University of Arkansas – CSCE Department
Capstone II – Final Report – Fall 2021

Phone Phit - Workout App

Jack Chalmers, Jackson McCauley, KimMarie Counts, Brad Greenway

Abstract

Phone Phit is a workout app that tailors your workout based on the equipment you have available, the time you want to spend working out, and the part of your body you want to focus on. A list of exercises is generated for you based on what you input during the questionnaire.

1.0 Problem

People have troubles exercising. Whether it is a lack of time, a sense of self-consciousness, feeling too tired, or finding exercise boring that causes a person not to exercise, we still all need to get physical movement into our daily lives. Early humans got their exercise by hunting and foraging, but when our modern lives allow for us to sit on the couch and order take in, we don't get enough movement. Not moving can cause a prolonged state of lower energy resulting in poor insulin absorption (diabetes), higher risk for osteoporosis, muscle atrophy, slower metabolism, and heart disease. These features originally helped early humans conserve energy while not on the hunt. Exercise for early humans was meant for survival. Scientists theorize that no early human would have gone on a run just to go on a run due to the extra energy expenditure. This has caused modern day humans to not want to exercise. Getting some exercise every day can help prevent these undesirable affects and help manage your weight, improve your ability to do daily activities, increase your endorphins, reduce stress, bring you back into the moment, and increase your chances of living longer.

2.0 Objective

The objective is to create a webapp to help people exercise daily.

3.0 Background

3.1 Key Concepts

Angular – This technology is a development platform which is built on TypeScript. Angular includes a component-based framework for building scalable web applications. This platform also offers a collection of well-integrated libraires that cover a great mixture of features which will benefit in creating our application. In addition to the other features there will be a wide

variety of developer tools to help with developing, building, and testing the code that is needed.
Firebase – The Firebase technology is a Backend-as-a-Service which provides a variety of tools and services. Firebase's key features are authentication, real time database, hosting, test lab, and notifications.
Ionic Framework – Ionic is an open-source UI toolkit for building mobile and desktop applications using web technologies. This technology also integrates with popular frameworks like Angular which is the development platform we have chosen. Ionic centers there work on the frontend UX and UI interactions of an application.

3.2 Related Work

There are a lot of resources about our project area which is the problem. These other applications can cause confusion to the client since they would not be able to find what is best for their situation. Our application will consider what they are wanting to do and what they have available to them. Another problem is that other platforms cost money right out of the gate which may cause clients to be deterred from their goal if they have to pay for something before they start.

4.0 Design

4.1 Requirements and/or Use Cases and/or Design Goals

The requirements for this project are as follows:

- Given a list of constraints by the user, the application should return a set of possible recommended workouts from a database. These constraints may tentatively include:
    - The equipment the user has on hand
    - The amount of time that the user must exercise
    - The area of body or muscle group the user wishes to exercise
    - The user's rough experience level, to keep exercises at a level of strenuousness that the user can handle.
- While not essential to the application's primary use case, there are several secondary use cases and features that could be included if time allows. These secondary use cases include:
    - Being able to select a set of exercises as favorites that can be quickly retrieved and viewed.
    - Similarly, being able to construct routines and run through them using the application as a timer.
    - Being able to log total exercise time and time spent per exercise/routine as a means to track progress.

4.2 [High Level / Detailed] Architecture

The application will use a database to store exercises according to criteria relating to their performance requirements and difficulty. These criteria will look similar but will not necessarily exactly match the input data. Possible places where these two lists could differ could be if two exercises are unadvisable to be performed together. The user has no reason to search with that criterion, but it would be necessary to be kept in the backend.

The primary functionality would be contained in the user submitting a form with all the constraints they desire to filter exercises by what they can perform. The details of this form will depend on the field but will mostly be a dropdown list of options the user can select from or a field of checkboxes. Exercises will also be able to be queried by name, but this should largely be to recall exercises already given by the database, or as a tool to check the requirements for a specific exercise the user is interested in.

For example, time will be a number that the user inputs. Equipment on hand and body area will be sets of checkboxes, and experience level will be a dropdown list.

The application will take the form and use its contents to query the exercise database and display these results to the user. Also displayed will be details on their performance, with diagrams and potentially an example video showing the exercise being performed.

Stepping away from primary functionality into the range of stretch goals, these queries will also be able to be logged into a set of favorites or individual routines. This process will be like how services like YouTube or Spotify handle sorting videos or songs into playlists, and will allow the user to name, edit, and specify custom routines.

This would work as a separate option from of the launch screen, under the query function. Routines would be listed in a manner of ways editable by the user, such as by name, date created, length, or more esoteric groupings like area of body, length, or difficulty.

From this screen, routines would renamable, delete-able, and if time allows, playable. Once a specific routine has been selected, the user would be able to move and edit the individual workouts that have been added, as well as remove or add more exercises.

About ‘playing’ a routine, the app would be able to run through the individual exercises on a routine by displaying the instructions and serving as a rep-counter and/or timer depending on the needs of the exercise.

This play function would be pause-able at any time, and an individual exercise had its instructions repeated, restarted, or the entire routine restarted or exited.

Finally, the subject of logging time. The application would be able to track the time spent playing each routine and run a few basic statistics on the data, like approximate calories burned or which routine is most used. Alternatively, performing the routine by themselves was preferred to using the play function, the user could input the approximate amount of time spent on each exercise and get roughly the same data. The application would also be able to log which dates exercises were performed on and help keep the user to a schedule that they would be able to set.

4.3 Risks

Risk	Risk Reduction
Physical injury while conducting an exercise recommended by the application	While the application has no control over what the user does with the exercise information once it has been presented, ample warning can be given. For instance, exercise is potentially harmful if done in excess, and reminding the user to take breaks. Additionally, cautioning the user to start small with weights and routines, as it is better that their first few sessions are too light than too heavy.
Unhealthy expectations associated with the body	The application would take strides to warn users that exercise is not the end-all-be-all for weight loss or muscle building, since other factors like diet play such a large part. The

	application would warn users that they might see less improvement than they would like, even with a frequent schedule of exercising.

4.4 Tasks –

- Collecting a list of exercises. This could either be a independent search, or it could take the form or sifting through repositories of exercises and comparing data.
- Researching the requirements, effects, and general exertion level of each entry.

1. Design of database schema
   This will be a relatively simple task and should largely follow from the requirements and design set out above. All that will be required is looking at the data and our requirements, then deciding what each field needs to be.
2. Sorting of entries into database
   Following the first two tasks, this will mainly be bookkeeping task rather than a programming one and would just see us methodically sorting each entry according to the fields set out in step 2. It will largely be grunt work, and thus the time spent on it should not be variable, seeing as each entry should take roughly the same amount of time.
3. Implement form and query system
   This step will consist of developing the form in Angular and wiring it to the database through use a query language such that the database can be searched.
4. Implement the results screen
   This step will consist of designing and implementing the screen where the results are displayed, including sub-screens for each individual result with all their data.
5. Design routine screen and implement sorting functionality (Secondary)
   This step consists of implementing the menu for routines and the functionality for allowing users to create, add to, and manipulate routines.
6. Implement play functionality (Secondary)
   This step consists of implementing the functionality that would allow the app to work through a routine and display instructions, then serve as a timer for the individual exercises in a routine.
7. Implement logging functionality (Secondary)
   This step would consist of the allowing the application to log usage data, and the implementation of basic statistic keeping, to show the user a rough approximation of progress made. This could also include schedule functionality, allowing the user to keep themselves more closely to habit-forming progress.

4.5 Schedule – Create a schedule which reflects the proposed timeline for each of the tasks listed in section 4.4. Each task should be scheduled, and the schedule should have more detail than the example below (e.g. break timeframes down into weeks or months).

Tasks	Dates
1. Get the team up to speed on	9/13-9/16

angular development	
2. Team github repo set up	9/16
3. Hello World displayed	9/16
4. Firebase up and running with Hello World	9/17-9/18
5. Component work	9/18-11/1
6. Combine individual components to the main project	11/2-11/8
7 Port angular to mobile using ionic	11/9-11/20
8 Bug Testing	11/20-11/30
9 Final Report	12/1-12/13

4.6 Deliverables – Give a thorough listing and description of each item which will be submitted with your final, working project. Each major component should be described. The below is just an example list which should be replaced with your own.

- Design Document: This is going to be a further breakdown of how each angular component will be broken down. We will investigate using a visual designer such as figma as necessary
- Initial angular page working with Hello World displayed
- Delegate out components and algorithms as needed
- Framework for scrollable carousel that will have the data populated based on filters
- Polishing up the visual design of the web app
- Port the angular app to mobile by using ionic
- Final Report

5.0 Key Personnel

Jack Chalmers – Chalmers is a senior Computer Science major in the Computer Science and Computer Engineering Department at the University of Arkansas. He has completed relevant courses and only has electives left to take besides capstone. He has completed two previous internships. One at a local NWA branch of a technology consulting company where he was given basic C# tasks to complete and was eventually banished to WordPress websites. The other internship he completed was at a Chicago cybersecurity startup company where he helped craft the user console from the ground up using angular to create reusable components to be dropped in wherever needed. He has accepted a return off there as a junior software dev for after he graduates this coming fall. He is the team leader and will be leading the team through creating an angular app focused on tailoring your own workout experience.

Jackson McCauley – McCauley is a senior Computer Science and Computer Engineering major in the Computer Science and Computer Engineering Department at the University of Arkansas. He has completed all relevant courses and only has one other class besides capstone left. He has been a Research Assistant at the University of Arkansas for Jia Di since Fall 2020. The research he is a part of is in trustable digital hardware design and analysis at all levels.

KimMarie Counts – Kimmarie Counts – Counts is a senior Computer Science major in the Computer Science and Computer Engineering Department at the University of Arkansas. She has completed all required courses except for Capstone II and Formal Languages to graduate. Counts has taken Computer Graphics, and Artificial Intelligence for her electives and is currently enrolled in Mobile Programming. She has been a Teaching Assistant at the University of Arkansas for Yauri Peng since Spring 2021 where she has led a lab for CSCE 2114 Digital Design.

Brad Greenway – Greenway is a senior Computer Science and Computer Engineering major in the Computer Science and Computer Engineering Department at the University of Arkansas. She has completed or is currently enrolled in all relevant courses. In addition to being a student, she is currently working as a Teaching Assistant for several sections of Discrete Math.

6.0 Facilities and Equipment

This project is going to require each team member to have a laptop with a developer environment set up that can run a light framework such as angular and firebase. These are simple things that can be worked on in any text editor and run locally through the command line or live through the firebase page.

7.0 Results

To use the app, a user clicks through the splash screen, and is presented with the questions about exercise specifics. These questions each take place on separate screens that are cycled automatically, to reduce clutter. Time to Perform and Difficulty are sliders, Required Equipment has the user select any equipment they have on hand from a list, and Part of the Body functions the same way, using a list of supported body parts. Difficulty is cumulative, meaning that if selecting a higher difficulty will also give results for lower difficulties.

After answering the last question, the app automatically presents the matching exercises. Exercises are presented in a randomized list, and the user will begin the workout that the app has created for them.

8.0 References

[1] Why Is Exercise Hard? | Harvard Magazine, https://www.harvardmagazine.com/2016/09/born-to-rest

[2] Exercise and Stress: Get Moving to Manage Stress – Mayo Clinic, https://www.mayoclinic.org/healthy-lifestyle/stress-management/in-depth/exercise-and-stress/art-20044469

[3] Benefits of Physical Activity | Physical Activity | CDC, https://www.cdc.gov/physicalactivity/basics/pa-health/index.htm