**Find a Doc**

**Table of Contents**

|  |  |
| --- | --- |
| **1** | **Introduction** |
| **1.1** | **Purpose** |
| **1.2** | **Intended Audience** |
| **1.3** | **Project Scope** |
| **2** | **Overall Description** |
| **2.1** | **User Classes and Characteristics** |
| **2.2** | **Operating Environment(Web Application)** |
| **2.3** | **Design and Implementation Constraint** |

Introduction:

“Find a Doc” is a web application which lets you find a doctor appropriate for your specific condition. There’s a big gaping hole in the current market where, when people get sick they don’t know what to do, many go to the doctors they have known for a while but they might not be the perfect person to visit and people end up paying their time and many, and many times the doctors would prescribe medicine instead of referring to an appropriate Doctor who specializes in that field. So this web application helps you cut the middle man and save you time, money and any risk associated with waiting long enough to visit the person who can make a better diagnosis.

Purpose:

The purpose of this project is to reduce the amount of risk associated with not being able to visit an appropriate doctor at the right time. It also aims to reduce the expense of time and money.

Intended Audience:

The intended audience is anyone who is looking for a doctor for themselves or a loved one. But if it were to be narrowed down, then people who are good with tech and have to take responsibilities in their day to day life, so anyone from the age of 17 and above would be the ideal target audience.

Project Scope:

This system provides a very well sorted out environment which abstracts a lot of unnecessary information and helps reduce time, efficiency and discrepancies. Once the people find this application it will help them in various aspects, considering that millennials tend to do a lot of research for themselves, this platform will help them narrow down their search and find an appropriate doctor or solution.

Overall Description:

The web application will have a front page where there will be an option where a user can click in and it will provide a bunch of options to choose from to tell the system what type of disease they may have. If the user is unsure then there will also be a symptom checker which will help them narrow down to what disease they may have, then the system will spit out a bunch of doctor’s profile closest to their location, and they can click on the profile and set up an appointment.

User Classes and Characteristics:

Physical actors:

User: Users will be able to login in and look for a doctor.

Doctor: A doctor can log in and check their appointments and patient’s history if there are any.

System actors:

Client – The client actor will connect with the server and fetch data and make appropriate changes.

Server and Database – The Server will help connect with the database and the database will store data.

Operating Environment (Web Application):

It can run on any modern OS with an internet connection because it’s a web application.

Design and Implementation constraints:

The constraint with the implementation is that when the customers create an appointment, how will the system manage these appointments for the doctors because they may have prior appointments from other sources, one solution is to create a system for the doctors to manage all their appointments from all sources, or just leave the management portion to them.