Project Proposal

Doctor Robert

Project Name	Doctor Robert
Project Leader	D. Weidinger
Document state	In process
Version	v. 1.0

Revisions

Date	Author	Version/Change
September 19, 2019	Weidinger, Gruber, Tanzer	First version
October 04, 2019	Weidinger, Gruber, Tanzer	Second version
October 07, 2019	Weidinger, Gruber, Tanzer	Third version
October 08, 2019	Weidinger, Gruber, Tanzer	Fourth version
October 10, 2019	Weidinger, Gruber, Tanzer	Final version 1.0
October 21, 2019	Weidinger, Gruber, Tanzer	Final version 1.1
October 24, 2019	Weidinger, Gruber, Tanzer	Final version 1.2

Contents

1 Introduction

The end result of our project acts like a helping hand in the everyday life. Instead of googling ones symptoms or complaints, the product uses a Machine-Learning Model, trained on medical data, to provide answers in the form of diagnoses and possible treatment plans. The product takes the user-input and computes possible solutions for ones disease.

2 Initial Situation

When people start to show symptoms of a possible sickness or get mildly injured, they often tend to misjudge or downplay the severity of their conditions, to justify not making a doctors appointment, since going to the doctors is a big inconvenience for most people in this day of age. This behaviour often leads those people to seek out to google for a diagnosis.

Googling symptoms can be quite tedious and unnecessarily time-consuming. It involves opening a lot of sites, skimming trough loads of text and most of the time the results are rather inaccurate.

Although there are already good individual solutions for medical third opinions, there has yet to be a simple unified platform for all individual cases.

For example Netdoktor, the market leader for medical articles and questions in German speaking countries, has a program called Symptom-Checker that tries to simplify the process of searching for medical treatments on their own website by asking the user questions about their gender, age, area of pain, etc. This process also includes redundant questions and limits the user to input only by selection, which makes it hard to accurately describe symptoms. Netdoktor only supports the German language so if a user wants an English diagnosis they have to switch to a different service.

Such services also lack features like scanning skin in order to diagnose skin diseases, so a user has to use, or even download, another application like for example Skin Vision, which is a program used to recognize skin diseases based on images. This could lead the user to not look up their suspicious looking piece of skin.

As of now we are not aware of any healthcare assistants or symptom checkers that make use of sophisticated Artificial Intelligence algorithms. Neuronal networks trained on medical data, like BioBert, already exist and outperform established solutions in regards of accuracy and convenience but are not easily accessible for the ordinary consumer.

3   General Conditions and Constraints

The proposed system has to the deal with the following conditions and constraints:

3.1   Confidentiality and Legality

We have to communicate that we are no substitute for a doctor

To avoid legal consequences, the user needs to be informed that we take no responsibility for the diagnoses or information provided. Our competitors, namely NetDoktor, solved this problem by clarifying this before and while the interaction. Additionally, we have to shield ourselves from possible lawsuits by making the user to agree to our Terms of Service.

All interactions with the user must be confidential

Trust is one of the most important factor in the medical field. Nobody is willing to share their personal well-being when their is doubt in the confidentiality. This is why end-to-end encryption for user interactions is obligatory.

Even if we do not save medical data from our users, men-in-the-middle attacks are a possible threat. Attackers could potentially intercept data sent by the user. For example, ones employer could get their hands on a female workers pregnancy diagnose and fire her as a result.

Personalize the experience with general information about the user

The user, if wanted, signs-up for a more personalized experience. Depending on how the user is logged in, either anonymous or with an third party OAuth2 login like Google's, the user experience will slightly change in the way that he will for example be addressed by his name, the last time he logged in or other meta-information.

No medical data will be stored(For the moment)

We currently do not have the capacity to deal with sensible data like medical information. Saving the personal medical data would be an additional legal threat, since we also cannot use it to improve our product we are not planning to take a risk in doing so. This could change in the future.

3.2 UX and technical constrains

We have to be more convenient/precise than google when it comes to medical questions

Otherwise, there would be no reason to use our product.

We have to avoid notoriety by giving false alarms so our critical evaluations like skin-cancer must be accurate

It is easy to get a bad reputation in the medical field. One bad diagnosis can lead to an public outcry. This factor is especially important since our application is built on trust between the user and Doctor Robert, therefor our diagnoses have to be accurate.

The output of our model must not be to technical and/or needs further explanation-methods like descriptions of medical terms

Outputs like: "generally ekg will identify the long qt and brugada syndrome." need to be more readable. A lot of times the output text includes medical terms. For these words we have to provide further explanation.

4 Project Objectives and System Concepts

Our main goal is to represent a helpful third opinion. Potential users can ask Doctor Robert all questions regarding physical health and well being. The unbiased recommendations of Robert can then be used as is or for further research.

4.1 Login

The user, if wanted, signs-up for a more personalized experience. Depending on how the user is logged in, either anonymous or with an third party OAuth2 login like Google's, the user experience will slightly change in the way that he will be addressed by his name.

4.2 Diagnosis
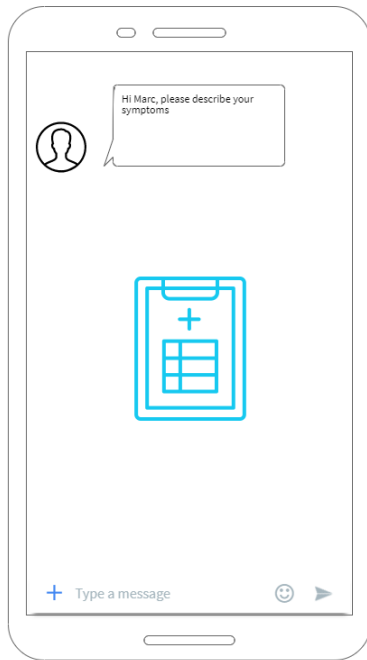
The user enters his ailments into the app. Using words or whole sentences to describe the current unwell-feeling. Since the predication will be more accurate if more keywords are read, a more precise description of the illness is better than just words. Even though words alone are sufficient.

The output informs the user about possible symptoms causing the ill-being and/or about feasible solution and if desired further elaboration on certain results.

The user then can decide how to act from that point. Depending on the outcome he can leave it be, do further research through provided links or arrange a doctors appointment.

Figure 1: Prototype

4.3 Computer-Vision based Healthcare Tasks

A picture taken by a user is used as base of operations for an elaborated computer vision algorithm. The algorithm can make viable diagnoses of the picture, thus about the state of the user. For example, skin irritations are scanned and then it is evaluated, whether the user has skin cancer or not.

Since such applications already exist around the world, the question Why self implement? may arise. The answer is fairly straight-forward. There are no services that actually provide integrable functionality. So self-implementation is practically forced.

On the contrary to the "Diagnosis tool", Computer-Vision based tasks will not be dealt with on the back-end, but locally on the mobile device. Since this will result in much lower server workload.

4.4 Readability and Language

Translation

The user interface will be available in multiple languages, mainly English. As the system is based in English, a translation is made to German or other languages via the Google translation API.

Keyword Recognition

The given diagnoses often contain medical terms that are not understandable for the normal person. So, a redirection or further explanation has to be given for those appearances. This will be resolved in a pop-up window that gives a short explanation and provides a link for further research.

4.5 Mobile-Application

The mobile version features Login, "Diagnosis" telling and Computer-Vision Healthcare Tasks. The mobile-app and the web-app, both form the front-end. The mobile-app includes all functionalities stated above, whereas the web-app has to give up on the Computer-Vision Tasks.

4.6 Web-Application

The web-app will be integrated in the website of the product. The overall experience for using the app remain relatively unchanged. The only downside it has against the mobile-app is the fact, that computer-vision scans like skin-cancer recognition will not be available due to the intricate integration of said functionality. Otherwise the usability stays the same. "Diagnoses" can be made just like in the mobile version.

4.7 Back-End

The back-end is mostly used as a REST API and for storing the users data, for the back-end consists of Google's Firebase and the REST API Server. Also the AI based "Diagnosis" giving feature is seated in the back-end for a more efficient response time.

5 Opportunities and Risks

5.1 Google Alternative

Our project presents the opportunity to form an user-base by enticing away people already using platforms like google to look up their symptoms, by providing a simpler way to get more accurate results. The process of googling can be simplified with the power of artificial intelligence. So a person does not have to dig trough various websites to get an fairly inaccurate medical diagnosis. The user will have an simple user-interface to enter their aches and pains and an accurate third medical opinion will be provided.

5.2 Unified Platform

There are already working programs and methods for giving medical diagnoses. Which gives us the opportunity to embed those methods in our platform, since the general aim of our platform is to give the user the convenience of using only one platform to give them an accurate medical opinion.

5.3 Notoriety

It is critical for a system like ours to gain a good reputation and on the other hand to avoid bad notoriety for our predictions. About 70 percent of Americans say that online reputation had influenced their choice of physician(Source). If Doctor Robert manages to get a good reputation, we could be a possible replacement for a google search. This implies a possible user-base of people equipped with internet-ready devices. The same holds true for the opposite. For example if our model gives a lot of wrong skin-cancer alarms, we could lose our reputation, respectively our relevancy.

5.4 Intelligent advertisement

One possible source of income could be advertisement of certain doctors for certain diagnoses in ones area. In order to back our claims of advertisement we would have to collect data on the usage of our application. Like we declared in our Conditions

and Constrain section, we do not store medical data. This is why the collection of the data has to be of statistical nature and purely anonymous.

5.5 Legal risk

First and foremost, it has to be said that the whole topic of AI supported medical treatments/diagnoses are intensely discussed in terms of the law. Up to now, no one, including experts, has come to the point of perspicuity - it is a grey-zone. Since we are no pioneers on the topic of digital medical diagnoses we know that it is legally possible.

6 Planning

6.1 Schedule

Project begin: Mo, 14.10.2019
Project end: End of 5th grade

Our first hurdle will be the setup for our back-end. Parallel we will start to develop a front-end. The development will start with the 4th of November.

The first working prototype is set to be available by the end of the first semester.

6.2 Management

For software development we will be using Scrum as the method for managing and tracking development process.

6.2.1 Team

Daniel Weidinger:

- Role: Teamleader
- Focus: Back-End and AI development

As a team leader the duty is neat and efficient course of action over the whole project. Thus required to ensure a fluent and communicative team work.
In charge of the development of the back-end and AI services.

Jan Gruber:

- Role: Developer
- Focus: Front-End

As the leading role in the Front-End development the main focus is in creating a clean app and website. Also partly operating in the AI development.

Rafael Tanzer:

- Role: Developer

Applying and working in all fields(back-end, front-end and AI). Main emphasis on back-end and AI and posing as a sort of intermediary between Front-End and Back-End development.

6.2.2 Versioning

The price for every used product will be provided in the parenthesis after the name

Versioning: Git(**free**) - Git is a distributed version-control system for tracking changes in source code during software development.

Repository: GitHub(**free**) - GitHub is an American company that provides hosting for software development version control using Git

Git GUI Client: GitKraken(**free**) - GitKraken is an intuitive and seamless UI for git

6.2.3 Other organisational software

Scrum Board: GitKraken GLO(**free**) - GitKraken Glo is the only incandescent issue board for task and issue tracking that is fully integrated with GitKraken.

6.2.4   Resources

The price for every used product will be provided in the parenthesis after the name

IDEs and Development Platforms:

- JetBrains's Android Studio + Flutter and Dart Plugin (**free**)

- Visual Studio Code (**free**)

- Python 3 Interpreter (**free**)

Backend:

- Firebase for user-data storage(**free up to a certain point then we pay per usage.**)

- Server with an Python runtime for REST calls (**free up to a certain point if we do not use it commercially**)

- Tensorflow Serving for faster machine-learning inference. (**free**)

- Google Translation API (**free up to a certain point**)

Frontend:

- Access to the Google developer console in order to make our application available (**20 Euro**)

6.3   Milestones

This section lists a brief overview of our most crucial milestones.

Milestone	Date	Description
Mobile application	31.12.2019	Cross-platform mobile application(Android/IOS) which can communicate with our back-end
User (Login/Register) integration	14.2.2020	back-end and front-end integration of a login feature and a user database
REST API	14.2.2020	Working REST API for our DocNet(our Neuronal Network)
Skin Cancer Recognition	31.04.2020	functionality for a camera based inference to recognize skin cancer.
Explanation of medical terms	31.6.2020	Based on certain medical keywords we provide further explanation
Web front-end	31.10.2020	Web front-end with the same functionality as the mobile font-end expect the computer-vision tasks

7 List of Abbreviations

- UI ... User Interface
- AI ... Artificial Intelligence
- UX ... User Experience