**[Title Here]**

**Introduction**

Without proper record keeping and planning, weight training can feel like an exercise in futility. A couple suboptimal sessions can easily cause an athlete to lose sight of the big picture if no road map is maintained. To make matters worse, keeping track of a logbook at the gym can be a workout itself, and punching exercise details into the notepad application of a smartphone is a chore. Other smartphone based options are available. However, they often come at a price and are difficult to use. (Adam: I’m totally pulling this out of my ass, I’ve never used one.) An inexpensive workout tracker application focused on maintaining records with little user input would solve these issues. This workout tracker, henceforth to be referred to as GymTrackR, will also serve as a platform upon which additional solutions for the trainer and athlete can be implemented. This proposal will elaborate on (Adam: Maybe break this sentence down into a bulleted list) the motivations behind GymTrackR, the base functionality of GymTrackR, the GymTrackR user experience, and finally the growth potential of GymTrackR. The proposal will close with a summary of topics covered and provide references where appropriate.

**Motivation**

**Project Summary**

With the rising trends of people changing their diets and exercise routines by purchasing gym memberships to get in better shape and improve personal health. Technology has become a major asset to a person's workout routine, whether it be for music or to keep track of workout routines or both. The goal of this project is to create a mobile app on Android that is simple to use and user friendly, that a user can setup workout routines and keep track of them.

**Project Details**

*Project Scope and Elements*

GymTrackR will be an application intended for use on an Android platform. At its core it will be a tool used for tracking workout statistics with as little user input as possible. Most time spent in a tracker application is from navigating the applications menus. To this end a user will be able to plan what exercises they wish to do well in advance. This will cut down on time spent in the UI at gym. Other optimizations will be in place to help streamline the process if the user has to deviate from the specified plan, or is simply keeping track of their progress without a plan. Other features will be added time permitting.

The application will be broken down into three primary pieces the GUI, the application logic, and the database. Little will be required to attack these problems, outside of man-hours, github, and Android Studio. (Adam: Not sure what else we’re using.)  
  
*Implementation Issues and Challenges*

While conceptually simple, GymTrackR will provide some definite challenges. The first and foremost of which will be familiarizing the team with the technologies required for the project. The team can quickly familiarize itself with github. However, the mastery of android programming required to implement GymTrackR’s GUI and application logic will take some time to develop. Programming challenges are difficult to foresee, but are absolutely expected. Fortunately expenses will be quite low as all of the tools required to create GymTrackR are free.  
  
Additional challenges can be expected to arise on the teamwork frontier. The team itself is composed primarily of novice programmers with little experience on team projects. Certainly no member has noteworthy experience in project management. In light of this, creating a unified vision of the project may take additional time. It will also take additional time to view design tasks with clarity, and delegation of these tasks is likely to occur in a suboptimal manner.

*Deliverables*

The end of the project will yield GymTrackR, a workout planning and tracking application with an elegant, intuitive user interface. GymTrackR in itself will be composed of a Start Screen, a Workspace, an Exercise Browse Menu. Elaborations of each follows.  
  
**Start screen**  
At application startup, the user will be presented with several options. They will have the choice to select “Workout” or “Plan a Workout”. Each of these options will load the workspace and put it into either “Plan” or “Workout” mode. “Plan” mode will save workspace data as a plan, whereas “Workout” mode will save workspace data as history.  
  
When “Workout” is selected the user will be prompted to select “With Plan” or “Without Plan”. If “Plan a Workout” is selected the user will be prompted to select “Edit an Existing Plan” or “Plan a New Workout”.   
  
“Plan a Workout > Edit an Existing Plan” and “Workout > With Plan” will load a plan to the Workspace.

“Workout > Without Plan” and “Plan a Workout > Plan a New Workout” will present the user with a blank Workspace.

**Workspace**Depending on the user’s selections, the user will be presented with either a blank or populated workspace. A populated workspace will consist of a vertical list of items. Each item will represent a part of the user’s workout, and will have relevant information listed on it. Exercise name, repetitions, weight, time, etc. Additionally, each item will have a vertical bar on the left used to drag the item, and an “x” on the right that when selected removes the item. Tapping an item will “select” it. Multiple items may be selected. Items may be rearranged by dragging them via the bar, thus modifying the workout order. Tapping and holding an item will open the “Workout Edit Menu.”   
  
The workspace will contain six buttons at the top: “Search” “Cardio” “Rest” “By Muscle Group” “By Equipment” and “+”.  
  
“Cardio,” “By Muscle,” and “By Equipment” will open up menus used to search GymTrackR’s database of exercises. An optimal methodology is still being discussed as to the specific manner with which the user will refine the search. However, it has been agreed that exercises ought to be listed by order of use, then alphabetically. The user can select the desired exercise from the list to open an “Edit Exercise Menu.” “Search” will allow the user to type in an entry to search for it.

“Rest” will be used to specify a rest period and will add a rest item to the workspace. This item can be held to open an edit menu. Additionally, it will mark the end of a circuit (a sequence of exercises performed in rapid succession.) This will allow for easy distinction between traditional weight sets, and the various forms of circuit training.  
  
“+” will be used to perform quick add functionality aimed at saving the user’s time. If an item/items are selected on the workspace, selecting “+” will duplicate the selected item/items. If no items are selected “+” will add a copy of the last added exercise.

**Edit Exercise Menu**  
The Edit Exercise Menu will allow the user to edit information relevant to the selected exercise such as repetitions, weight, and duration. The user will then have the option to hit a “+” button to add the exercise to the Workspace or to submit changes if the menu was entered via workspace item.   
  
If the user is in workout mode, the exercise data will be saved to history. If the user is in plan mode, the exercise data will be saved as part of a plan. This will keep useless information out of GymTrackR’s history.   
  
Additionally, the Edit Exercise Menu will display a scrolling menu of the user’s past exercise history with the exercise they are about to add.   
  
If this menu is entered via an item in the workspace, the user will be able to swipe left to the next item in the workout, or swipe right to the previous item in the workout. This should greatly streamline the usage of the app when working out from a plan.  
  
Other potential project products include:

QR scan functionality  
 To streamline the user experience further

History Graphical Data Display  
 To help the user visualize their fitness roadmap  
 Shared Workout Plans and Workout Reports  
 To allow GymTrackR to be a tool used for trainers to work with clients

Workout Explanations and Demonstrations  
 To allow GymTrackR to further the user’s training expertise

*Timeline*

*Budget*

**Conclusion**

[Conclusion Here]

**References**