Software Requirements Specification

Vladlena Ermak, Roman Chulkov, Alexander Karavaev
Computer Science Center, CSC

Spring 2021

1 Introduction

1.1 Purpose

The purpose of this document is to propose buddy app, that will help users to detect their body errors while doing exercises.

1.2 Document convention

This document used following conventions

API	application programming interface
RPS	requests per second

1.3 Intended audience and reading suggestions

This project is for people, who want to exerise and their homes or in other places where proffessional gym trainer is absent.

1.4 Project scope

Project will include developing of mobile app, corresponding web services and graphical design for convenient user experience.

1.5 References

IEEE. IEEE Std 830–1998 IEEE Recommended Practice for Software Requirements Specifications. IEEE Computer Society, 1998.

2 Overall Description

2.1 Product perspective

Product will be a mobile app, that will analyze live and recorded video feed from smartphone's camera and detect errors in exercising. On the video errors in limbs trajectories will be highlighted and later compared with proper exercise movement.

2.2 Product features

- Error Notification
- Instructional Video
- Individual Training Program
- Achievement System

- External Services Integration
- Customer Support

2.3 User class and characteristics

User classes are the following

- People who regularly and consistently exercise. Their will use almost all features of the app, including deep statistics
- Regular people, who exercise from time to time and don't strictly do the gym programm. This class will mostly use only video instructions and look for errors in analyzed video.

2.4 Operating environment

Software will operate on iOS 13+ and Android 7+.

2.5 Design and implementation constraints

Issues that will affect the product are

- Regulatory policy about data, since video of the user will be analyzed.
- Hardware limitations of the phone will results in need of very optimized software.
- GPU should be used, when it's possible to speed up the video analyze.
- All traffic about users statistics should be secure and ciphered.

2.6 User documentation

Documentation will include: text manual, video tutorials in app, customer support.

2.7 Assumptions and dependencies

Issues can aries in legal policies.

Dependencies will consist of open-source third-party software libraries, which licenses are OK to use in commercial use.

TBD

3 System features

3.1 Error Highlighting

3.1.1 Description and Priority

Error highlighting is one of the most crucial features of VTrainer. The goal of the product is to help our customer detect mistakes in their trainings and to achieve result they desire.

3.1.2 Stimulus/Responce Sequences

In first, one should record video of their training. This can be achieved through our app by opening camera interface from main screen of the app. After finishing training one should stop recording and confirm video analysis. After completing analysis VTrainer will notify person and user can watch analysed video with highlighted errors.

3.1.3 Functional Requirements

In addition to the ability to record and store video needed for this feature VTrainer system must have deep learning model to analyse given video in order to extract one's body position and transform it to a set of vertexes. After such process these vertexes should be analysed and compared with the perfect excercise example, prebuilt in the system, and all of the errors made by a user should be rendered to the video and displayed back with some explanations of what have been done wrong.

3.2 Error Notification

3.2.1 Description and Priority

Error notification is zero-delay notification feature about errors made during excercise. VTrainer will notify excercising person about all the errors made through the one of predetermined notification channels (audio signals, flashlight, fitness bracelet vibrations, etc).

3.2.2 Stimulus/Responce Sequences

One can enable this feature at a video recording stage. After starting the excercise user will receive notifications about made mistakes which can be ignored without any consequences.

3.2.3 Functional Requirements

This feature uses the same deep learning model and vertex analyzation algorithms as error highlightning feature, but processes video in a real time and transmits error's information to notification API instead of rendering it on top

of existing video. Due to high computational capabilities of the one's device, this feature restraints the set of appropriate devices.

3.3 Individual Training Program

3.3.1 Description and Priority

The goal of this feature is to suggest an optimal set of excercises and their tension depending on the current parameters of one's body and the history of previous trainings.

3.3.2 Stimulus/Responce Sequences

On the first launch of VTrainer user will be prompted to calibrate app's understanding about user's capababilities in certain excercises (pushups, pullups, squats, etc). After that VTrainer will be able to propose different training programs to user. After every excercise app will adjust some parameters depending on user's progress. Additionally there will be opportunity to recalibrate current data in settings menu.

3.3.3 Functional Requirements

In order to implement this feature we should develop a system that will generate new training programs depending on user's physical metrics and recent trainings. We should also research modern fitness and sport techniques and consult with professionals to get all the information needed for this system

3.4 Instructional Videos

3.4.1 Description and Priority

Set of instructional videos shall provide some of the basic and advanced understandings about certain exercises.

3.4.2 Stimulus/Responce Sequences

Instructional videos will be divided to a sections corresponding to a class of exercise (legs, arms, cardio, etc). In every section there will be a list of exercises with a few tutorial videos about them. After watching all videos about certain exercise, user will be prompted to consolidate their knowledge by doing this exercise.

3.4.3 Functional Requirements

This feature requires recommendations from qualified fitness trainers and sportsmen. After finding those instructors we should be able to decide which of all possible exercices we should explain to a user. After that we will record a set of lessons for every exercise we took and add them to the app interface.

3.5 Achievement System

3.5.1 Description and Priority

Statistics and achievement system will provide some aspects of gamification and will help user to stay motivated for a long time and be proud of their progress.

3.5.2 Stimulus/Responce Sequences

User can get all available information about statistics and achievements through the main screen of the app. All data needed will be calculated after one's excercise and user will be notified about their achievements if they got any.

3.5.3 Functional Requirements

To add this feature we need to decide which metrics we should track and which achievements should be added. Then we need to draw icons for all of the achievements and add all the functionality to track those metrics. All of the user's statistics and achievements will be stored in a cloud data center and will be synchronized with it to allow our users to save their progress and retreive it in case of losing their device.

3.6 External Services Integration

3.6.1 Description and Priority

External services intergration catches up to a modern approach of simplified registration and login processes and allows our users to share their achievements on desired platforms.

3.6.2 Stimulus/Responce Sequences

Upon first launch of VTrainer user will be prompted to login to system either existing account or create a new account through the existing one. They can also do it later through the special account menu on a main screen of the app. Also user can share their achievements and compare their progress to either friends or worldwide audience.

3.6.3 Functional Requirements

Feature requires development of a login/registration functionality depending on platform's API and their policies. It also needs the ability to retrieve user's friends list, support worldwide rankings and ability to post news through plarform's API.

3.7 Customer Support

3.7.1 Description and Priority

This feature allows users to contact the support service to get help with their trainings, application errors and other issues.

3.7.2 Stimulus/Responce Sequences

This feature can be accessed through the main screen of the application. User should pick request category first and then describe their problem in a dialog box.

3.7.3 Functional Requirements

This feature needs additional API in our servers along with qualified support team which should include tech specialists and sport/fitness professionals.

4 External Interface Requirements

There is no specific requirements in this section

5 Other Nonfunctional Requirements

5.1 Perfomance Requirements

5.1.1 Target Operating System

Due to specific system's API needed and device's computational requirements user's device should be operated by iOS 13+ or Android 7+.

5.1.2 Target RPS

Project's backend system should be able to sustain at least 200 RPS in first three months after product launch.

5.1.3 Low Power Consumption

Application should not consume more energy than necessary for its proper functioning. High energy consumption is always one of the main factors to uninstall an application.

5.1.4 Proper Functioning with no Internet Access

Application should be able to provide most of its features without Internet connection.

5.2 Safety Requirements

5.2.1 User's Traumas

Company should not be responsible for any kind of traumas and other physical and moral damage caused during use of VTrainer. Despite this our goal is to minimize all of the possible risks of injuries. All training programs should be designed with a focus on its safety and the user should be informed about possible risks of unproper workouts.

5.3 Security Requirements

5.3.1 Indirect Data Access

Any data stored should be protected from anyone's direct access including developers.

5.3.2 Data Safety

User data should be protected from unauthorized changes and thefts. All data should be encrypted and should be accessed only through system API.

5.4 Software Quality Requirements

There is no specific requirements in this section.

6 Other Requirements

6.1 Application Permissions

Our app should not request more device's permissions than necessary. Designing features that will extend current list of app's permissions is undesired.