Project Documentation

Company Name: Fynal Rep

By:

- **Hyobin Im**
- **Manraj Garg**
- **Robin Lansiquot**
- **Paul Tracz**

Table of Contents

Introduction

Fynal Rep is a health and fitness application that utilizes voice recognition to track the users reps, sets, and a timer. The purpose for making this application is so that users do not have to constantly interface with their application to record their progress. As a result, the user spends less time on their phone and avoids being distracted by potential text messages, emails, the internet, etc.

The application allows the user to create an account, the credentials for which are stored on our database. Once the user log's in, they can create workouts and begin adding exercises. Each exercise allows the user to adjust the reps, sets, and weights. Once the user navigates to the start workout layout, they will be prompted to select a workout. The timer for which is set to a standard 1 min 30 second rest period, after which the user will hear a noise to begin their next set. The user can use keywords complete, pause, and resume to control their workout progress.

Performance and Compatibility

- Offline usage of vosk allows for lower battery consumption
- Without access to the internet much of the functionality within the application remains, however, saved exercises for all users, and custom keywords for paid users will require an internet connection to access stored data on the database.
- Developed on android studio using java programming language. Compatible with android mobile devices with API 29 or above
- Small keyword Dictionary in order to improve classifier accuracy
- Continuous listening during workout, disabled on other screens
- Interface developed using xml pages in the resources folder of android studio
- Google login, you can use your g-mail account to sign up for the application. Your credentials will be stored on our database for authentication
- Google speech boasts a 4.9% word error rate and Microsoft speech claims to be around 5.1%. Vosk has a slightly higher word error rate at 5.43%.
- Firebase offers up to 10 GB of free storage. The user will have to contact Paul Tracz or Robin Lansiquot to be given permissions to view current usage and billing here

Major Requirements

- Security
  - Firebase authentication provides backend services, and ready-made UI libraries to authenticate users with ease
  - Strong performance with small to medium sized databases
  - provides federated identity, meaning that a user's account can be linked through various different providers such as: Facebook, Twitter, Google, GitHub
- Background Noise
  - Requires external microphone in the form of headset or headphones for improved accuracy of speech recognition.
  - High WER without microphone
- Custom Keywords
  - Paying users will have custom keywords stored on the database to be used in the Dictionary. These requests will require a developer to make adjustments based on the request.
- API
  - We are using Vosk as our speech to text API. It is enabled on all active workout screens and our application recognizes the converted text and performs functions based on the keywords we've set up. You can view full documentation here
- Github
  - The application is hosted on github repository here for which users can contact Manraj Garg for access. All changes must have commits, if the user requires to do testing without affecting the master branch, place a request for a personal branch.
- Android Studio
  - All coding is to be done on this platform in Java. Users are required to use an SDK of 29 or above as per vosk API requirements.

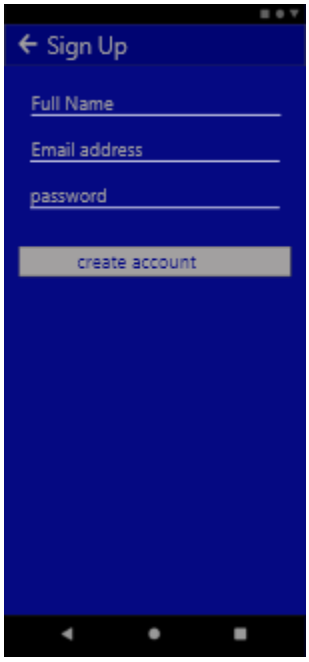
Development Standards

We are using Java as our language, so we want to follow the coding standards for Java. Which are the following:

- **Components** - write components name by its purpose for readability and maintainability of code
- **Classes** - class name should be a noun starting with uppercase letter and if it is a multi-word class, beginning of every word should start with uppercase **(e.g. Dog, StringBuffer)**
- **Interfaces** - interface name should be adjective starting with uppercase letter and if it is a multi-word interface, beginning of every word should start with uppercase **(e.g. Runnable, Serializable)**
- **Methods** - should either be a verb or verb-noun combination starting with a lowercase. If it is a multi-word method, beginning of the inner word should start with uppercase **(e.g. print(), setPrice())**
- **Variables** - should be a noun starting with a lowercase. If it is a multi-word than every inner word should start with uppercase **(e.g. name, phoneNumber)**
- **Design for reuse** - for reuse, the code needs to be high-quality, which means that it should be safe, secure and reliable

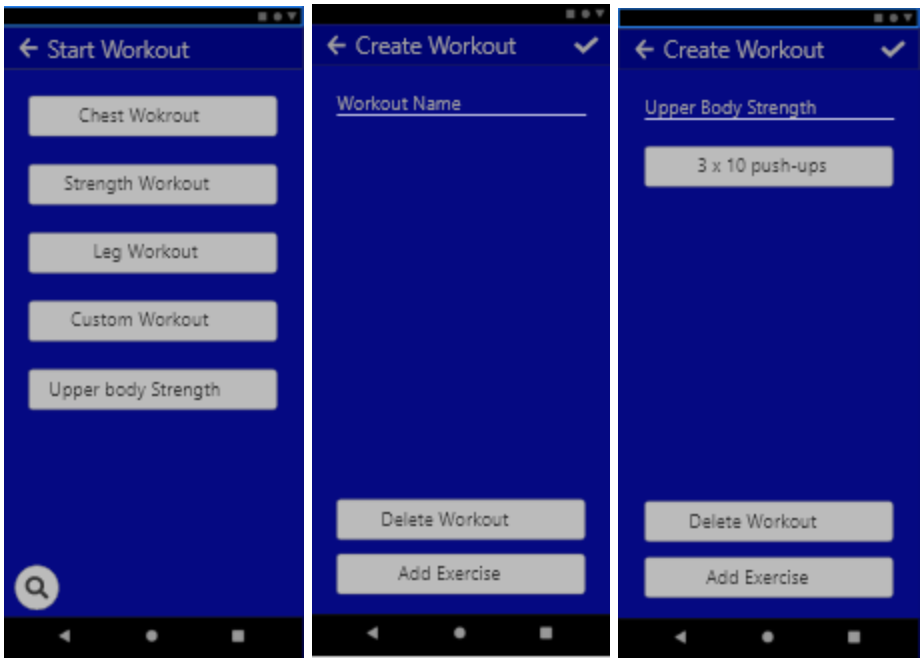
Wireframes

1. Sign up with full name, email address and password.

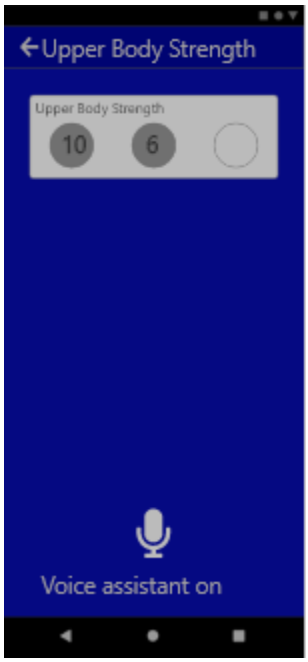


2. Log in using the information

3. Create custom keywords for the application to respond to (for premium users only)
4. Select or create workout based on user needs



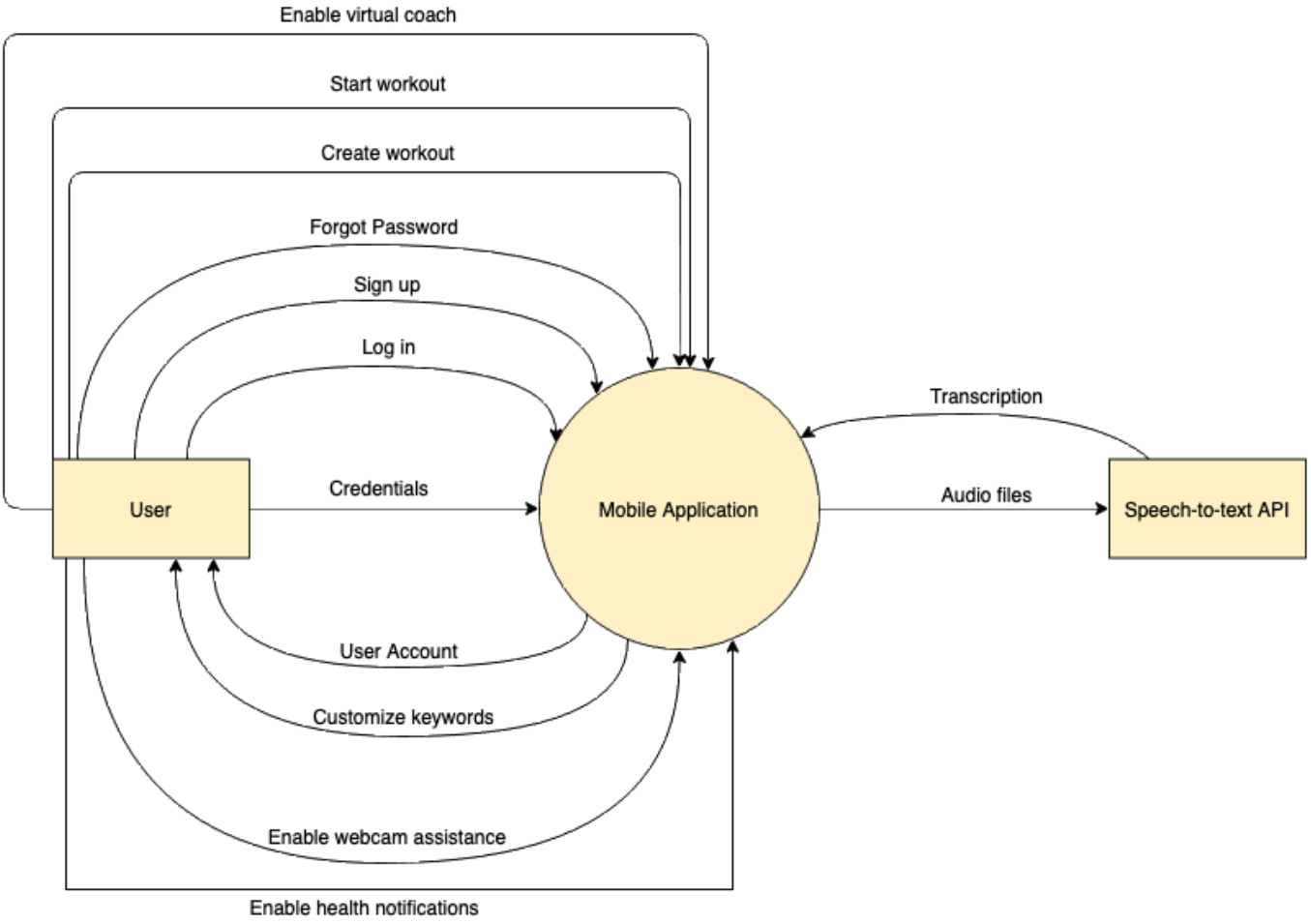
5. Start workout, which will activate the voice recognition (this feature makes our application different from the competitive applications allowing users to control the application with voice)



6. Use default keywords to pause/continue your workout (“Start workout” and “Pause workout”)
7. After each set of workouts, the application will begin a countdown and after the break, will make a noise alerting the users to continue with the next set. (This will prevent users from getting distracted or side tracked during their workout and continue with the workout)

8. The break time can be determined by the user for shorter or longer breaks in between sets.
9. Custom keywords can be used to count reps in each set for the premium users.

Context Diagram



Entity Relationship Diagram

- Most of the critical features are available within the application
- Application will have a built in Dictionary for keywords, however, paid users can create custom keywords to be stored on database