SOFTWARE REQUIREMENTS SPECIFICATION

for

Lungcheck Django Server

Capstone Project

by Fu Lei

Contents

1 Introduction

1.1 Purpose

Lungcheck is a project carried out within the Capstone Project course held at Turku University, Finland, from January to October 2017. The Lungcheck project aims at providing a platform to connect patients with some lung disease (or even healthy people that regularly checks the lungs in order to prevent or detect any disease as soon as possible) together with doctors that can diagnose those patients without ever meeting them. Not only that, in order to have a preliminary result on the health of one's own lungs, the Lungcheck team has worked on an Artificial Intelligence powered algorithm that can detect many different kinds of diseases automatically. In order to do so, the Lungcheck team has also worked on a hardware prototype that can be used to record the lungs' sounds by following a set of instructions given through the Lungcheck Android app. The project consists therefore in three main differentiated parts:

- A custom developed digital stethoscope that can be connected to the audio jack port of a smartphone.
- An Android app which provides communication with the assigned doctor as well as an interface with instructions on how to record properly the lungs' sounds.
- A Django-based Python 3 app which runs in a cloud server and which serves both as a bridge platform between doctors and patients and a tool for automatically assess the probability of having a lung disease based on the processing of the recorded sounds.

This document focuses on the specification of the Django app. The purpose of the Django-based app running in a cloud server is to provide:

- A public webpage to promote the Lungcheck project, by introducing the idea and benefits to both potential doctors and patients, as well as the general public.
- An intranet available to both registered patients and doctors, which serves not only as a communication tool but also as a reporting interface of the results of the artificial intelligence powered algorithm that automatically analyzes one's lungs' sounds.
- An automated tool that analyses the main features that can be extracted from the recorded sounds so that patients can have a preliminary idea of their lung's health without a doctor that needs to check them.

- A basic REST API in order to communicate with the currently developed Android app as well as any other future applications built for any other platform.

1.2 Intended Audience and Reading Suggestions

This document is intended for those willing to run their own instance of the Lungcheck Django app.

1.3 Project Scope

The scope of this project is to provide a basic functional Python web application that can get the recorded data from an Android smartphone, analyze it and report that analysis both to the patient and his/her assigned doctor, if any.

2 Overall Description

2.1 Product Perspective

This SRS is part of a larger system description, which includes the specification of the sounds analysis algorithm as well as the digital stethoscope specification.

2.2 Product Functions

As introduced before, the main product functions are

- A public webpage to promote the Lungcheck project, by introducing the idea and benefits to both potential doctors and patients, as well as the general public.
- An intranet available to both registered patients and doctors, which serves not only as a communication tool but also as a reporting interface of the results of the artificial intelligence powered algorithm that automatically analyzes one's lungs' sounds.
- An automated tool that analyses the main features that can be extracted from the recorded sounds so that patients can have a preliminary idea of their lung's health without a doctor that needs to check them.
- A basic REST API in order to communicate with the currently developed Android app as well as any other future applications built for any other platform.

2.3 User Classes and Characteristics

The interface provided by this project in the form of a web app is intended for:

- Public website: intended for the general public, specifically potential doctors that might want to contribute to the project and patients (clients) that want to use our product.
- Intranet: intended for registered users; both patients and doctors.
- Analysis tool: intended for both registered patients and doctors (if any is assigned to a patient), who will get the result in the form or a report with the status of the lungs: healthy, sick, or need doctor reevaluation (if the algorithm cannot assess the healthiness of the lungs with enough probability).

2.4 Operating Environment

Since this is a Django-based app it can run in any system that can run Python 3. It has been tested under a machine running Ubuntu Server 16.04.2 LTS with 512 MB of RAM. The current configuration uses MySQL as the database layer but it can be easily configured to use sqlite and therefore making the instance lighter (less memory usage) and platform agnostic.

2.5 Assumptions and Dependencies

The only requirement is to have Python 3 installed in the system as well as Django, which can be installed directly, through Python-pip or other methods (refer to the Django website for reference).

3 External Interface Requirements

3.1 User Interfaces

The only user interface needed in order to be able to use this web app is any device with a browser and connected to the global Internet network. However, in the case of a patient, in order to fully use all the features provided by the Lungcheck project, then he/she will need to have access to our digital stethoscope as well as an Android smartphone.

3.2 Hardware Interfaces

The hardware interface for the digital stethoscope is described in its own SRS.

3.3 Software Interfaces

A browser connected to the Internet.

4 System Features

This chapter describes more in-depth the three features of the system introduced in the first chapter.

4.1 Public website

The lungcheck public website, currently hosted at lungcheck.tk, provides information about our project in general, as well as in particular to both doctors and websites. The content of the website is currently in development stage but will include the following:

4.1.1 Main page

Main page (home), including the benefits of using our system for society: easy, in-home periodical revision of lungs' sounds in order to prematurely diagnose potential lung diseases.

4.1.2 Patient Information Page

Patient information page, including links to the patient intranet, but also introducing the product (web app, Android app and digital stethoscope) as well as explaining how easy it is to confirm one's lung's health without visiting a doctor, but at the same time getting a notification asking to visit the doctor if the recording indicated that there might be some infection or disease.

4.1.3 Doctor Information Page

Doctor information page, including how easily they can help patients not only in their surroundings but all over the world with a communication interface and access to all their recordings, as well as the automated reports, so that they don't need to listed and reevaluate every single recording but only those that the Lungcheck analysis tool flags as uncertain or sick.

4.2 Intranet and Automatic Evaluation Tool

The web app intranet provides a user-friendly interface for both users and doctors, as well as a base for the automatic evaluation tool.

4.2.1 Patient Intranet

By logging in in the web app, patients have access to all their recordings' history, as well as the automatic report generated for each one of them individually. They can also use this intranet to communicate with their assigned doctor, if any, and to check the notes that their doctor might have left for any recording that has been reevaluated.

4.2.2 Doctor Intranet

By logging in in the web app, doctors can check all their assigned patients' recordings, as well as reevaluate any of them, but mainly those that are notified to them as uncertainly classified by the automatic evaluation tool (and those classified as sicks). Doctors can also use this intranet to communicate and send messages to any of their patients.

4.2.3 Automatic Evaluation Tool

The automatic evaluation tool is a Python script that uses signal processing techniques as well as machine learning algorithms to extract the main features from the lungs' sounds and classify the set of those features as healthy or within a group of known diseases for which the algorithm has been previously trained.

Appendix A: Code

All the code necessary to run this Django-based web app can be found under the GitHub repository sisqui/lungcheck.