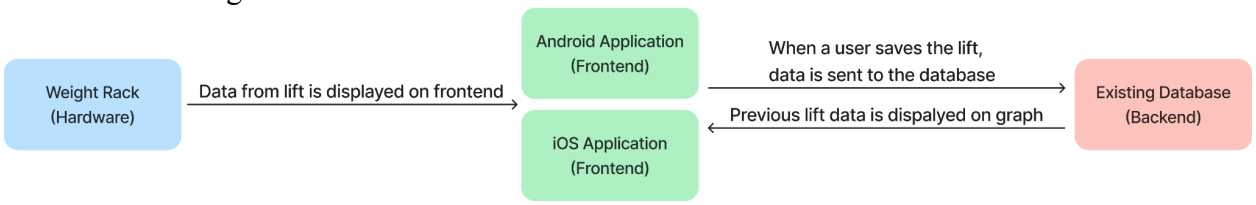
4.5 DESIGN ANALYSIS

So far, our team has built several screens and implemented navigation between them in both the iOS and Android teams. Most of the first half of the semester was focused on research and understanding our client’s needs. None of the team members were familiar with iOS development so there was a learning curve before beginning. Our proposed design has worked so far. Our team utilized the existing designs and added improvements. Most of the design has functioned as expected. Since the designs have been mapped in Figma, it has been easy to implement our design choices. Currently, we are stuck with connecting to the backend. Our client has the backend built out and has chosen the stick with the current database. However, this means our team has been lacking information on the design choices made for the backend and the connection points for our front-end designs. For future implementation, our team will need to connect to the database and work out the functionality for connecting to the hardware via Bluetooth. The design choices we have gone with seem achievable.

Appendix

4.3.1.1 Weight Rack

4.3.1.2 Block Diagram



4.3.3 Functionality Diagram