SRS for HealthCare

TEAM NAME: NO.1 in Korea
Team members: FENG,CHENLU-20184921; Xu Haobo-20186558; Liu Tonghao-20186551;Wu Tong-20186557 & Mikkel Sinkjaer-20186496

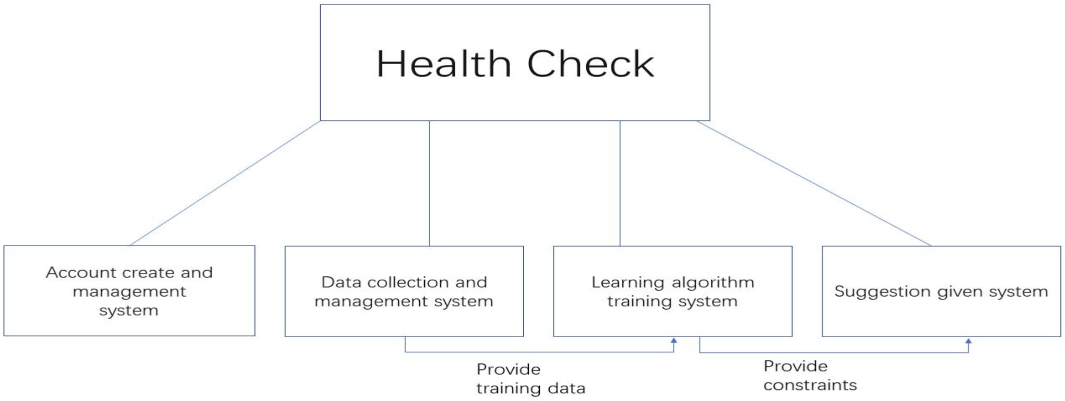
2. Introduction

Our product is an application which is used by users typing in informations about their body status, habits and other factors that may cause their diseases. Our product is user-oriented. Users can use it to check their health and get suggestions to improve or stay their health. At the same time, doctors can use it to give some suggestions to users and get health data of users. On the other hand, our product used only in the field of health is supplied to government for investigating the situation of public health and enhancing the healthy level of citizens. So the users of our product is the citizens whether they have diseases or not.
The goal of the project is to help you check the standard of health and if there is some abnormal, we will send you some useful suggestions. We use the machine learning to train the dataset and we also discuss a lot about the algorithm which can be used to compare your data with the dataset then get a relative accurate result. The most top level requirements is that we can compare your data with the dataset and then get an accurate result so that we know which is the most potential disease you will get and give your suggestion to avoid it.

3. Overall Description

A. Product Perspective

The name of the app is HealthCheck. Topic for the app is personal health and how to improve users health by trying to avoid lifestyle diseases. The should collect users own personal data to be used for understanding public health. The app is not meant for making a profit, but a tool governments can provide for their citizens and collect data for research. The app is a new, self-contained product with subsystems, testing files and documents. The following diagram shows the overall program.


Health Check
Account create and management system
Data collection and management system
Learning algorithm training system
Suggestion given system
Provide training data
Provide constraints

B. Product Features

The major features of our app are friendly user interface because our major users are old people we design clean and simple user interface with big buttons and clear text; prediction, we will use public data to find a learning algorithm to predict if user has a certain disease; data collection, if user wants to use our app, they need to input their health related data. At the same time we will collect such data for further study and research or provide them to government; safety, we will check users' identity by using their security number.

The major functions of our app are creating a new account, logging in, inputting health data, giving suggestions and setting. First, when users download our app they could create a new account or just choose to do prediction and get suggestion. If they create a new account, we will trace their data and next time they could update their data. For creating a new account, users just need to input their basic information given in the instructions and set their password. Users could change their information in the setting function. Before doing prediction, users need to choose which disease they want to predict and then input the corresponding health data and we will give users instructions to tell them which data to input. Then they could push the suggestion button, the app will jump to suggestion page with their health condition and top 2 high ranking suggestions.

We also have another kind of user, doctors. For doctors, they have functions of create a new account and check their patients health data with patients' permissions. In this part, we will set safety check function-user authentication to check the identity of our users.

C. Operating Environment

Our program will run on platform which has python because our program is a terminal program. No matter what hardware, smartphone or laptop, once they have python installed, they could use our app. The tool we will use is python with version 3.6. The packages we will use are Numpy, Scikit, Pandas, Jupyter, Xgboost and some basic packages. We have not decided whether we will build database or not. If so, we will also use SQLite. We will also access to email system to send emails to our users.

D. Assumption and Dependencies

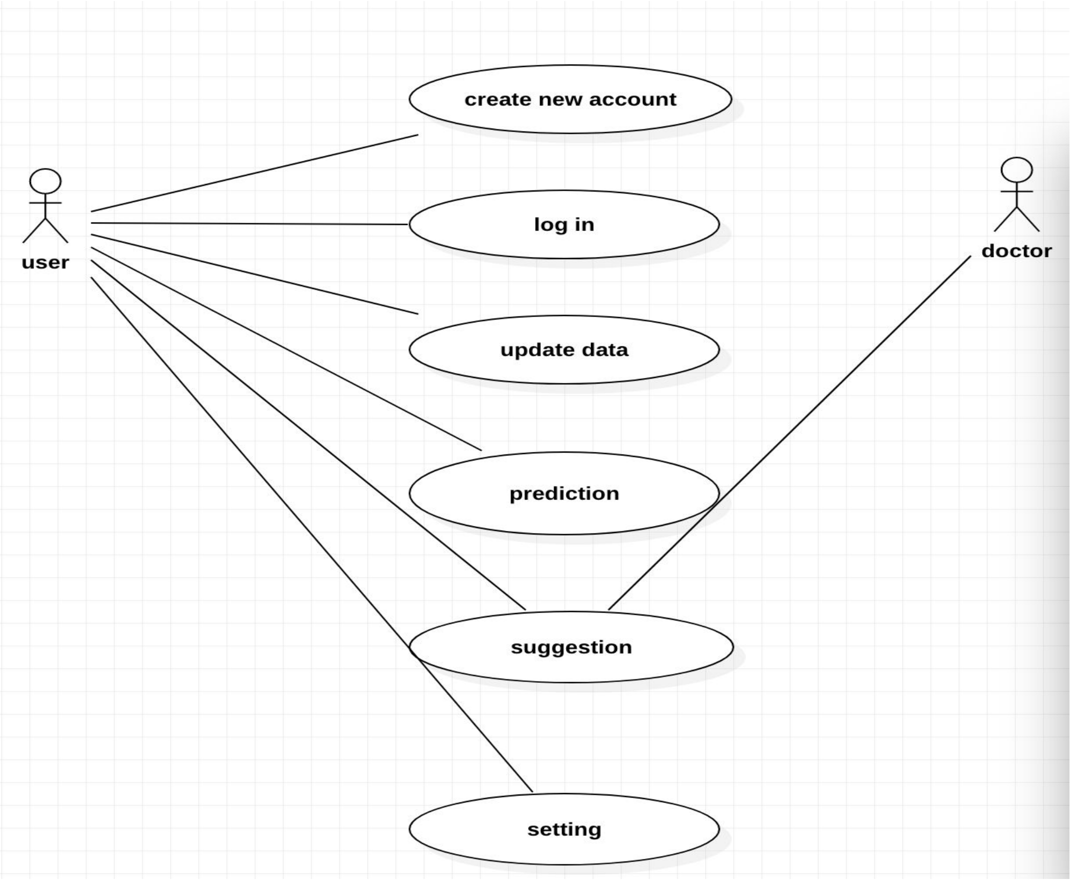
We will not use any third-party or commercial components and the issues maybel come from our system including training dataset and learning algorithm issues, like the number of training data will affect my learning algorithm.

4. System Features

A. Functional Requirements

First, the user can create their own account with their email address and log in to input their own data like height or weight.
Second, the doctor can get the health data from user, analyzing the data. Finally , give the healthy suggestion.

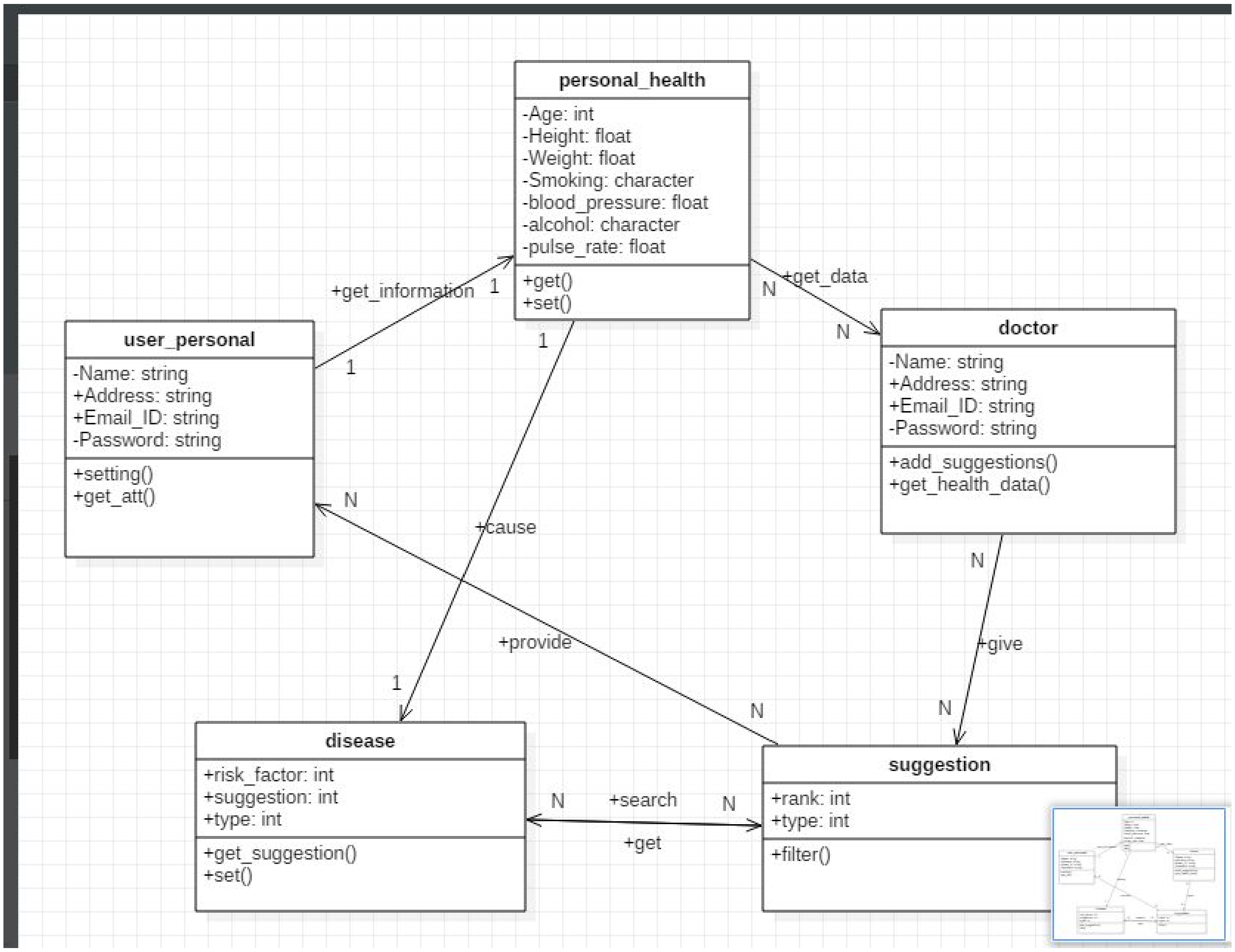
B. Use Case Diagram & Descriptions



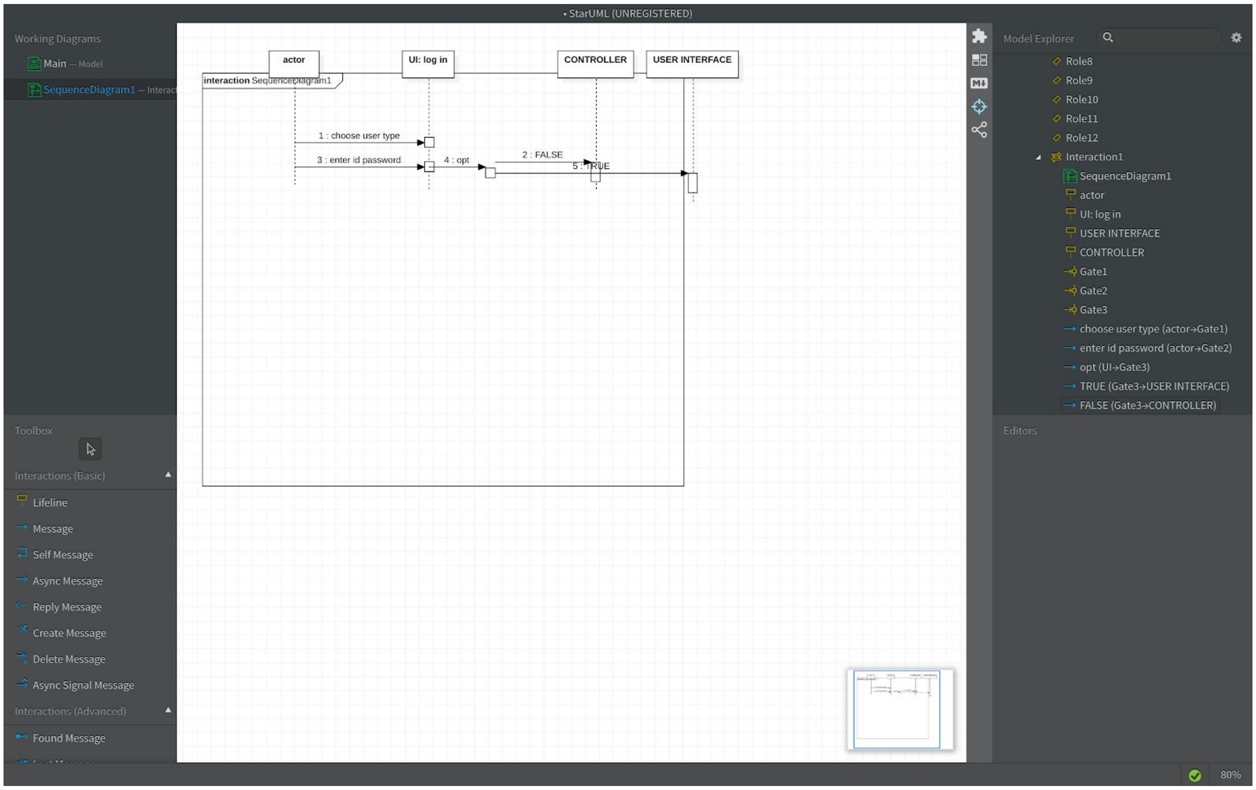
Use case name	Create new account	Log in	Update data
Related requirements (indication when this user case completely fulfills)	The user got an unique account with their own data.	The users can log in with their own account.	The user can modify their own data when they need to.
Goal in context	A new user require a new account with their email from the administrator	A new or old user who already get an account should be able to login with their email and password	A user should have the function to modify or update their data if their health data have some change
Preconditions	The system is limited to recognized users and so the user needs to have appropriate proof of identity.	The system is limited to make a prediction about the disease for different user so the users should login with their email	The system is limited to maintain accurate if the data from user changed so the user needs to update data.
Successful end condition	A new account is created for a new user	A user class is built with his own health data	A new user class is built after modification finished
Failed end condition	The application for a new account is rejected	The user can not login with the email address	The modification of the users' data is rejected
Primary actors	user	user	user
Secondary actors	user account data	User information data	User information data
Trigger	The system asks the user to create a new email account	The system asks the user to login with email address	The system asks the user information database to modify the data
Main flow	1.The system asks the user to create a new email account. 2.The user enters the email address with the password and data about their body 3.The new email account is	1.the system asks the user to login 2.the user input their email address and password 3.the user has login successfully	1.the user asks the system to modify his own data 2.the user input the new data 3.new data saved after modifying

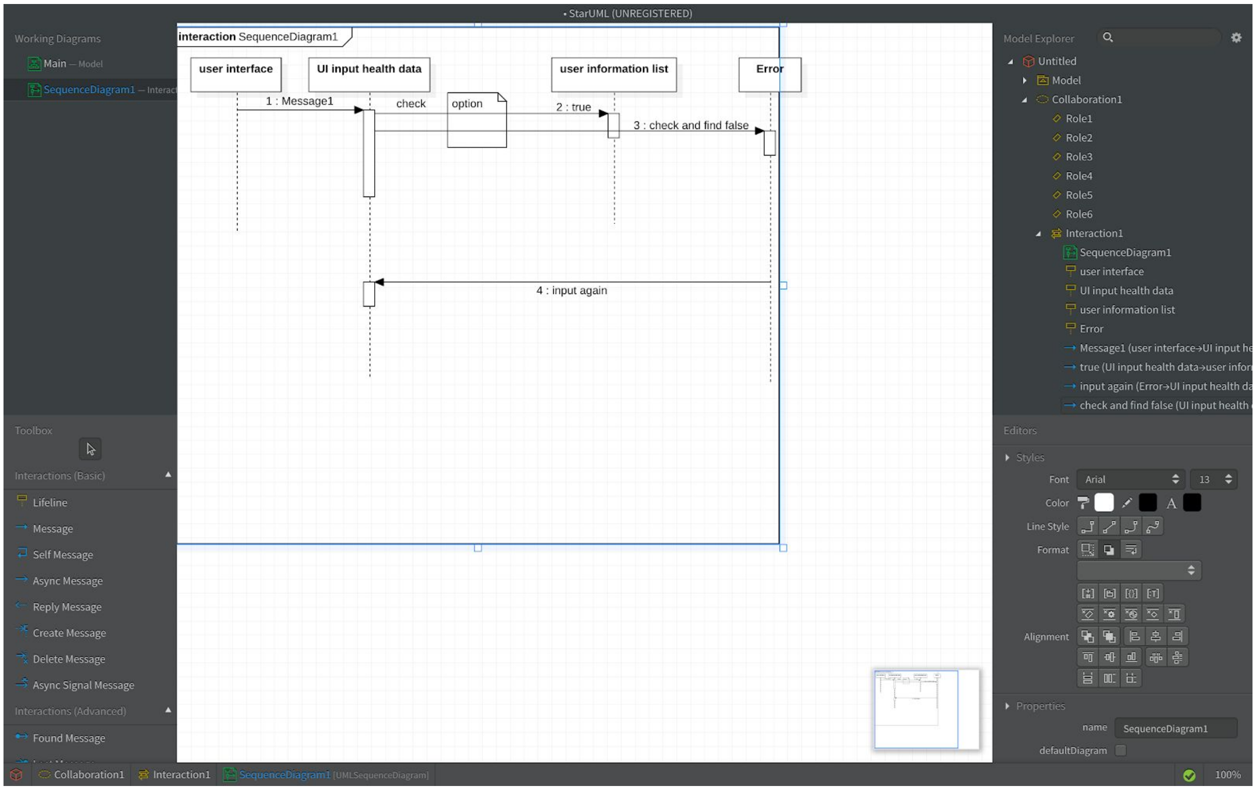
	created.		4.the data has been modified successfully.
Extensions (Branching action)	The application for a new account is rejected	The user can not login with the email address	The modification of the users’ data is rejected

C. Domain Model



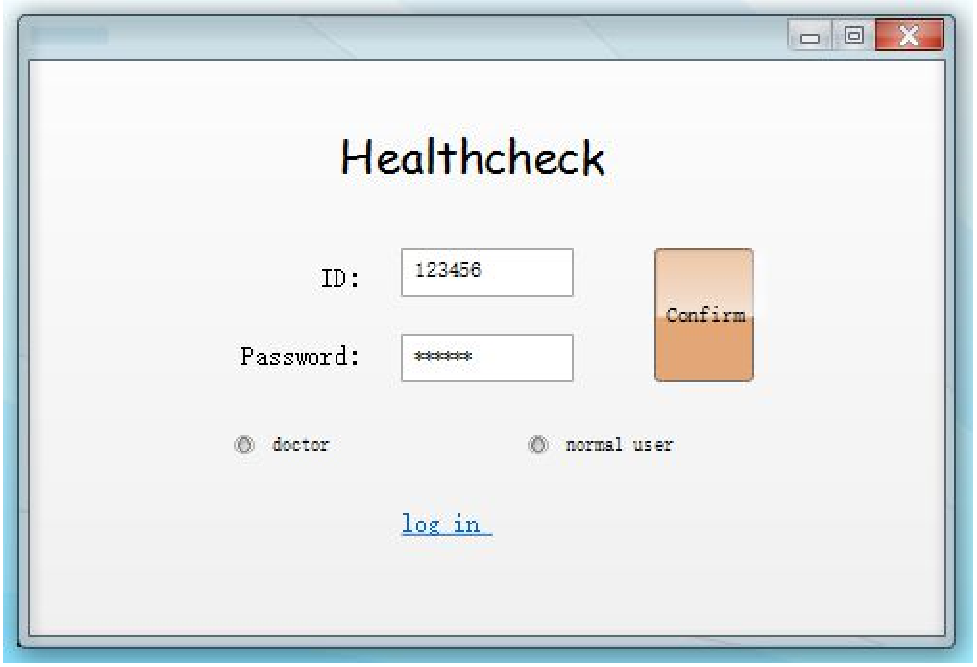
D. Sequence Diagrams





5.Preliminary User Manual

(1)This is the home page, you can choose your character to access, or click log in.

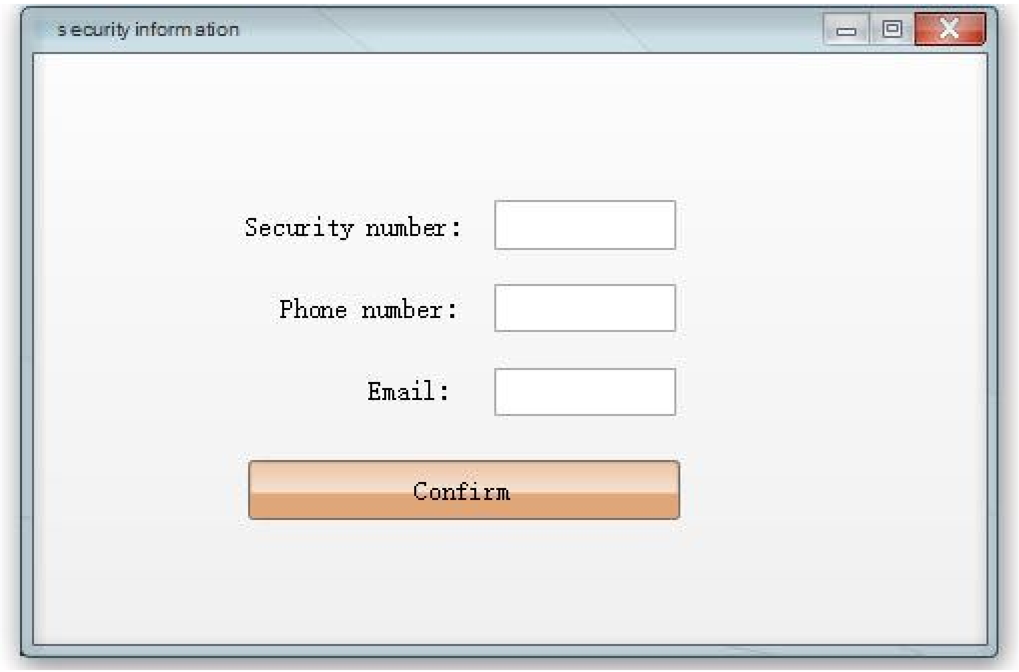


Healthcheck

ID: 123456

Password: ******

(2)The security information for normal user to log in.



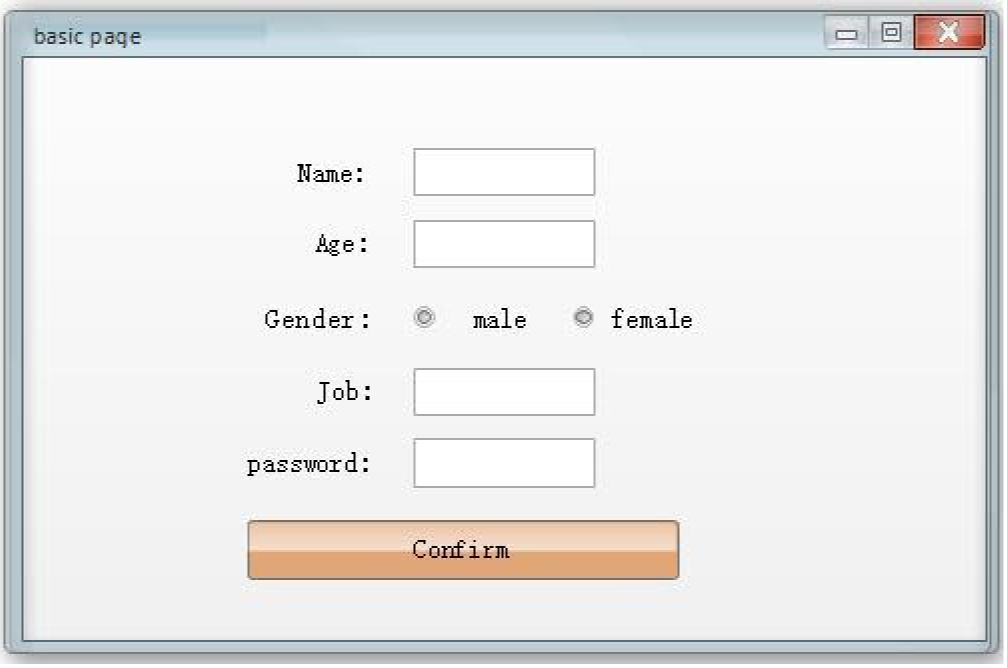
s ecurity inform ation

Security number:

Phone number:

Email:

(3)The basic information for normal user to log in.



basic page

Name:

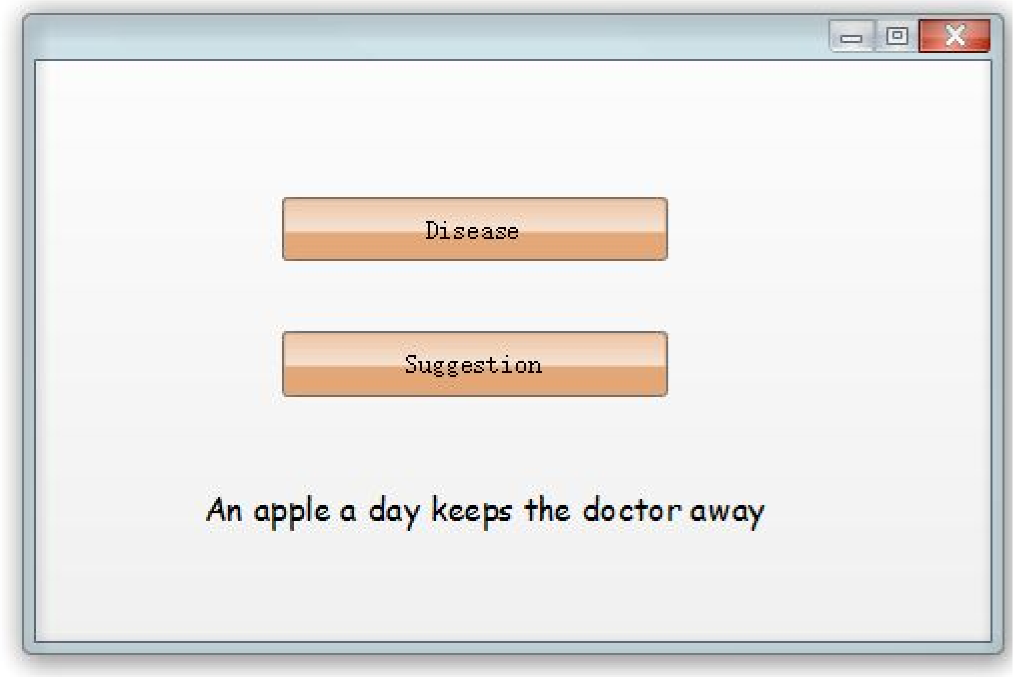
Age:

Gender: ○ male ○ female

Job:

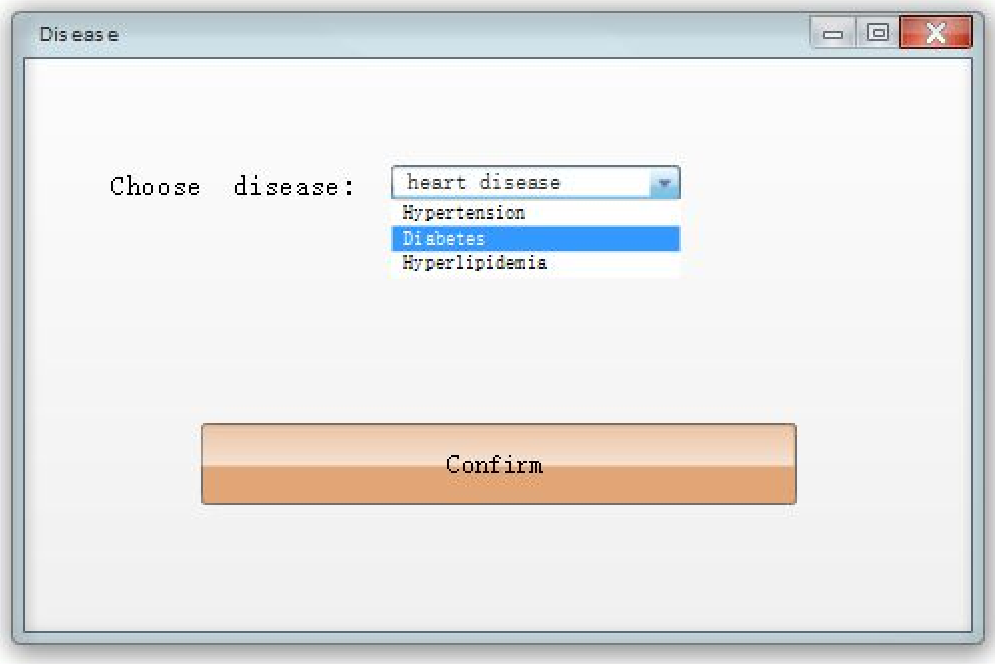
password:

(4) Complement information to predict or get suggestions



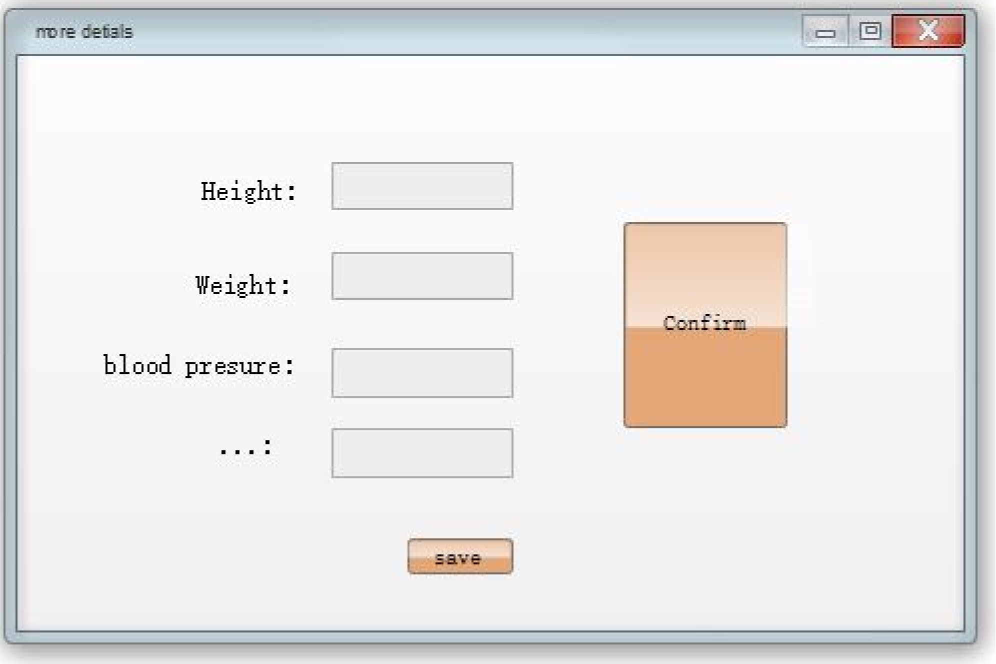
An apple a day keeps the doctor away

(5)Choose diseases

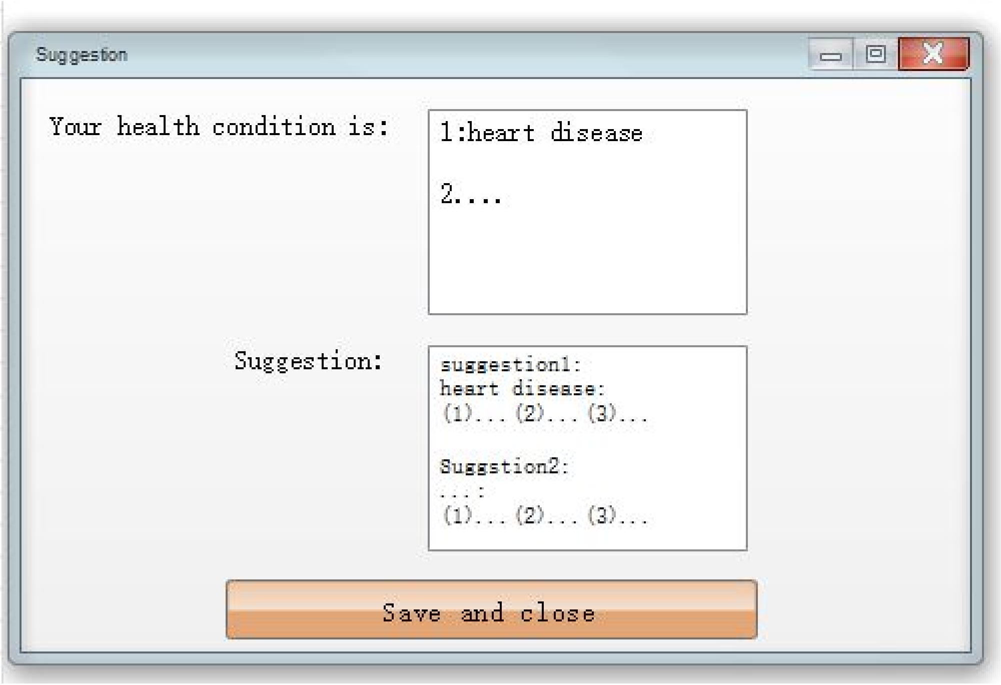


Choose disease: heart disease

(6)more details of health information



(7)The user gets the suggestion



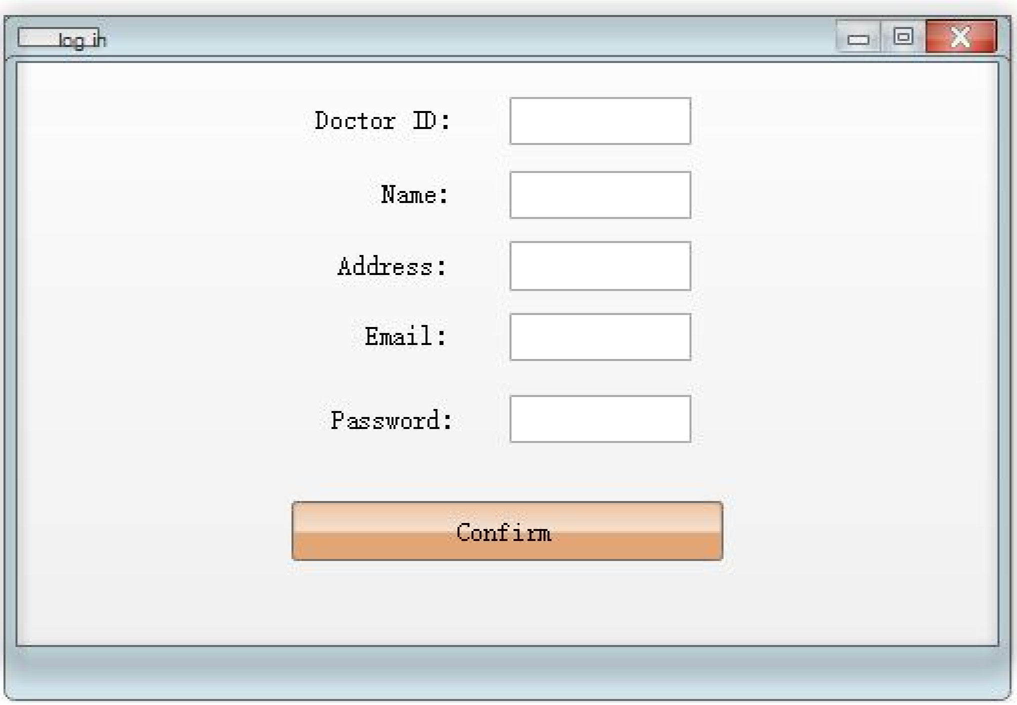
1:heart disease

2....

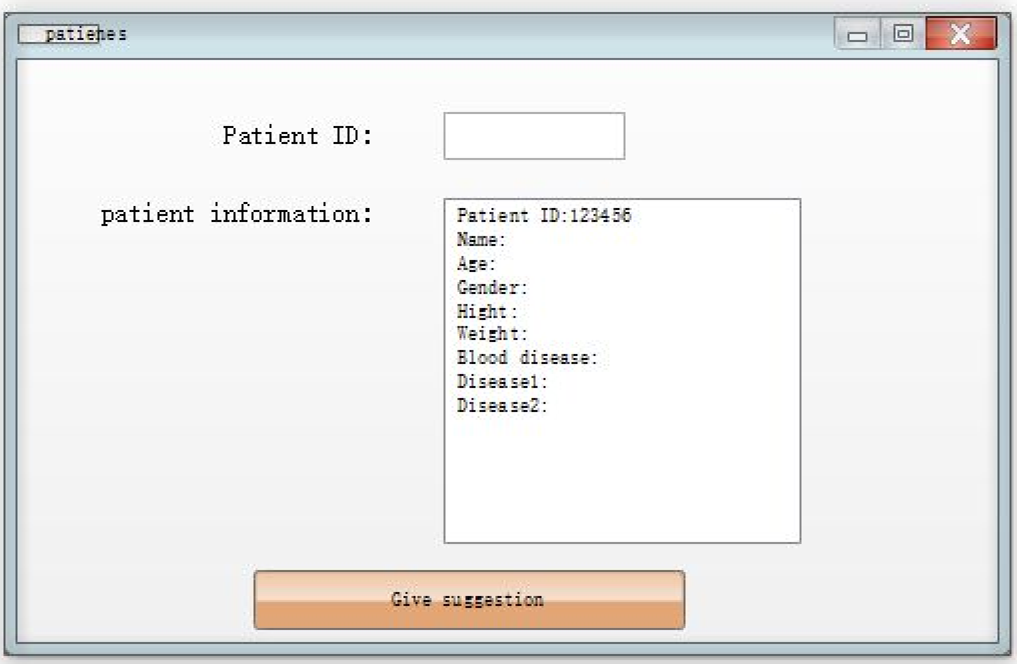
suggestion1:
heart disease:
(1)... (2)... (3)...

Suggstion2:
...:
(1)... (2)... (3)...

(8)doctor login page

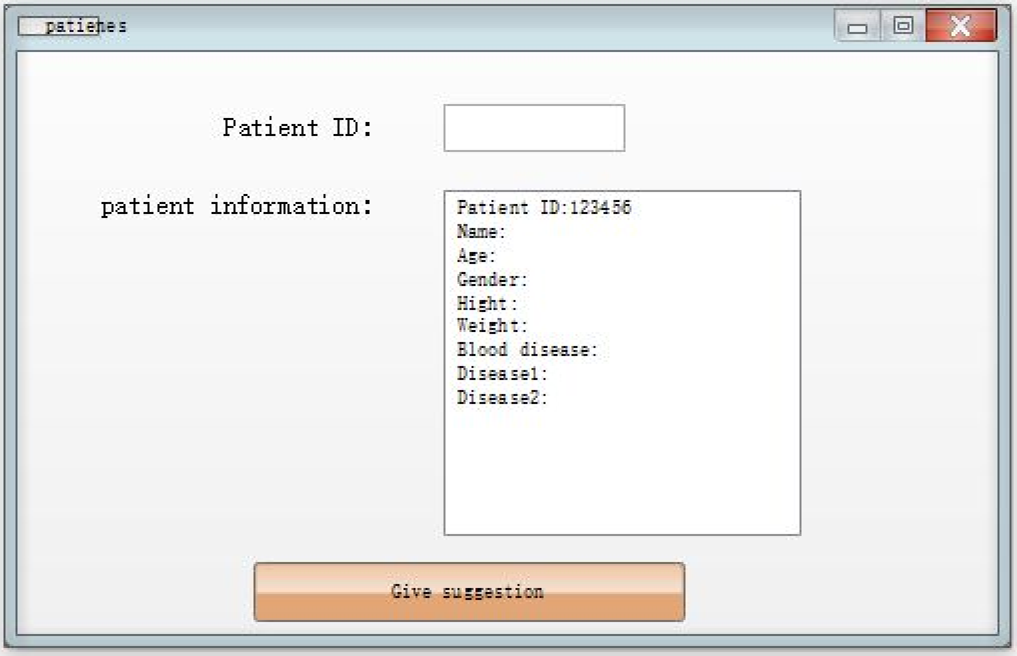


(9) Doctor search patients’ ID



Patient ID:123456
Name:
Age:
Gender:
Hight:
Weight:
Blood disease:
Disease1:
Disease2:

(10)Give suggestion



Patient ID:123456
Name:
Age:
Gender:
Hight:
Weight:
Blood disease:
Disease1:
Disease2:

6. Acceptance Criteria

Unique users: It should only be possible to create one user per email address. Attempts on making multiple user on the same account should get fail message and asked to use another email or log-in to the existing account.

- First time users must not be able do get a health analysis before then have been through the procedure of creating new account.
- Previous input data should be accessible by the user
- Two different users with the same health input must get the same prediction and suggestion result.
- Users should not get suggestions about health improvements which are very irrelevant
  - For example smoke less if the user doesn't smoke.
- Data collected about users should be traceable to unique user.
- User must be able to change information about them self.
  - Health data
    - e.g. Weight, height and diseases
  - General personal data
    - e.g.Email and address

- Deletion of account and collected data.
- User in the for of ‘doctor’ should be able to obtain data set about the patients heath features.

7. Non-functional Requirements (Quality Attribute)

(1) User interface and Human factors

The actual user of the product are middle age people who probably have little to medium skills in using computers. For them to use the application it should require no training and the input and out for the app should be explained in the interface. The interface should be very easy to use, else users might not choose the application due to frustration. Users don’t have to be very protected from making error because if they e.g. input wrong health data it is easy to change and will not cause many problems for the user or other.

The other type of users for the app is doctors. For them it require more skills to use, as they are apple to edit in suggestions and pull out data.

(2) Documentation

We will make a documentation of implementation, for each functions, we will create a small documentation to explain how do we achieve each functions and we will also declare the variables in the main branch to make it clear for other developers. Documentations for doctors and normal users will be created to guide them use the app. Also we will create the documentation of agreement of using our app which declares some policies and our responsibility of protecting personal information.The audience are doctors and normal users which are addressed by each document.

(3) Hardware Consideration

The app can be used on laptop or pc or smartphone, 500MB is enough to store the product and run it.

(4) Error Handling

If there is some input errors, system can reject and the user can also update his information by himself following instructions.

(5) Quality Issues

It requires the same python edition, the correct input of users’ information and exact algorithm which used to match user’s data with our disease dataset. The system maybe can trap faults but technically not all faults.

The acceptable down time per 24 hours period can be 1 hour.

The system can be portable if another computer has the same version of python and enough memory to store and execute the application. We are not sure if it’s possible to use on smartphone, so we specify our app as pc app.

(6) System Modification

Dataset, algorithms, computer speed are expected likely to be modified.

New data used to predict diseases will cause a modification of the algorithme finding risks of diseases user might have. This can also mean the input from the user will have to be modified.

(7) Security Issues

The access to user’s information must be controlled. Both the profiles and the data collected as they contain very personal information.

The system can be backed up once per day, and the developers are responsible for it.

(8) Resource and Management Issues

Source codes ,documents and data in a folder will be required to build, install, and maintain the system. The developer also needs the knowledge of programming, testing and analysing the errors. The final deadline for system development is almost Nov. 15th. There is a intermediate deadline the 30th of October, on this date the data analysis must be done . Users are responsible for system installation, however, developers are responsible for system maintenance.

8. Roles & Responsibilities

Suggestions & data collection = Mikkel Sinkjaer
Data analyzer= FENG, CHENLU
Programmer= WuTong
UI = Liu Tonghao
Tester = Xu Haobo

9. Acknowledgement

Authors and editors of the document: Feng Chenlu, Liu Tonghao, Mikkle Sinkjaer, Xu Haobo, Wu Tong

Section:

Feng Chenlu:	section 3
Xu Haobo :	section 2, section 4.A, section 4.C
Liu Tonghao:	section 5
Wu Tong:	section 2, section 4.A, section 4.B
Mikkle Sinkjaer:	section 6

Other sections are finished by discussion of all of us.