True Force Technologies Mobile App Development

Design Document

sdmay25-10
Client - Benjamin Durbin
Advisor - Matthew Post
Team Members/Roles

Tessa Lanzel - Team Product Leader / Frontend Development Co-Master

Molly Rooney - Team Communications Leader / Frontend Development Co-Master

May Edel - App Development Co-Master

Paige Schneider - App Development Co-Master

Joshua Steven Chiang - Software Testing Master

Alexandra Rauer - Cyber Security Co-Master

Elicia Baranowski - Cyber Security Co-Master

sdmay25-10@iastate.edu

https://sdmay25-10.sd.ece.iastate.edu

Executive Summary

True Force Technologies is seeking to expand its product to a global market. The "Force Rack," built by True Force Technologies, features a barbell attached to load cells designed to convert mechanical isometric force applied by the user into a digital output. This output translates directly to exercises performed in the weight room, providing measurements and data that help coaches and athletes make more informed decisions using objective data retrieved from the "Force Rack. True Force Technologies is currently targeting college athletes and athletic coaches as their main customer for the “Force Rack.”

This project will utilize the existing website, backend, firmware, “Force Rack”, and tablet application to create a new mobile application. It will incorporate all user and admin features of the existing tablet application and have access to the website/database, leveraging the existing Bluetooth stack to ensure a reliable hardware connection with minimal user effort. The new mobile application looks to improve on the existing product by fixing bugs the current application is facing and redesign existing features to improve customer usability. The goal is for athletes and coaches to easily be able navigate the application during workouts in a quick and intuitive manner. The app will also meet industry standards for development and security.

During the first semester, this project was started by splitting into 3 teams. One focused on iOS development, one focused on Android development, and another focused on security considerations. The project design was created and implemented for two apps, one using Swift and the other using Java. The feedback given to the team from the faculty review panel was to create one singular app using React Native. Starting in the second semester, the team switched directions to a singular app and reconfigured the project to be developed by 2 teams. One focused on the new React Native app and the other continued to focus on security considerations.

Screen design and navigation is completed on the react team. Security options have been researched and are ready for client confirmation. The app is currently being developed to connect with the rack. Next steps include integrating live data from the rack to the device, adding higher level functionality to the app, and incorporating security features.

Learning Summary

Development Standards & Practices Used

- Peer Code Reviews
- Software Testing
- Commenting Code
- Normal naming conventions
- ABET Criteria:
    - Apply knowledge of mathematics, science, and engineering
    - Design a system, component, or process to meet desired needs within realistic constraints
    - Identify, formulate, and solve engineering problems
- ISO 639 - Language Code
- ISO 14000 family - Environmental management
- ISO 31000 - Risk Management
- ISO 9660 - ISO images for computer files
- ISO 9241 - Ergonomics of Human-system interaction
- ISO 27001 - Information security management
- IEEE P3411 - Standard for smart identification in Internet of things

Summary of Requirements

- App shall have the same functionality as original tablet application provided by the client
- App shall be easy and intuitive to use for fitness focused users
- App shall measure the maximum strength of a lift and store that information to a user's profile
- App shall be able to display data about the lifts
- User shall be able to use the application on their personal device
- Product should be able to create new users
- App shall be able add users to different teams
- User shall be able to create new exercises
- Hardware shall disconnect with the software when the application is closed
- App shall be able to connect via bluetooth to the hardware on the weight rack
- User should be able to see data about past lifts in a graphic display
- App shall be scalable on different screen sizes (tablets, phones, etc.)
- Hardware should be available to all devices to connect properly via bluetooth.
- App needs to reference the api and database to log and chart data.

- App shall use a graph to display the force the user exerts on the rack.
- App shall be in compliance with design standards and should be easily used by all types of users
- App shall follow a clear and consistent color scheme
- App shall rotate depending on the device's orientation.
- App shall not use red/green colors as these are hard for user's with color blindness to see
- App shall have different features for different types of users (coach can add users to a team)
- App shall be able to work in a room with multiple devices with their bluetooth on
- Hardware shall stay connected to one device if others have their bluetooth on

Applicable Courses from Iowa State University Curriculum

- COM S 227 - Object-Oriented Programming
- COMS 228 - Introduction to Data Structures
- COM S 363 - Introduction to Database Management Systems
- S E 309 - Software Development Practices
- S E 317 - Introduction to Software Testing
- S E 319 - Constructions of user interfaces
- CBYE 230 - Cyber Security Infrastructure
- CBYE 231 - Cyber Security Concepts and Tools
- English 314 - Technical Communication
- English 309 - Proposal and Report Writing
- S E 491 - Senior Design Project I

New Skills/Knowledge acquired that was not taught in courses

List all new skills/knowledge that your team acquired which was not part of your Iowa State curriculum in order to complete this project.

- ReactNative
- Swift programming language
- Bluetooth stack connectivity
- Live data incorporation into a mobile application
- Real world cyber security advancements
- Knowledge of cloud databases and management
- Bluetooth connectivity and data transfer

Table of Contents

List of figures/tables/symbols/definitions

1. Introduction

1.1. Problem Statement

As technology progresses, so does every industry alongside it, including the way that we train. Learning how different environments and factors can affect performance is imperative to athletes. Small day-to-day habits can affect the strength athletes produce and influence performance. However, there is no exact science behind how these factors can affect each athlete. Therefore, the best way to measure an athlete's force output is to personalize their workouts. This, compared to other competitors on the market, utilizes the power and training a person has, rather than only being driven by personalized inputs such as weight, height, and calorie intake. TrueForce Technologies is built around the idea of athletes as individuals and testing their power to produce the best workout possible for each athlete. This technology can also prevent injury by not pushing an athlete past their strength level on an "off" day, when several injuries occur.

Currently, TrueForce Technologies has an existing application embedded into a M10 Lenovo tablet, but the application can't be downloaded from any other device. This is an issue as multiple coaches might want to look at the output of their athletes on their own devices rather than sharing one tablet for a whole team or program. To address this issue, our team will create a React application. This will allow any user to be able to download the app on their preferred device using the Apple App Store or Google Play Store. A user will then be able to easily log in to view the data for their athletes or connect to any TrueForce rack and record a lift. The current tablet application includes many bugs that affect the user's navigation throughout the app. It does not currently meet our users' standards and needs to effectively use this application. Our team will address these bugs when developing the new application in order to successfully satisfy our user's and client's needs as well as meet industry security standards.

1.2. Intended Users

The intended users of our project include student-athletes and coaches (athletic and positional). The athletes are split into two categories, in-season and off-season. One persona of our project is an in-season athlete (Figure 1.1). An in-season athlete needs an easy and accurate way to measure their maximum strength for the day because they want to perform and train at their best. They also do not want to spend their energy on training instead of their sport. They are less concerned about increasing their strength and more concerned about maintaining it. The in-season athletes are busier than off-season athletes, so using this product will help them decide what training to do for the day and how heavy to lift. By using this product, they are able to decrease the amount of time they worry

about what to lift and how much. Using the product would also decrease their risk of injury because the athletes would know what their maximum force output for the day is and wouldn't overexert themselves during workouts. These benefits connect to the overarching problem because by using the product, they get the answer they need from how they perform to what they need to do to increase their strength. Using the product will allow this user to be a standout in their sport and become an overall better athlete. The product will allow for better availability of their personalized data. The athletes will be able to see their data from any device, which will allow them to train correctly during their season without having to use the hard-coded tablet.

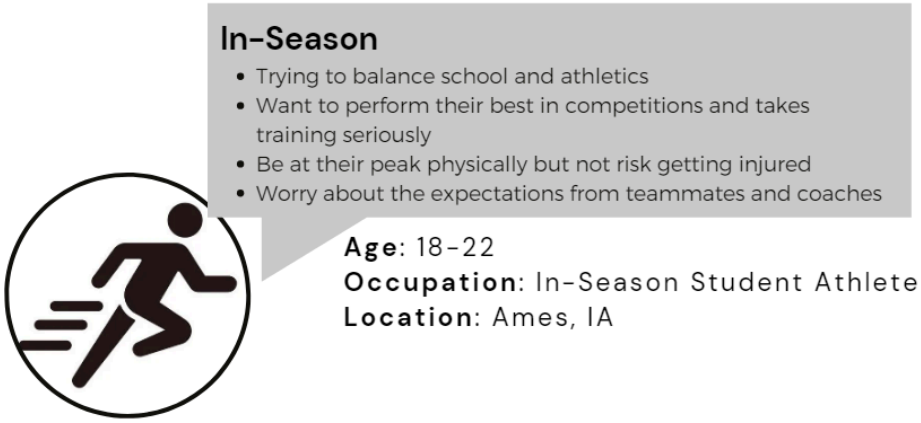

Figure 1.2.1 In-Season Athlete User Persona

Another persona we created for our project is an off-season athlete (Figure 1.2). An off-season athlete needs an easy and accurate way to measure their metrics to gain strength and reach their off-season goals in a healthy way. They are not as busy as in-season athletes, and they would use the product to measure their metrics over time to help gain strength and prepare for their upcoming season. A benefit would be that during the off-season, athletes would be able to clearly see the progress they're making with the data tracking and see how well they are meeting their strength goals. The benefits relate to our main problem statement concerning personalized workouts for each athlete, and data tracking that monitors athlete performance and progress. The product will allow for better availability of their personalized data. The athletes will be able to see their data from any device, which will allow them to train correctly when preparing for their season without having to use the hard-coded tablet.

Figure 1.2.2 Off-season Athlete User Persona

The final user is the coach (Appendix 1.3). The coach can be an athletic trainer whose job is to oversee athlete strength progression, or it can be a positional coach, such as head/assistant coaches. Both types of coaches are interested in their athletes' progression in strength to improve their overall performance. Coaches need a way to monitor and track the strength and progress of all of their athletes because it will allow them to help each player perform their best. The coach would use this application to track individual athlete progress and design workouts around their progress. This results in coaches being able to keep track of each individual athlete's needs, making sure that they're provided with workouts that are best suited for them, which also reduces the coach's worry about athlete injuries. These benefits are accomplished by the product's availability and mobility, allowing multiple coaches to monitor data for multiple athletes across multiple teams.

Figure 1.2.3 Coach User Persona

2. Requirements, Constraints, And Standards

2.1. Requirements & Constraints

Functional requirements

- The application shall measure the maximum strength of a lift and store that information to a user's profile
- The application shall be able to display data about the lifts
- The user shall be able to use the application on their personal device (iOS and Android apps) **(constraint)**
- The user shall be able to see data about past lifts in a graphic display
- The product shall be able to create new users
- The application shall be able to add multiple groups (teams) of users
- The product shall be able add users to different teams
- The user shall be able to create new exercises
- The hardware shall disconnect with the software when the application is closed.
- The application shall be able to connect via bluetooth to the hardware on the weight rack.
- The appl shall be available on both Apple App store and Google Play

Resource Requirements

- The hardware shall be available to all devices, in order to properly connect the application to the rack via bluetooth.
- The application shall reference the api and database to log and chart data.
- The application shall use an external API to graph the force the user exerts on the rack.

Physical Requirements

- The rack will be placed weight room or any space where a coach might want to conduct their training
- The application shall be scalable on different screen sizes (tablets, phones) **(constraint)**
- The software shall run on any smart application that uses Google Play Store or Apple app store. **(constraint)**

Aesthetic Requirements

- The application shall be in compliance with design standards and should be easily used by all types of users
- The application shall follow a clear and consistent color scheme
- The application shall use colors that are accessible to all **(constraint)**

User Experiential Requirements

- The application should rotate depending on the device's orientation.
- The application should have different options for different types of users (coach can add users to a team)
- The application shall use colors that are accessible to all **(constraint)**

Economic/market Requirements

- The product should be affordable for schools, gyms, and other training centers.

Environmental Requirements

- The software shall be used in a weight room**(constraint)**
- The product should be able to work in a room with multiple devices with their bluetooth on
- The hardware should stay connected to one device if others have their bluetooth on
- The software shall be used with the physical weight rack near it.

UI Requirements

- The application should have a user-friendly interface that enables quick navigation for users of all experience levels.
- The UI should have clear, easy-to-read fonts and buttons
- The layout should be adaptive to different screen size (constraint)
- The application should provide a clear display of the lift data and user stats

Security Requirements

- Security measures will fit into client's budget
- Users' data will be secure and not visible to other users through database management system
- Able to securely manage user permissions through database management system to ensure data privacy
- Login will have an extra layer of security with two-factor authentication

2.2. Engineering Standards

Importance of engineering standards:

Without maybe realizing it, engineering standards are everywhere around us. Not just at school or work, but at home, when using transportation, and doing day-to-day activities. Therefore, we need a defined way of doing things, which is why standards exist. They exist as an agreed-upon way to do things. The IEEE engineering standards are set in place so that they can be universally understood and applied, and as a way to promote technology that functions safely, securely, and sustainably. Standards help when developing new products to define the requirements and best practices to follow to make sure the end result is dependable and reliable. In our senior design project, we will utilize standards at each step of development to make sure our product is reliable and adheres to the best practices defined by the existing standards. [5]

Relevant Standards:

ISO 9241 - Ergonomics of Human-system interaction [9]

This standard covers usability and user experience design, involving components such as the presentation of information and menu dialogues. It is meant to ensure that the software is user-friendly, learnable, and accessible to all users.

ISO 9241[9] is important to the project because we are designing accessible and user-friendly software. The functionality and features of the software should be presented through the interface in a learnable manner to allow users to complete functions with little to no confusion or errors. The design of the user interface is meant to be simplistic yet easy to understand when carrying out tasks. This requires the software to conform with the users' expectations, provide system feedback to user interactions, and be self-descriptive when displaying tasks.

ISO 27001 - Information security management [7]

This standard is intended to provide guidance for managing a security system. Conformity with this standard means that all the best security practices have been put in place. It is intended to accomplish better security practices and systems for companies of all sizes.

ISO 27001[7] is relevant because there currently is not an information security management system in place for the application. The standard requires examining the organization's security risks, designing a system to manage and control those risks, and adopting an ongoing process to ensure that the security needs remain met over time. These same three steps are necessary when creating a security management system for the

TrueForce Technologies application. Conformity with security standards will be necessary for the app as it is planned to be available on app stores and used by institutions that do not want their information leaked.

IEEE P3411 - Standard for smart identification in Internet of Things [4]

This standard is intended to provide details on how the IoT devices should be handled within the project. It provides a framework between IoT devices using transparent data transfers and existing protocol standards to ensure the proper use of IoT devices.

IEEE P3411[4] is highly relevant since the project involves hardware-software communication via Bluetooth, as it standardizes identifiers and addresses for smooth, secure data transport. Without this standard, our transfer of data could be at risk of leaking information. Since our application owner is trying to sell the app to multiple schools around the country, it is extremely important to have secure IoT devices.

Other Standards:

ISO 14000 family - Environmental Management [6]

This standard specifies how organizations can minimize their environmental impact and comply with related laws and regulations through environmental management. It is important to recognize the impact that the physical rack will have during its creation to ensure that those involved in the process are safe. It is also important to recognize the environmental considerations for the physical rack in a fitness location such as safety measures, toxicity standards, and more. However, this standard applies to the physical aspect of this project, so it is not a major focus for us.

ISO 639 Language Code [8]

This standard specifies the formatting and expectations of systems to allow different languages. We saw this as important because this technology is intended to be used around the world, however, we are not strictly focusing on this standard as it does not fit within our timeframe to have the application switch between languages that our team is not fluent in.

Design Modifications:

To incorporate ISO 9241, we are redesigning the app to allow users to find information easily, this includes visible tabs on the home page as well as the orientation of the application following the orientation of the tablet or device in use. We also are going to change the contrast colors that show the differing strength coming from the left and the right(of the body) so they are visible by those who are color blind. To add onto this, we will

be structuring the data received by each lift in a manner that the most important data is easy to see after the lift is done, whether that is to enlarge the key data or structure the information in a way that the key data is the first to catch the eye of the user.

To ensure information security management, the main focus will be to secure the database, since it currently resides on an easy-to-use website to be accessed by anyone. Current ideas are to use Okta to provide a secure login which also prevents injection and brute force attacks. This standard will also include us looking into the structure of the database to ensure that coaches who have access to more aspects of the application to oversee athletes cannot look at athletes from different schools. This will include a knowledge of permissions in different languages.

For the IEEE P3411, our main design to incorporate this aspect is to make sure that the device/application does not continuously search for a bluetooth device after connection, as this drains battery life and can be used in a security threat.

3 Project Plan

3.1 Project Management/Tracking Procedures

The project management style that is best suited for this project is agile. Bi-weekly meetings have been set up with our advisor and client. Agile works well for us to be able to run two-week sprints and then demo what we have completed in those two weeks the next time we meet with our advisor and client. Agile allows us to take an iterative approach to developing the application. We don't need to be held up for another part of the project because it has been broken down into specific features that anyone can start working on without waiting for another part of the project first. Lastly, it allows us to measure our progress as we go, rather than waiting for a specific milestone to be hit.

In order to track our progress throughout the course of both semesters, our team is utilizing Git. A GitLab has been created for us to track the project. We have created issues for each component of the project so that when someone is ready, they can move the issue into the developing column of our issue board. Once completed, we assign another team member to code review the issue before merging to our main branch. Our advisor and client also have access to GitLab to view our design documentation and track our progress too.

3.2 Task Decomposition

The tasks for our project are broken down into 2 tasks: app development and security solutions. The development sections include the same subtasks to create the application, develop the screens for the project, connect the screens together, connect the application to the existing backend, connect to the physical rack, and testing. The security section included the subtasks to develop cybersecurity solutions, research industry standards and requirements, integrate solutions into the application, integrate solutions into the database, and penetration testing (See figure 3.2.1).

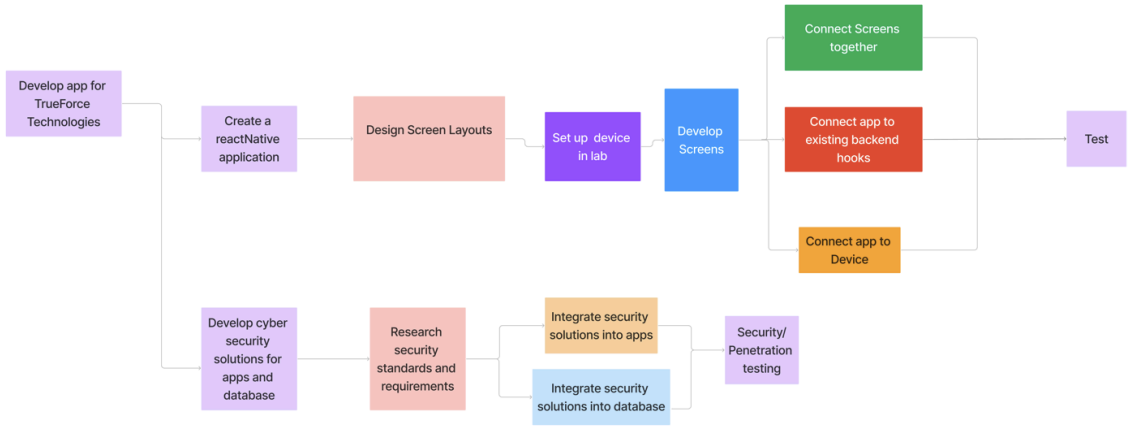

Figure 3.2.1 Task Decomposition Chart. Flowchart of tasks needed to complete the project that specifies order in which development must be done.

Not all of these subtasks are dependent on the ones before it. For example, screens can be connected together before all the screens are created. This allows us to work in sprints to get working functionality for our client without being held back by small issues in specific subtasks or missing client information.

3.3 Project Proposed Milestones, Metrics, and Evaluation Criteria

First semester milestones:

Screen designs are implemented following the user interface designs described by the client's documentation and demonstration. 100% of the screens are implemented and viewable in both iOS and Android applications.

Navigation between screens is implemented aligning with the client's documentation. 100% of the screens are accessible through the navigation flow of the iOS and Android applications.

Second semester milestones:

iOS and Android applications are migrated together into one React Native application. React Native application has 100% of the screams implemented with the same navigation as semester 1.

The Cyber Security team successfully wrote a set-up document for future teams to implement the security solution our client and team decided on. Document is 100% complemented.

User manual, how-to video, bluetooth documentation, and overall project documentation is 100% completed by the development team.

3.4 Project Timeline/Schedule

Our Gantt chart (See figure 3.4.1) shows the time in which each task is projected to be finished. However, the connection and implementation of the application may be extended into next semester due to bugs in the backend and database.

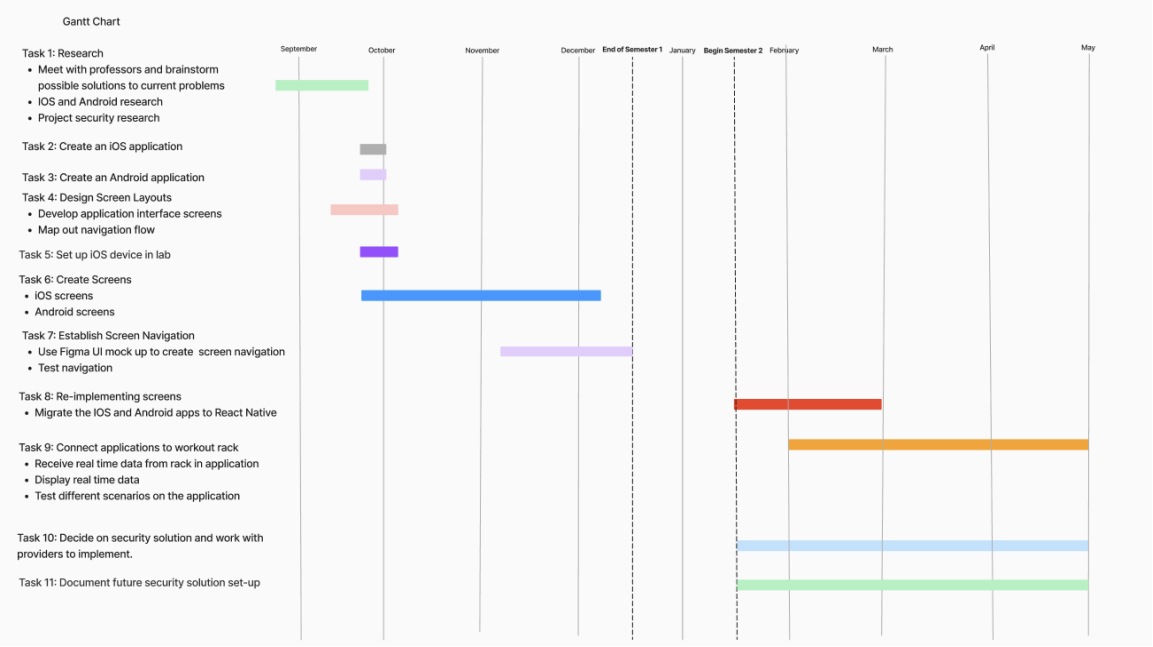

Figure 3.4.1 Gantt Chart. Image of proposed tasks to complete project and timeframe for semester 1 and 2.

Below are the subtasks that are included in a Gantt chart:

- **Task 1: Research**
    - Meet with professors and brainstorm - end of September

    - Research and familiarize with Android and IOS - mid-October
    - Pen test application and website to find connections in the product - beginning of October
    - Research applications to better secure database and two-step authentication - end of October
- **Task 2: Creating an Android application**
    - downloading and creating Android application - beginning of October
- **Task 3: Creating an IOS application**
    - downloading and creating IOS application - beginning of October
- **Task 4: Create Design screen layouts**
    - Create Designs in Figma and replicate the actions to connect the screens together - mid-September
- **Task 5: Set up the IOS device in the lab**
    - Log in to the computer device and set up ssh/establish connection to lab computer on personal computers - mid-September
- **Task 6: Create Screens**
    - Create IOS screens - mid-October
    - Create Android screens - mid-October
- **Task 7: Establish screen navigation**
    - Use Figma to create mock-up of screen navigation - End of September
    - Test navigation - beginning of October
- **Task 8: Re-implement Screens**
    - Mitigate the IOS and Android apps to React Native - End of February
- **Task 9: Connect applications to workout rack**
    - Receive real time data from rack to application - Mid May
    - Display real time data - Mid May
    - Test different scenarios - Mid May
- **Task 10: Decide on Security Solutions**
    - Decide on security solutions and work with providers to implement- End of May
- **Task 11 : Document future security solution set-up**
    - Create a well-detailed document for easy setup for future teams- End of May

Due to our agile model, we also have the following sprint schedule

- **Sprint 1: Product Design (Start of September - Start of October)**
    - Meet with client and advisor to get stakeholder needs and required functionality
    - Decide how to approach project logistically
        - Team member assignment, concurrent/sequential app development, etc.

    - Create screen designs for system UI and navigation overview
    - Set up add development environment
- **Sprint 2: Screen Development and Security Features (Start of October - Mid December)**
    - Create dynamic screen in development environments according to UI designs
    - Establish navigation between all screens
    - Implement basic functionality of system
        - On-click events, editable text fields, etc.
    - Decide on security features to be implemented based on research and penetration testing
- **Sprint 3: Migrate Applications to React Native (Mid January - March)**
    - Combine IOS and Android applications into one React Native application
    - Ensure the same functionality and user experience as previous apps.
- **Sprint 4: Hardware and Security Integration (Mid January - May)**
    - Create bluetooth connection between hardware rack and software
    - Implement live data updating between hardware and software
    - Begin implementation documentation of security features for the application
    - Implement any missing app functionality
- **Sprint 5: Testing (February - May)**
    - Ensure correct app functionality
    - Prepare for final product presentation

3.5 Risks and Risk Management/Mitigation

Database Access

- **Risk:** Unable to store data collected in the app to the database. Unable to make sure that the data is stored safely and transferred securely.
- **Likelihood:** Moderate (0.5)
- **Consequences:** Major (Would not be able to secure or publish the app)
- **Mitigation:** Ask the client for database information or create a cloud-based solution on our own. If unable to gain access to the database, the applications will be able to demonstrate the quickstart feature with live data and be ready to add the API calls once database information is available.

Lab Setup

- **Risk:** Unable to set up an iOS environment, board, and rack in the lab before testing
- **Likelihood:** Unlikely (0.2)
- **Consequences:** Major (Would not be able to test the app with live data)

- **Mitigation:** Ask the client for a rack and use existing or made-up data.

Backend Connectivity

- **Risk:** Unable to get the API hooks from the current development team to connect our screen designs to the database and the hardware on the rack
- **Likelihood:** Unlikely (0.2)
- **Consequences:** Major (No data could be stored or retrieved, making our project unusable)
- **Mitigation:** Alert the client of necessary database information needed so the advisor and client can work towards delivering them to us.

Hardware Connectivity

- **Risk:** Unable to connect to the Force Rack through the Bluetooth stack, the team does not work on the hardware portion of the product
- **Likelihood:** Moderate (0.5)
- **Consequences:** Major (App would not receive any athlete data)
- **Mitigation:** Planned out minor testing in parallel with app feature development to frequently check the connection status as more functionality is added to the application, enabling the team to notify the advisor if there are issues.

Integrated Security Solutions

- **Risk:** Unable to identify and patch all security vulnerabilities
- **Likelihood:** Moderate (0.5)
- **Consequences:** Major (Leaves app and backend database vulnerable to attacks)
- **Mitigation:** Setup an intrusion detection system and two-factor authentication to mitigate as many security vulnerabilities as possible. Create an incident response plan for future security incidents.

Most of our risks came to fruition, all but one. We were unable to get access to the database, API hooks and firmware for the backend. This caused the security risk to also be a reality. Because of this, our team had to pivot with the outlook on the project. Instead of connecting to the backend, we made documentation for the next team to successfully use what we created to connect to the backend. We did the same for the security solutions since we did not receive any of the database or backend information, we were not able to implement the security solutions. Our team made it a priority to document what our team has completed and what still needs to be done to help the next team move forward with the ultimate goal of a working application.

3.6 Personnel Effort Requirements

Estimated Personnel Effort Requirements Table

Task	Time (hrs)	Explanation
Existing Application Research	1	This task involved the whole team working through the existing tablet application, noting improvements, studying the flow, and gaining an understanding of our application to be created.
Create Android Application	1	This task involves creating a new project on Android Studio and connecting GitHub to the project.
Create iOS Application	2	This task involves creating a new project on Xcode and connecting GitHub to the project. This task included time to set up the emulator and learn how it worked. Our team has no experience with Swift.
Design Screen Layouts	5	This task involves using Figma to create a reference for the layout of every possible screen in both horizontal and vertical orientation based on the pre-set designs given by the client. This task also includes a flowchart of the navigation between screens.
Set up iOS development environment	2	Setup of project workstation in the senior design lab. This involves: MacOS setup, downloading Xcode and SwiftUI playground, setting up remote access, receiving locker, tablet, and extra monitor.
Develop screens for Android	55	Creation of each screen as its own file with correct formatting for different orientations and sizing. Each screen development should be a separate Git issue and branch. This includes merge conflicts and other Git issues.
Develop screens for iOS	60	Creation of each screen as its own file with correct formatting for different orientations and sizing. Each screen development should be a separate Git issue and branch. This includes merge conflicts and other Git issues.

Connect Screens for Android	40	Addition of screen functionality into the project. This includes proper screen navigation based on user actions, simple features that don't require database information, and the skeleton code for backend implementation if possible. This includes merge conflict and other git issues.
Connect Screens for iOS	40	Addition of screen functionality into the project. This includes proper screen navigation based on user actions, simple features that don't require database information, and the skeleton code for backend implementation if possible. This includes merge conflict and other git issues.
Connect Android application to existing backend	20	Establishing the connection between the frontend and backend. Implementing the API hooks to call and push information to the database. Finishing code for functionality requiring backend. This will likely have a lot of problem solving as we can't see the backend code. This includes merge conflicts and other Git issues.
Connect iOS application to existing backend	20	Establishing the connection between the frontend and backend. Implementing the API hooks to call and push information to the database. Finishing code for functionality requiring backend. This will likely have a lot of problem solving as we can't see the backend code. This includes merge conflicts and other Git issues.
Connect Android application to hardware rack	20	Establishing the connection between the frontend, backend, and hardware so that data from the hardware rack is properly transmitted through frontend and backend. This will include research in regards to connection the rack with mobile devices as well as Bluetooth implementation research. This includes merge conflicts and other Git issues.
Connect iOS application to hardware rack	20	Establishing the connection between the frontend, backend, and hardware so that data from the hardware rack is properly transmitted through frontend and backend. This will include research in regards to connecting the rack with mobile devices as well as Bluetooth implementation research. This includes merge conflicts and other Git issues.

Application testing	80	Test both Android and iOS apps to ensure correct functionality. Things to test include for example: rigorous navigation, improper data inputs, attempts to “break” the system, and graph scaling changes. This includes fixing code when problems are found and git issues.
Develop Security Solutions	40	Design possible security features for both Android and iOS applications and the database. Includes meeting with security professors to determine applicable solutions and feature research. Communicate with clients about the best course of solutions and find solutions within budget.
Research Security Standards & Requirements	50	Investigate industry standards to ensure compliance and identify necessary security controls for the application.
Integrate Security Solutions for Applications	40	Put the developed security solutions into both Android and IOS applications, ensuring secure user authentication and data handling.
Integrate Security Solutions into Database	40	Implement encryption and secure access protocols within the database to protect sensitive data at rest and during transactions.
Penetration Testing	30	Conduct comprehensive penetration tests on both applications and the database to identify vulnerabilities and strengthen security measures. This includes testing the original application to find security gaps and testing the new iOS and Android applications to ensure the gaps were covered.

Table 3.6.1 Estimated Personal Effort Requirements Table. List of project tasks and the estimated time required to complete each along with description of subtasks.

Actual Personnel Effort Requirements Table

Task	**Time (hrs)**	**Explanation**
Existing Application Research	1	This task involved the whole team working through the existing tablet application, noting improvements, studying the flow, and gaining an understanding of our application to be created.

Create Android Application	1	This task involves creating a new project on Android Studio and connecting GitHub to the project.
Create iOS Application	2	This task involves creating a new project on Xcode and connecting GitHub to the project. This task included time to set up the emulator and learn how it worked. Our team has no experience with Swift.
Design Screen Layouts	5	This task involves using Figma to create a reference for the layout of every possible screen in both horizontal and vertical orientation based on the pre-set designs given by the client. This task also includes a flowchart of the navigation between screens.
Set Up iOS Development Environment	2	Setup of project workstation in the senior design lab. This involves: MacOS setup, downloading Xcode and SwiftUI playground, setting up remote access, receiving locker, tablet, and extra monitor.
Develop Screens for Android	55	Creation of each screen as its own file with correct formatting for different orientations and sizing. Each screen development should be a separate Git issue and branch. This includes merge conflicts and other Git issues.
Develop Screens for iOS	60	Creation of each screen as its own file with correct formatting for different orientations and sizing. Each screen development should be a separate Git issue and branch. This includes merge conflicts and other Git issues.
Connect Screens for Android	40	Addition of screen functionality into the project. This includes proper screen navigation based on user actions, simple features that don't require database information, and the skeleton code for backend implementation if possible. This includes merge conflict and other git issues.
Connect Screens for iOS	40	Addition of screen functionality into the project. This includes proper screen navigation based on user actions, simple features that don't require database information, and the skeleton code for backend implementation if possible. This includes merge conflict and other git issues.

Create React App	5	This task involves creating a new project on Visual Studio Code and connecting GitHub to the project. This task included time research React Native Application. Our team has no experience with React Native.
Redevelop Screens for React	50	Creation of each screen as its own file with correct formatting for different orientations and sizing. Each screen development should be a separate Git issue and branch. This includes merge conflicts and other Git issues.
Connect Screens for React	40	Addition of screen functionality into the project. This includes proper screen navigation based on user actions, simple features that don't require database information, and the skeleton code for backend implementation if possible. This includes merge conflict and other git issues.
Connect React Application to Hardware Rack	80	Establishing the connection between the frontend and hardware so that data from the hardware rack is properly transmitted to frontend. This will include research in regards to connection the rack with mobile devices as well as Bluetooth implementation research. This includes merge conflicts and other Git issues.
Application Testing	30	Test both Android and iOS apps to ensure correct functionality. Things to test include for example: rigorous navigation, improper data inputs, attempts to "break" the system, and graph scaling changes. This includes fixing code when problems are found and git issues. This includes testing the new Reactive Native Application.
Research Security Standards & Requirements	30	Investigate industry standards to ensure compliance and identify necessary security controls for the application.
Determine Security Solution and Work with Providers to Implement	30	Research and Test different products through free trial, and research what tools should be used by clients to meet Industry standards and report on our findings.
Document Future Security Solution Set-Up	20	Create a step by step document that fully explains the most influential tools while explaining steps to create a database. Also included a cost breakdown

		and comparison with a competitor.

Table 3.6.2 Actual Personnel Effort Requirements Table. List of project tasks and the time spent to complete each along with description of subtasks.

From the first semester the total number of hours estimated for this project was 566 hours as calculated from Table 3.6.2. Taking into account the fall semester (14 weeks) and the spring semester (16 weeks), the estimated time per week anticipated to be spent on the project can be rounded up to 19 hours. Our team has 7 members, which means per person, each individual will spend around 2.5 hours per week.

However, our team ran into some roadblocks and did not get all of the estimated work done as expected. The actual total number of hours spent working on this project was calculated to be 491 hours. Taking into account the fall semester (14 weeks) and the spring semester (16 weeks), the time per week spent on the project can be rounded up to 17 hours. Our team has 7 members, which means per person, each individual spent around 2.5 hours per week on the project.

3.7 Other Resource Requirements

Resources needed to complete this project are the physical weight rack and sensor. These both already exist and will be utilized to connect to our application. Another resource required to complete this project is the database. TrueForce Technologies has an existing database that our app will connect to, in order to save specific information for each user. However, this database is currently controlled by a third party that does not allow us to have access. To overcome this, we have decided to create a mock Azure database to simulate data transfer and find how this product will be used by the client in the future. Another resource is the firmware connected to the rack. The firmware allows the rack to connect via bluetooth to the new app. But since we do not have the code for it, we were unable to connect. Since the original team used a special configuration of BLE connection, we are not able to connect to bluetooth without knowing what the connection is. To combat this we have documented everything we have been able to do so the future teams can build upon our work when they receive the code.

4 Design

4.1 Design Context

4.1.1 Broader Context

Currently, the project is specific to improving the performance of athletes, with the main users being coaches and their athletes. In the broader context, this is a product that has the potential to help users of all athletic abilities. This ranges from people stepping into the gym for the first time who have no idea where to start to seasoned athletes who want to be the best athletes they can be. It could be used in both commercial gyms as well as university athletic programs. However, these communities are very different and require different designs. Therefore, the community we are designing for is the college athletic programs.

Broader Context Areas

Area	**Considerations**
Public health, safety, and welfare	Encourage more people to work out Reduce chance of injury while working out Usable for those with colorblindness
Global, cultural, and social	Not usable by those without access to affiliated gyms Not usable by those without personal devices Not culturally offensive Encourages relationship between coaches and athletes
Environmental	Metal mining and processing for rack Increased electricity consumption
Economic	Increase gym membership Reduce strength training jobs

Table 4.1.1 Broader Context Table. This table lists some considerations made about the product's effect on broader contexts.

4.1.2 Prior Work/Solutions

Existing Products in the Market

Existing Products	**Description**	**Pros**	**Cons**
CoachMePlus [2]	Data collection and analysis software to help coaches manage their athletes	Useful features including in-app messaging, cross-data reporting, historical trend reporting, leaderboards, custom dashboards, and	Users have to put all the data manually. The data can be inconsistent, and it can take a lot of time for

		store photos/videos. The company also has experience working with pro athletes and teams.	the user to input the data
Catapult One [1]	Delivers real-time data from a sensor that athletes wear during practice from yards, speed, force output, and more.	Used by many DI programs as well as several major league sports. The amount of data collected in real-time makes it appealing to coaches.	Drawbacks exist in the battery life of the units connected to the app. As well, the live tracking on the app doesn't handle the transition between outside (GPS) and inside (LPS) data.
Output [10]	Sensor and app used by coaches for velocity-based training	The sensor is small and has a lot of different functionalities. There is an app that tracks all the data over time. The data is stored and visualized in graphs to see improvement.	Can be expensive, making them prohibitive for nonprofessional athletes and coaches.

Table 4.1.2 Existing Product In The Market Table. This table showcases three products currently on the market that are similar to our product and the positives and negatives of each.

Our team's product/solution will stand out from the existing products because it is taking the current product and making improvements. The product already exists, we are tasked with making improvements to the UI, functionality, and ability to be more readily available. Instead of the product being specifically designed for a single type of table. Our product will be available on both the Apple and Google Play Store allowing any device and any client to use our product. Our product will have automatic data entry and understandable data output. It will focus more on individuals' strength instead of BMI, height, and weight when determining correct workout procedures, giving more accurate information to females and those with non-average muscle ratios.

4.1.3 Technical Complexity

Internal technical complexity:

- Involves the integration of both hardware and software components
- Interconnection between the rack, board, database, and application

External technical complexity:

- Demonstrated by the improvements made on an existing technology

Technical complexity analysis:

- Unfamiliar with bluetooth connectivity
- Unfamiliar with reactNative
- Lack of information about established backend, bluetooth, and database.
- Need a graphing widget that can work with live data
- Unfamiliar hardware to connect with

4.2 Design Exploration

In this section we discuss how our team came up with the final design. This includes the bigger design decisions we had to make, our ideation process, and discussing pros and cons for various designs.

4.2.1 Design Decisions

Our team was given the task of implementing an existing design into a react application. This was decided on at the beginning of second semester. To do this, our team made remedations to migrate both applications from the first semester into one, into a react application. This decision affected the success of the project the most because it set us back a whole semester. Because our group decided to change our design at the end of 1st semester after faculty feedback, the development team had to start over. But because of this change, the future teams to work on this project will have an easier time developing and maintaining changes to the application.

The next design decision we made was to change what data about their lifts 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.

Another design that will change in the future is the database. Through research and exploring different options with our client we decided that Microsoft Azure cloud database will be the best solution. With an already reputable popularity among large corporations and countless add ons to create a database tailored to our application, Azure was the best choice among the contenders. Many alternatives we considered were databases and storage to be held and protected in a company's systems. It would not have the advantage of being able to tailor the storage systems, nor have the ability to have an in-house monitoring system of interactions with the data, like Microsoft Defender provides in their many services available. However, this will come later due to a pre-existing contract that does not allow us to transfer data from the pre-existing database to the cloud. Instead of implementation we have decided to create a doc to lead our client or future cybersecurity engineers through this design change in their system.

An 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

The potential options for our graph display was decided 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 by 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

A matrix was used 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.

Pros Vs Cons Table Comparing Current Design and Our New Design

	Previous TrueForce Design	**Our TrueForce Design**
Pros	• Accurately measure user's force input fast • Good graph visualization • Logical screen navigation	• Works on iOS and Android devices • 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 an 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

Table 4.2.1 - Pros Vs Cons Table Comparing Current Design and Our Design. This table shows the positives and negatives for the current product's mobile app and our proposed mobile app.

Pros Vs Cons Table Comparing Two Application and One React Application

	Two separate applications	**One react application**
Pros	• More personalized features • More experienced developers on each app • Developed in native language	• Easy upkeep • Smooth transition between teams • Compatible on all types of devices • Faster development for teams (Less duplicate tasks)
Cons	• Harder to upkeep • Challenging to pass to next team • Each app is only available on one platform	• Less experience developing • Less specialized by platform • Less freedom with special features • Not developed in native language

Table 4.2.2 - Pros Vs Cons Table Comparing Two Vs One Application. This table shows the positives and negatives for separating into two applications and combining into one application.

4.3 Final 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). 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 4.3.2

Our final deliverable for security is a set up manual that will be used to lay the foundation for security of the app. It includes detailed descriptions of various security add-ons that are available through Microsoft Azure as well as a step-by-step guide detailing how to create a new database.

Figure 4.3.1 True Force Weight Rack. This image shows the existing model for the weight rack that our mobile app will connect with.

Figure 4.3.2 System Block Diagram. This image shows the connection between the external systems in our project. The red parts indicate future development

4.3.2 Detailed Design and Visual(s)

Our design is developed using React Native, so the application will be available for all types of devices (iOS and Android). 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 backend are not being implemented or changed by our team. The backend and database were developed by a third-party 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. The rack's lift data is transmitted from the load cells to a connected application via Bluetooth Low Energy using a STM32 microcontroller.

The security manual provides an overview of the Microsoft Azure add-ons and database configuration details. The database set-up shows how to create a secure cloud-based database for the application. The security considerations and solutions section highlights the specific security risks we identified in the current application and describes which Azure products can be used to solve the issues.

4.3.3 Functionality

The design is 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.

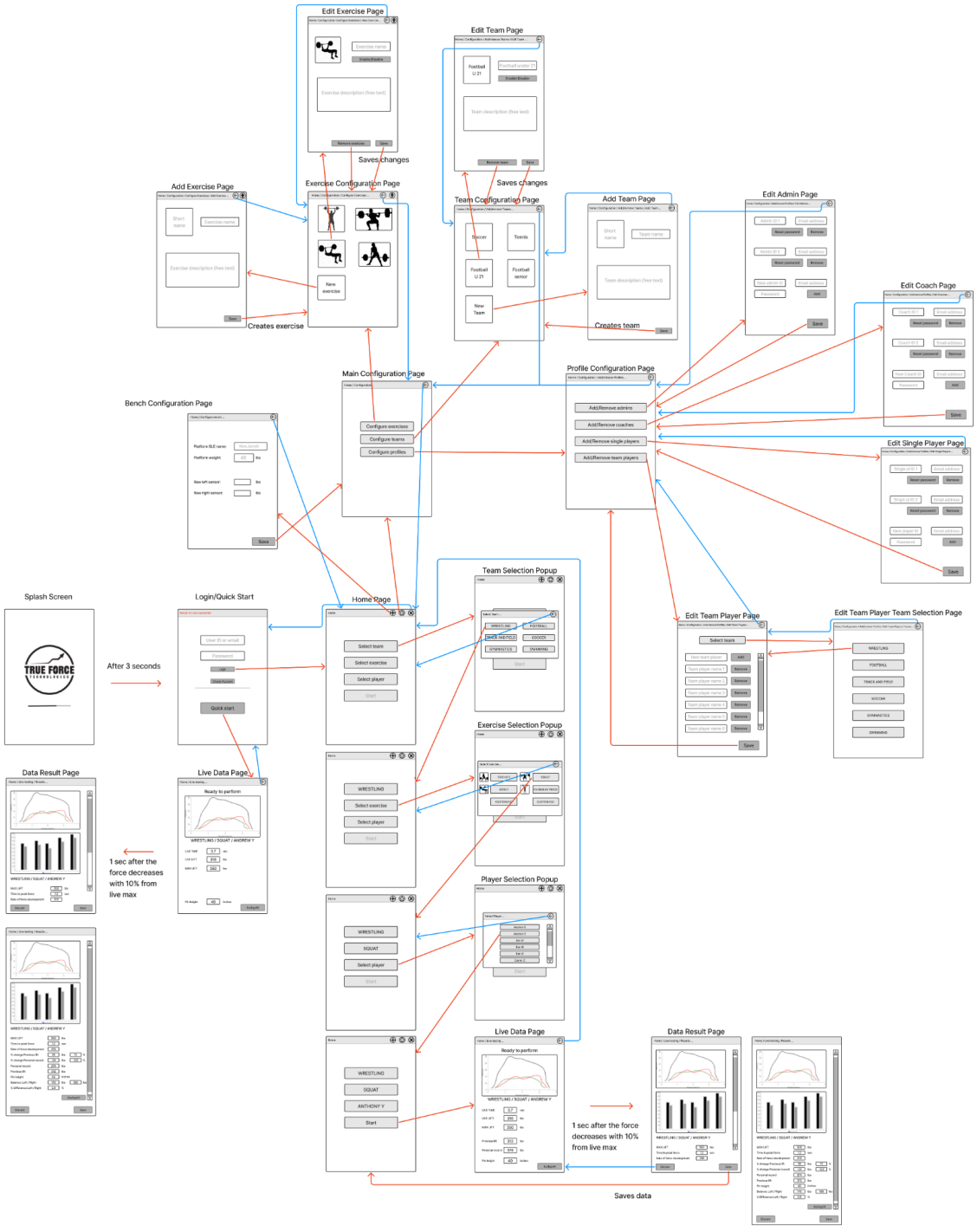

Figure 4.3.2 App Functionality Diagram. This image provides screen layouts and a navigation flowchart for our app.

4.3.4 Areas of Challenge

Based on our current design, the primary areas of concern are connecting the app to the rack and database. Our immediate plan of action at the beginning of the project was to receive database access from the client so we could store and retrieve data. However, the backend and database access were not granted throughout the development of the project. We addressed this challenge in the second semester by limiting our development goals down to migrating our existing UI screen implementations to React and attempting to implement the Bluetooth data transmission for the lift functionality.

Another challenge we faced was initiating the data transfer between the weight rack and our application. The Bluetooth Low Energy microcontroller that acted as a peripheral device had several services and characteristics to identify data values used for its implementation. Our intentions were to implement our application to act as a central device to detect the rack, connect via Bluetooth, and initiate data transfer for the lift data. The challenge we faced with the Bluetooth implementation was that the services and characteristics used to implement the rack's functionality were custom. Therefore, the data structure and processes for receiving data from the rack were unknown. Neither the firmware nor the original application's source code were accessible to aid in implementing our application's Bluetooth functionality. We attempted to overcome this challenge by requesting the rack's firmware in order to identify the functionality needed to trigger the data streaming for the weight rack. Unfortunately, the firmware access was not granted in time for our team. So, we turned our focus to documenting our implementations to help with an easy transition to future developers.

There are multiple security concerns when it comes to the implementation of the database. Last semester, we determined that the database and server were the most at risk parts of the design simply by our small interactions with them. Because we had no access to the backend, there was little that we could do. We decided to research security standards for industry, and after consulting with our client, our best course of action was to explore possible future solutions for the backend. Here we decided to explore one of the top contenders to see how it could be used through a free trial. We were able to determine how exactly we could set the new system up to meet the standards of our client and provided documentation.

4.4 Technology Considerations

One technology that we are in the process of developing is a cloud-based database. This will exponentially increase the security surrounding our data as it will be far from reach from any potential attackers instead of our current database which is more accessible at this moment. Azure also has more distinct technologies that we will be able to add on later to connect to the database to create a substantial security system with

two-factor authentication as well as Microsoft defender which allows us to go through logs of interactions with our systems and create alerts in the event of an attack, as well as privileged access control.

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 not have access to the API hooks to achieve our desired functionality as previously stated. This of course, has drawbacks since we do not have the names of the hooks or what they do so we cannot move forward with connecting them.

5 Testing

The testing plan mainly focuses on the user interface screens of our application. We verify that the design elements such as labels or buttons exist, verify that the interface matches with the proposed design, and verify that there is correct navigation between the application screens. It is unique to our project because we were provided with an existing version of the app that we used to create our screens. The testing we are conducting does not relate to all of our project requirements. Currently, our testing relates predominantly to the requirements about usability and appearance. This is because we are only concerned with the display shown by our created screens and how effective and intuitive they are to users. The requirements that are excluded from our current testing involve backend and database operations such as creating an account, saving data from a lift, or retrieving data from past lifts. As a result, our current testing plan is not comprehensive to the entire project because the functionality that uses the existing backend code is not being developed this semester because we don't have access to it currently. Our development team does testing during the development of the application screens with additional testing conducted through peer-reviews from other team developers before and after merging the code in GitLab.

5.1 Unit Testing

UI components and elements such as buttons or text fields are being tested to make sure they work properly. This is done manually by the developers of our team during development, and done after development by other developers on the team. The tools used to do this include the Android Studio emulator, the Xcode canvas preview, the XSimulator, and both Android and iOS tablets. We also verified Bluetooth state handling using hooks (e.g, useBLE) to ensure the app could detect BLE devices and initiate connections.

5.2 Interface Testing

Interface testing focused on verifying communication between user-facing components in our React Native application. We tested key interfaces such as the login screen, home screen, and analytics page to ensure proper navigation, interaction handling, and a consistent user experience. Actions like transitioning between screens and interacting with buttons were validated to confirm they triggered the correct behavior.

We also performed preliminary Bluetooth interface testing. This included verifying that the app could scan for nearby BLE devices, detect the True Force Rack, and initiate a connection using our Bluetooth logic. While we were able to establish a connection, we could not test data transmission due to lack of access to the firmware and characteristic definitions.

Since backend access was not provided, our interface testing remained focused on UI navigation, Bluetooth interaction, and internal app logic. Testing was conducted using tools such as React Native Debugger, Expo, Android/iOS emulators, and Bluetooth test devices.

5.3 Integration Testing

Some critical integration paths in our design are lift configuration, login, and account settings. Configuring the team, player, and type of lift before taking a reading from the Force Rack is important because lift data can vary depending on who lifts and what exercise is being performed. Login and account settings are necessary for the users' experience and different needs.

To test these, we will focus on how the data is exchanged between the UI screens, checking if it is correct and accessible to the screens in the navigation flow. For example, when a user selects the three parameters for performing a lift: team, exercise, and player, the data measurement and result screens display the correct labels for each lift. We also tested that the app could successfully scan for nearby BLE devices and initiate a connection with the rack. This included verifying that the Bluetooth module correctly detected the Force Rack, attempted a connection, and responded appropriately to connection success or failure on both iOs and Android. Our frontend code is responsible for interacting with the Force Rack, displaying user and lift data, and performing user actions such as logging in or adjusting preferences. The frontend then sends its data to the backend, which processes, stores, and retrieves data. We will also use the Android Studio emulator, the Xcode canvas preview, the XSimulator, and both Android and iOS tablets.

5.4 System Testing

We will use system testing to validate the application by making sure all existing components and features work together correctly to meet the requirements. This involves combining unit, interface, and integration tests to check that the full system operates correctly. Unit tests would verify that the UI components and other elements are functioning correctly. Interface tests will verify that interactions between the UI components, such as navigation, happen according to the design without any issues. Integration tests will check that the data exchanges between screens and backend occur correctly. We will also test that Bluetooth connectivity is integrated into the system by confirming that the app can scan for devices, connect to the Force Rack, and handle connection feedback properly. Using the combination of tests will ensure that complete actions can be performed and the components of the application implementation work together smoothly.

5.5 Regression Testing

When screens are created, it is done in a new, separate file from the other screen implementations. Any functionality that could be broken would be the navigation links between screens. We ensure that new additions during development don't break the existing functionality by conducting end-to-end testing after each new screen is added to our application. Screens that are created and added to the application are identified where they lie in the application workflow. Any existing application screens are also identified, and we test the new screen using a real-world scenario that includes the screen and the surrounding screens, making sure that the workflow is complete and correct.

5.6 Acceptance Testing

Demonstrating the functional and nonfunctional requirements can be done through a demonstration of the Quick Start functionality, shown to the client. The demonstration would include navigating through application and measuring lift data from the Force Rack, displaying the data on a result screen. Our client would be able to observe the demonstration to verify that the application meets user needs during the demonstration.

5.7 User Testing

Users were not involved in the testing process because of the limitations on the functionality of our application. Our team's developers took on the role of a typical user to assess how well our application design and implementation met user needs.

During the development of the application's user interface screens, our team used physical iOS and Android devices to display our coded interfaces. We assessed usability and the ease of navigation through the application's common workflows, such as performing a lift or making configuration changes. Another quality attribute we tested our design and implementation was interpretability. Because we had split our efforts to implement the application screens among several team members, our team had to coordinate and use a consistent and understandable format throughout the application.

5.8 Other Types of Testing

Because we were never able to access the backend, we worked with a cloud database (Microsoft Azure) to simulate creating a new database and how and where everything we do have (data, server, etc.) would be stored. However, because no one on the team could access the backend, we were not able to test the trial database nor connect our database to the old database. Thus it will be the next team's job to personalize and test the database.

5.9 Results

For our existing application screens, all components are present and as functional as they can be without backend functionality. The navigation paths between all existing screens are present and correctly directed, allowing for full navigation between existing screens. This allows the application to meet user needs by presenting users with all available functionality, and providing users with an interface to initiate actions.

Things we have learned are iOS development with Swift, Bluetooth connection and data transfer, more advanced Android development with Java, and additional libraries or frameworks for graphing displays. The development team learned how to migrate two applications together using React. We also learned more about cloud databases and management through research. Additionally, we explored different database solutions and how to apply our security knowledge to research and understand different security solutions.

6 Implementation

This semester, we developed a React app. We chose to build the app in React because it allowed us to maintain a single codebase for both Android and iOS platforms, making development and maintenance more efficient. We successfully developed the app screens and implemented screen navigation between them. Additionally, we created a development build of the app, preparing it for native functionality such as Bluetooth connectivity. Although we managed to establish a Bluetooth connection to the rack, we were unable to retrieve data from it due to a lack of access to the firmware information and the code from the previous developers. Without these details particularly the necessary characteristic UUIDs we were unable to complete the Bluetooth data communication functionality. Finally, we did not implement any backend functionality, as we were not provided with a database. A demo of our prototype can be found in Appendix 1.

6.1 Design Analysis

Our implemented design partially met our expectations. The mobile app was successfully built using React Native, allowing for a unified codebase that supports both Android and iOS platforms. Screen navigation, layout, and visual consistency were achieved across all major views. These elements worked well because React Native simplifies cross-platform UI development, and we were able to test on both emulators and physical devices to ensure consistency.

One of the key technical accomplishments was establishing Bluetooth connectivity with the Force Rack. We were able to scan for nearby BLE devices and initiate a connection with the hardware. However, we were unable to complete the data transfer functionality because we lacked access to the firmware documentation and the characteristic UUIDs required to receive data from the rack. This limited our ability to fully test and validate Bluetooth-based lift measurement.

If we had received access to the firmware information or documentation from the previous development team earlier in the semester, we could have completed the Bluetooth data handling feature. Despite this, the foundation we built in the React Expo app which can support Bluetooth once the necessary details are obtained.

7 Ethics and Professional Responsibility

Throughout our design there are multiple areas that were our main focuses, such as usability to our users. We took the time to consider different approaches to the app that users may have. Whether they are a coach, athlete, or carry some sort of disability like dyslexia or color blindness that could cause a user to have trouble understanding a screen full of words and numbers. This ties into the Ergonomics of Human-system interaction

which covers how we need to care about how people are interacting with our designs and products. This is why in collaborations, we talked extensively about each design, how it can be improved, and how the interaction with a user will change. The focus is solely based on how our design is improving a person's experience. A large part of our project also includes the security around users, ensuring that their data stays secure. This leans into information security management which is a large part of our project both for security and software. This means not leaving certain areas that could leak information or could be exploited. This goes into how we are all checking each other's code and keeping each other accountable in following our ethics and responsibilities as engineers on this project. This way we can also individually keep track of changes and ensure that each change that is made aligns with the ethics of the group.

Throughout the second semester we continued to keep the same ethics practices and awareness. We also have incorporated the same ethics when documenting our project and preparing to hand off the project to the next group. We want to make sure that we are following all the ethical practices to share client information and project details with the next team that we learned in our classes.

7.1 Areas of Professional Responsibility/Codes of Ethics

Areas of Professional Responsibility

Area of Responsibility	Definition	Relevance in SE Code of Ethics	Product Relevance
Work Competence	Take on jobs you can actually perform with your skill level. Delivered high quality work in timely manner	Product	Present a relevant timeframe to the client and disclose areas of skills that you don't currently have.
Financial Responsibility	Help client make well-informed financial decisions	Client and Employer	Present different security resources at varying price points to client that meets their needs
Communication Honesty	Communication truthfully with client in a way they can understand	Client and Employer	Client has no understanding of software, use we use many figures to help understanding
Health, Safety, Well-Being	Reduce risk of harm to users and client physically, fiscally, and mentally	Judgement	Product reduce's chance of users injuring themselves when working out
Property	Protect the data and property	Judgement	Securely store users' strength

Ownership	of users and client		and personal information in the database and during transmission.
Sustainability	Use sustainable practices to reduce environmental impact	Self	Design product to work on users' personal devices to negate need to purchase more devices
Social Responsibility	Be responsible and aware of the ethical, legal, and public consequences of your actions	Profession	Inform client of decisions and features that are not with public interest in mind

Table 7.1.1 Areas of Professional Responsibility. This table presents each area of responsibility according to McCormack and colleagues. It gives the description of each area, the relevance to our project, and the corresponding principle from the ACM Code of Ethics for Software Engineers [3].

One area of responsibility our team did well with is communication honesty. Our client had no understanding of software development, so in order to ensure they understood questions and proposed functionality, we used visuals and generalized terminology to explain things. This included athletic examples to ask questions about functionalities. Another way we ensured our client understands our communication is by giving them access to relevant documents. This allowed for better client understanding in order to receive the most appropriate responses and allowed truthful representation of our work that wasn't modified specifically to trick the client.

One area of responsibility our team could improve upon is work competence, specifically timeliness. Currently, our team is not to the point of our project that we had estimated to be at due to unexpected delays not factored into our original estimation for work completion. We are still delivering high quality work, just not at the rate expected. Because of this, we decided to turn to quality of documentation so the next group can use our information and current work to build upon.

Broader Context-Principle Pairs

	Beneficence	**Nonmaleficence**	**Respect for Autonomy**	**Justice**
Public health, safety, and welfare	Improves user strength and allows for a better quality of life	Prevents user injury while working out	Many types of exercises available for users to choose	(Neg) Product only available to those with access to gyms
Global, cultural, and social	Usable by people with color blindness and different technical knowledge	Promotes better relationship between trainers and trainees	Design allows for internal privilege structuring by the user's organization	Product available to people with any type of smart device
Environmental	Users don't have to purchase a specific device to use the product	(Neg) Weight rack uses heavy metals to make	(Neg) No way for users to make decisions about product's environmental impact	Design won't impact one environment more than another
Economic	(Neg) Takes away strength training and fitness jobs	Design would bolster the fitness economy	(Neg) Only one design option available to customers for purchase	Implementation doesn't not financially burden specific groups

Table 7.2.1 Four Principles with Respect to Broader Context Table. This table shows the relation between the broader context areas and the principles of Beauchamp with respect to the project.

One context-principle pair that is important to this project is global, cultural, and social justice. In this project's context, it refers to designing the program in a way such that anyone looking to improve their strength and fitness can use the project regardless of what smart application they have. Many similar products only work on specific devices which diminishes the amount of people that can access the product. We are ensuring that we meet this standard by creating an application that is compatible with most devices including different devices' OS versions, orientations, and dimensions as well as android devices.

One context-principle pair that will be lacking with this project is economic respect for autonomy. There will only be one software product available for customers to purchase. While this restricts the ability to have specific features available for those who want to use the product in specific ways, it also makes it so that more complex features are not hidden behind additional paywalls. The customer doesn't have to struggle to decide which specific product would best suit their needs and possibly waste money on something that doesn't work for them. Once the customer purchases the product, they have access to all features of the software.

7.3 Virtues

The three virtues our team came up with were: cooperativeness, communication, flexibility. We believe that these virtues represent our team for our constant communication with each other and our client as well as our cooperativeness to work together on our projects and bounce ideas off of each other. Our flexibility resides in our ability to get our work done and find different solutions when our original idea doesn't work or we are missing pieces like access to our database. These qualities have driven our team from the beginning and are the driving force behind our team working so well together.

Team Member Virtue Assessment

Molly's Virtue	The virtue I have demonstrated through the senior design work thus far is cooperativeness. I feel I have been able to work in the group well and have done what is best for the team most of the time. This virtue is important to me because I value teamwork and I believe that if everyone is cooperative then their group will be more successful. One virtue I would like to improve on is courage. I want to improve on expressing my ideas and possible solutions with more confidence. I lack the courage to believe that my ideas are strong. This is important to me because I believe that if everyone uses courage within the group to share their ideas, the final product will be significantly stronger than without everyone's input.
May's Virtue	The virtue I demonstrated so far is cooperativeness. This virtue is important to me because the project involved combining two separate applications. This requires a lot of work to be done, and I believe that good teamwork between all members of our group is needed for the project goals to be achieved. I have been able to demonstrate cooperativeness by attending all team meetings, communicating effectively with the team, and contributing to or delivering required work for our project.

	A virtue I would like to improve on is perseverance. This is important to me because I would like to work through any challenges or issues that come up with the project more consistently. Even though I've made some contributions to the project, I think I could better demonstrate this virtue by improving my time management. This would let me make more consistent efforts in making progress on the project even when my schedule gets busier over time.
Paige's Virtue	Cooperativeness is a virtue I demonstrated. This is important to me because I strive always maintain a good relationship with my teammates. Good relationships help create the best end product by reducing team strife and by encouraging every member to work on subjects they excel in or have a passion for. I have demonstrated this by engaging with my teammates about topics outside of the project, helping out when possible, and doing my part. A virtue that I would like to improve on is accountability. I haven't been very accountable when it comes to coding. I can improve this by setting stricter deadlines for myself on code tickets and by setting aside personal time every week to code. This is important to me because I don't want my team members to lose respect or feel as though they can't count on me.
Tessa's Virtue	The virtue I believe I have demonstrated the most is cooperativeness. This virtue is important to me because I believe that the basis for a successful project starts with cooperativeness. It is essential to get everyone on the same page and working well together. I have demonstrated this virtue by trying to make sure everyone is on the same page and intentionally setting up meetings twice a week for each team member to bring things up to work through together as a team. One virtueI would like to improve on is humility. Throughout this semester, I have been voicing my opinions and thoughts without taking a step back to consider other alternatives or options. I haven't been the best at stepping to the side to allow my teammates to step up. I can improve this by reminding myself when we go to our team meetings to wait before I speak to allow my team members the time to gather their thoughts and present their ideas.
Steven's Virtue	The virtue I have demonstrated during our senior design work so far is time management. This virtue is important to me because good time management helps me meet deadlines and accomplish my goals for the project. I have demonstrated this by consistently attending weekly meetings, staying organized, and contributing to the team's assignments on time. My ability to manage my time effectively ensures that I can deliver quality work and remain reliable for my teammates.

	One virtue I would like to improve is communication. I feel that sometimes I don't express all my thoughts or ideas due to a fear of being wrong or that my opinion might be irrelevant. However, I realize that open communication is essential for teamwork and can lead to better solutions for the project. I would like to contribute more actively to team discussions by sharing my ideas, asking questions, and providing feedback. I can demonstrate this virtue by preparing my thoughts ahead of meetings, building confidence, and reminding myself that every opinion matters.
Lexie's Virtue	The virtue that I have been demonstrating so far in the senior design project is trust. It is important to me to have trust on a team because that means I know I can rely on my teammates to do well and get their work done and that they can rely on me. With a project at this scale and with the way we have split up the work and delegated tasks, there are just some parts of the project that I am not involved in. This means I am trusting others and their abilities with the things that I am not working on. The virtue that I have not been demonstrating and that I would like to improve upon in the upcoming semester is self-discipline or passion. I am not as interested in the project as I thought I would be, so I did not work on it as much or as hard as I should have. Having the self-discipline to work on a project, even when I do not want to, is important to me because I do not want to let myself, my teammates, and our client and advisor down. Passion is also important to me, because I want this project and this experience to be something that I look back on positively in the future. Being passionate about something also makes it a lot easier to work on it than if you are not.
Elicia's Virtue	I believe that a virtue that I have displayed is flexibility as working on the cybersecurity team has led to some drawbacks through limited access to controls through the database, which is an important part of our education at Iowa State to protect. Thus to fulfill our responsibility to secure our product we had to be flexible in our approach to find the best solution through a different avenue. The virtue I want to work on next semester is accountability as with cybersecurity, we do not upload to the git and therefore don't have a physical way to hold ourselves accountable. Thus a focus for myself is to make sure that cybersecurity will be something that we can physically show our work beyond research. This can be done by setting up the two-factor authentication that everyone will be able to see.

Table 7.3.1 Team Member Virtue Assessment. List of each team members' well demonstrated virtue and poorly demonstrated virtue.

8 Conclusions

8.1 Summary of Progress

The overall goals of this project were to migrate the two existing applications we developed into a single React app, implement the Bluetooth connectivity and data transfer between the rack and the app, and create documentation covering security solutions and a user manual.

Our team was successful in designing and implementing the user interfaces of the True Force application using React Native. Using physical devices for development testing, we also were able to establish a Bluetooth connection between the weight rack and and our application. We were able to identify the custom Bluetooth services and characteristics, but live data streaming couldn't be achieved because of the lack of access to the rack's firmware and the original application's source code. During development, various attempts were made to trigger the data transfer of the rack's lift data. The attempts included characteristic write attempts and BLE packet logging using Android snooping tools. These efforts only clarified which characteristics were involved in a data transfer. Further development for implementing Bluetooth functionality was limited by the restricted access to the preexisting system's documentation and code.

Our goal of converting to a single React application was achieved, but only partial progress was made on the second goal of implementing the Bluetooth functionality due to live data streaming remaining unimplemented.

Our goal of Security was to create a well protected system, however, through the challenge of having no access to the backend, this goal became having a well thought out security plan for future use.

Our team was successful in finding well developed tools that will connect to a cloud database for further protection. We used past knowledge from industry experiences to create plans in different situations such as documentation of user's use of the system and interaction of the database, while also looking to have secure logins through two-factor authentication. We created two plans one with Microsoft Azure and Amazon Web Services of what exact tools should be implemented into the system each with an approximate cost value of each. We created a more in depth guide of Azure as it was the top choice of the client as we broke down the different services further while providing a guide of how to set up a cloud database for use for future teams to reference when having access to the backend.

Overall, while unable to directly provide security to the system, we have full faith in the next group to implement our security design to effectively secure the app and user's information and connect the database to the app.

8.2 Value Provided

Our client was very pleased with our design. It met most of our users' needs despite our application not. Our design fixed the client's concerns with broader device availability by implementing the application on React Native instead of hardcoding it for a specific tablet. It addressed concerns on privilege levels and functionality for different users by limiting features and defining privilege levels. Our design adds security to the database where there was very little before so that the client has more control by moving to the Azure Cloud where the client can customize it as he wants. It also allows for a customer to set up their account and system themselves without our client having to do everything through the create account button and bench configuration.

Users can create and manage exercises, teams, and user accounts. Because we were not able to parse the data from the Bluetooth stack, user's can't see statistics about their lift. The lack of information regarding the backend and database meant that we were unable to implement data saves or recalls. The setbacks we had around backend, database, and bluetooth documentations limited the usefulness of the project in its current state, however, what we delivered met our client's expectations and our design addresses the majority of the problems addressed.

8.3 Next Steps

The next step for this project is for it to be passed to another senior design team to keep developing the product. The main functionality that we would like to see implemented next is the Bluetooth data transfer and the secure cloud database set up. We did a lot of the foundational work for this to be possible. If the next team is able to complete these two things, the app would be a lot closer to being used by clients of True Force Technologies.

After that, teams could work to make the app and rack be compatible with being used in commercial gyms, rather than just privately by college athletics coaches. This would require changing the functionality of the app to allow for private users, not only on coaches and their teams. There might also have to be some restructuring in the database for this to be accomplished.

9 References

List technical references and related work / market survey references. Use a professional citation style (e.g., IEEE). See link:
https://ieee-dataport.org/sites/default/files/analysis/27/IEEE%20Citation%20Guidelines.pdf

[1] *Catapult One Team | Football - Catapult One*. [Online]. Available: https://one.catapultsports.com/us/catapult-one-team-football/. [Accessed: Dec. 3, 2024].

[2] *CoachMePlus | Human Performance Software*. [Online]. Available: https://coachmeplus.com/. [Accessed: Dec. 3, 2024].

[3] IEEE Computer Society | Code of Ethics for Software Engineers [Online]. Available: https://www.computer.org/education/code-of-ethics. [Accessed: Dec. 5, 2024]

[4] IEEE P3411 | Standard for Smart Identification in Internet of Things [Online]. Available: https://standards.ieee.org/ieee/3411/11408/#:~:text=The%20standard%20provides%20an%20application%20framework%20to%20facilitate,IoT%20environments%2C%20leveraging%20existing%20protocol%20standards%20where%20appropriate. [Accessed: Dec. 3, 2024]

[5] IEEE Standards in Everyday Life. [Online]. Available: https://www.youtube.com/watch?v=2fgnPdBTTZ8. [Accessed: Dec. 3, 2024].

[6] ISO 14000 family | Environmental management. [Online]. Available: https://www.iso.org/standards/popular/iso-14000-family. [Accessed: Dec. 3, 2024]

[7] ISO 27001 | Information security management systems [Online]. Available: https://www.iso.org/standard/27001. [Accessed: Dec 3, 2024]

[8] ISO 636 | Language Code. [Online]. Available: https://www.iso.org/iso-639-language-code. [Accessed: Dec. 3, 2024]

[9] ISO 9241 | Ergonomics of human-system interaction [Online]. Available: https://www.iso.org/standard/63500.html. [Accessed: Dec. 3, 2024]

[10] *Output Sports | Test, Monitor & Program Athletic Performance*. [Online]. Available: https://www.outputsports.com/. [Accessed: Dec. 3, 2024].

10 Appendices

Appendix 1 – Operation Manual

The use manual with step by step instructions to set up the TrueForce Rack.

https://docs.google.com/document/d/1YPoKMjkhgaXNtGWZf4DH3L-Y2hhXrGFCbQR1n8AKnug/edit?usp=sharing

The user video to demonstrate how to set up the TrueForce Rack.

https://iastate.box.com/s/chdvutn4kh8lmh6xgf3548p7g60v5s1g

The prototype of our current application.

https://drive.google.com/file/d/1imUHH4w35A81DGcZ08MWQ7g1Dy_UZ9eq/view?usp=drive_link

The operation manual with steps to deploy and test/debug the TrueForce application on physical devices.

https://docs.google.com/document/d/19e3vr7THkoarOSHhkqPakFQSeOmh9VrRxisbU0MRGhU/edit?usp=sharing

Appendix 2 – alternative/initial version of design

The alternative version of the design is our first semester design that featured two applications, one Android and one iOS. This version was scrapped at the end of first semester after our faculty review panel. We received feedback on the challenges of the upkeep of two separate applications and decided to make the switch to one application to make future development easier.

Figma Design:

https://www.figma.com/board/ttljB6CAPMEbSl4OI8j8TE/491-Design?node-id=0-1&t=xomI3pnw4TckS9zS-1

Semester 1 iOS Prototype:

https://drive.google.com/file/d/1O_qv312C8tNsv8W-e1dckdsxbjmrvbRf/view?resourcekey

Semester 1 Android Prototype:

https://drive.google.com/file/d/1hu3ykGSVb-uX1I95UaQ29YdHcpC9CAkE/view?resourcekey

Appendix 3 – Other considerations

Sport bench annex.pdf

Application contract annex for the original application. It contains screen design and navigation information as well as technical specifications for the application.

CyberSecurity Database Solution

Report of all security solutions research, including database setup, break down of products, and cost analysis

Security Solutions

Report of security scenarios and solutions considered for the project.

Security Considerations

List of security points to check for threats and solutions for those vulnerabilities.

Appendix 4 – Code

https://git.ece.iastate.edu/sd/sdmay25-10#

Appendix 5 – Team Contract

Team Members

- Tessa Lanzel
- Molly Rooney
- May Edel
- Paige Schneider
- Alexandra Rauer
- Elicia Baranowski
- Joshua Steven Chiang

Required Skill Sets for Your Project

- Swift Development
- Java Development
- React Native Development
- BLE (Bluetooth Low Energy)
- Embedded software knowledge
- Communication
- Teamwork
- Problem Solving Skills
- Security knowledge

- Security Implementation
- Microsoft Azure knowledge
- Cloud database knowledge

Skill Sets covered by the Team

Swift Development

- Tessa Lanzel
- Joshua Steven Chiang
- May Edel

Java Development

- Molly Rooney
- Paige Schneider

React Native Development

- May Edel
- Paige Schneider
- Molly Rooney
- Joshua Steven Chiang
- Tessa Lanzel

BLE (Bluetooth Low Energy)

- May Edel
- Paige Schneider
- Molly Rooney
- Joshua Steven Chiang
- Tessa Lanzel

Embedded Software Knowledge

- May Edel
- Paige Schneider
- Molly Rooney
- Joshua Steven Chiang
- Tessa Lanzel

Communication

- May Edel
- Paige Schneider
- Molly Rooney
- Joshua Steven Chiang
- Tessa Lanzel
- Elicia Baranowski
- Alexandra Rauer

Teamwork

- May Edel

- Paige Schneider
- Molly Rooney
- Joshua Steven Chiang
- Tessa Lanzel
- Elicia Baranowski
- Alexandra Rauer

Problem Solving Skills

- May Edel
- Paige Schneider
- Molly Rooney
- Joshua Steven Chiang
- Tessa Lanzel
- Elicia Baranowski
- Alexandra Rauer

Security Knowledge

- Elicia Baranowski
- Alexandra Rauer

Security Implementation

- Elicia Baranowski
- Alexandra Rauer

Project Management Style Adopted by the team

Our team took more of an Agile approach to complete this project. We utilize GIT to create issues and to keep track of progress. The development team has set their own timelines within the main timeline to complete each task. We also have allowed for continuous improvement when feedback is given by our client and/or advisor. Since we use the Agile method we have had small milestones throughout the project to ensure the goals are being met. For example, the beginning of second semester was filled with recreating the app with react. The team had never used react before. Once they learned how to implement using react, they started developing each screen one by one instead of all at once. This shows that agile development is being used because instead of taking on the whole project, they separated into steps that are more manageable.

Individual Project Management Roles

- Tessa Lanzel - Team Product Leader / Frontend Development Co-Master
- Molly Rooney - Team Communications Leader / Frontend Development Co-Master
- May Edel - App Development Co-Master

- Paige Schneider - App Development Co-Master
- Alexandra Rauer - Cyber Security Co-Master
- Elicia Baranowski - Cyber Security Co-Master
- Joshua Steven Chiang - Software Testing Master

Team Contract

Team Members:

1) Tessa Lanzel 2) Molly Rooney 7) Joshua Steven Chiang

3) May Edel 4) Paige Schneider

5) Alexandra Rauer 6) Elicia Baranowski

Team Procedures

1) Day, time, and location (face-to-face or virtual) for regular team meetings:

   Semester 1

   - Monday 1:00 - 2:00 p.m. with advisor and/or client at SICTR
   - Wednesday 1:00 - 2:00 p.m. team meeting at SICTR

   Semester 2

   - Wednesday 10:00-12:00 p.m. team meeting in design lab
   - Wednesday 11:00-12:00 p.m. with advisor and/or client in design lab

2) Preferred method of communication updates, reminders, issues, and scheduling (e.g., e-mail, phone, app, face-to-face):
   - Preferred method of communication will be snapchat or face-to-face
   - Utilize email to communicate with faculty advisor and client
3) Decision-making policy (e.g., consensus, majority vote):
   - Group Consensus
4) Procedures for record keeping (i.e., who will keep meeting minutes, how will minutes be shared/archived):

   - Molly Rooney will be keeping meeting minutes and will be putting them in the Meeting Minutes folder within the shared Google Drive.

Participation Expectations

1. Expected individual attendance, punctuality, and participation at all team meetings:
   - All team members are expected to attend all Monday and Wednesday meetings. If a team member cannot attend a meeting, they must communicate that beforehand.
   - All team members should be on time for meetings.

- All team members should be engaged in meetings.

2. Expected level of responsibility for fulfilling team assignments, timelines, and deadlines:
    - All team members are expected to participate in completing all assignments during the timeline and before all deadlines.
3. Expected level of communication with other team members:
    - All team members are expected to communicate when they need help.
    - All team members are expected to update the team during meetings about progress.
4. Expected level of commitment to team decisions and tasks:
    - All team members are expected to participate in the decision making process and all tasks.

Leadership

1. Leadership roles for each team member (e.g., team organization, client interaction, individual component design, testing, etc.):
    - Tessa Lanzel – Team Software Product Leader and Client Interaction
    - Molly Rooney - Team Communications Leader
    - May Edel – Component Design Leader
    - Elicia Baranowski – Cybersecurity Research and Product Co-Leader
    - Alexandra Rauer – Cybersecurity Research and Product Co-Leader
    - Joshua Chiang Fung – System Analyst Leader
    - Paige Schneider – Development Leader
2. Strategies for supporting and guiding the work of all team members:
    - Communication is key, if a team member has something blocking them from being able to complete their task, reach out and let the rest of the team know
    - All members will work together to assign tasks and roles that are manageable and realistic to what can be accomplished
3. Strategies for recognizing the contributions of all team members:
    - Each team member will identify the contributions they have made for the week at the team meeting and on the weekly report
    - No hidden work, all team members will inform the team about work completed and upcoming planned work

Collaboration and Inclusion

1. Describe the skills, expertise, and unique perspectives each team member brings to the team.

- All team members have previous experience with app development from COM S 309.
- We have five Software Engineering students and two Cyber Security Engineering students.
- Two of the the team members have prior experience in Applications Development

2. Strategies for encouraging and support contributions and ideas from all team members:
    - Each team member brings something different to the project so we are leveraging everyone's talents to complete the project.
    - All team members will be accountable to make sure everyone's voices are heard in discussions and that no idea goes unnoticed.
3. Procedures for identifying and resolving collaboration or inclusion issues (e.g., how will a team member inform the team that the team environment is obstructing their opportunity or ability to contribute?)

    - Each team member should be open and honest about their feelings in the group and each member of the team should be respectful of one another.
    - If one team member's task is obstructed they should move to a different task until the first task is no longer obstructed.
    - Team members will identify why their task is being obstructed and communicate what help is needed to move past the obstruction

Goal-Setting, Planning, and Execution

1. Team goals for semesters:

    semester 1
    - To get started on 2 working applications (IOS and Android)
    - To work well together for the entirety of the project
    - To complete all work in a timely manner and with the most effort possible

    semester 2
    - Migrate the two applications into one React application
    - Get a bluetooth connection between the rack and application
    - Document security solutions and user manual
2. Strategies for planning and assigning individual and team work:
    - Each in class assignment will be completed by every member of the group
    - Each project based assignment will be distributed among the team members based on their individual strengths and the scope of the task
3. Strategies for keeping on task:

    - We will have a tentative schedule for project tasks that will be followed throughout the year.

Consequences for Not Adhering to Team Contract

1. How will you handle infractions of any of the obligations of this team contract?
    - We will have a conversation with the team member in question to resolve the issue.
2. What will your team do if the infractions continue?
    - We will get our advisor and/or the professors of the class involved to understand why this has become a blocker.

****

a. I participated in formulating the standards, roles, and procedures as stated in this contract.
b. I understand that I am obligated to abide by these terms and conditions.
c. I understand that if I do not abide by these terms and conditions, I will suffer the consequences as stated in this contract.

1) Molly Rooney DATE 9/18/2024

2) Tessa Lanzel DATE 9/18/2024

3) Paige Schneider DATE 9/18/2024

4) Elicia Baronowski DATE 9/18/2024

5) May Edel DATE 9/18/2024

6) Alexandra Rauer DATE 9/18/2024

7) Joshua Chiang Fung DATE 9/18/2024