# 

# Software Requirements

Version 2, last updated by Jarvis at 2021-04-15

# Software Requirements Specification (SRS)

Revision History:

|  |  |  |
| --- | --- | --- |
| Date | Author | Description |
| 4-12-2021 | Jarvis Liu | ·Finish the background writing  ·Describe the system purpose briefly |
| 4-12-2021 | Lily Zhang | ·Complete the algorithm and customer’s requirement analysis  ·Finish the function requirement |
| 4-12-2021 | Egbert Ding | · Complete the Security and Flexibility module  ·Complete the Others module |
| 4-12-2021 | Bryce Shi | ·Write the performance requirement  ·Describe the data management capability |
| 4-15-2021 | Jarvis Liu | ·Introduce the intended audience  ·Elaborate the offered services  ·illustrate the concept of operations |
| 4-16-2021 | Bryce Shi | · Finish the behavioral requirements part  ·Complete the input and output requirements |
| 4-17-2021 | Egbert Ding | · Finish the quality requirements and fundamental assumption |
| 4-17-2021 | Gabriel Carneiro | · Insert use cases |
| 4-18-2021 | Ana Borges | · Insert use cases |
| 4-18-2021 | Lily Zhang | · Revise use cases |

Contents

[Software Requirements 1](#_Toc51163622)

[Software Requirements Specification (SRS) 1](#_Toc51163623)

[1.  Introduction 2](#_Toc51163624)

[1.1    Intended Audience and Offered Services 2](#_Toc51163625)

[1.2    How to use the document 2](#_Toc51163626)

[2.  Concept of Operations 3](#_Toc51163627)

[2.1    System Context 3](#_Toc51163628)

[2.2 System capabilities 4](#_Toc51163629)

[3.  Use Cases 4](#_Toc51163630)

[Case 1: Application Server Register new User4](#_Toc51163631)

[Case 2: Application Server upload new x-ray 5](#_Toc51163632)

[Case 3: Application Server Retrieve User’s Data 5](#_Toc51163633)

[Case 4: Application Server Delete User’s Data 6](#_Toc51163634)

[Case 5: Algorithm want to get Train&Test set 6](#_Toc51163635)

[Case 6: Algorithm gets a customer’s New Data 7](#_Toc51163636)

[Case 7: Store the Analysis Result from algorithm 7](#_Toc51163637)

[3.    Behavioral Requirements 11](#_Toc51163646)

[3.1 System Inputs and Outputs 11](#_Toc51163647)

[3.2 Detailed Output Behavior 13](#_Toc51163648)

[4.2   Quality Requirements 13](#_Toc51163649)

[5.    Expected Subsets 14](#_Toc51163650)

[6.    Fundamental Assumptions 14](#_Toc51163651)

[7.    Expected Changes 14](#_Toc51163652)

[8.    Appendices 15](#_Toc51163653)

[8.1    Definitions and acronyms 15](#_Toc51163654)

[8.2    References 16](#_Toc51163655)

## 1.  Introduction

### 1.1    Intended Audience and Offered Services

This document is designed to provide information guiding the operation on the database, guaranteeing that all system demands are understood and fulfilled. The server team may refer to this document to render mentioned information services to the algorithm and client team.

Services for the algorithm team:

* Train&Test set - The server team should offer all available labeled data from the database to the algorithm team for reading and using during model training and iterative update by operating “search\_record” on the whole records in TT Dataset Table.
* Customer’s New Data - The server team should send the new x-ray film uploaded by the customer to the algorithm team by calling “insert\_record” in Patient’s Record Table.
* Analysis Result - The server team should help store the data results get by the algorithm team into the Patient’s Record Table in the database, including the relevant marked pictures, the maximum angle, the date of diagnosis, etc by operating “update\_record” on Patient’s Record Table.

Services for the client team:

* Register - when a new customer register for the diagnosis, the server should store his/her information into the Users Table by calling “insert\_record” in Users Table. The information is basically about name, gender, age, etc.
* Upload new Data - When the client uploads the x-ray film,the server team should send the new data(including medical number, shooting time and the picture) to the database immediately by calling “insert\_record” in Patient’s Record Table.
* Search&Request - When the client is intended to search his own information, the server should send the data in the Users Table and Patient’s Record Table to him/her, which is calling “search\_record” by Name, Type or Time.
* Unsubscribe - When the client choose to unsubscribe from this service, the server should delete all the relevant information from the Users Table and the Patient’s Record Table by operating “delete\_record” on all the lines that matches the corresponding medical number.

### 1.2    How to use the document

Table of Contents:  
  
 1. Introduction – Guide the server team to render mentioned information services related to the constructed database to the algorithm and client team.

1. Concept of Operations - Descriptions about the faced problem and the solution to be implemented.  
    2.1 System Context - System requirements the application will require to run for diverse operating systems.  
    2.2 System Capabilities – Overview of all capabilities(functions) available to the server team.  
     2.3 Use cases - A detailed look at each functional requirement, describing the application context both before and after an operation is taken.

3. Behavioral Requirements - How will the application interact with a user.  
  3.1 Input and output requirements – A description of allowed inputs and generated outputs.  
    3.1.1 Input - Describes any restrictions that will be placed on allowed input.  
    3.1.2 Output - Describes the range of outputs that can be generated.  
  3.2 Detailed Output Behavior - Output descriptions in prose.

4. Quality Requirements - Requirements not pertaining to the function of the application will be listed here.

5. Expected Subsets - Expected levels of functionality at checkpoints during development.

6. Fundamental Assumptions - Some specifics about input, output, or behavior upon which other requirements are founded will be listed here.

7. Expected Changes - Future features and directions the project is expected to take.

8. Appendices - Details aiding the understanding of this document  
  8.1 Definitions and acronyms - Any technical terms or abbreviations will be spelled out here for ease while using the document  
    8.1.1 Definitions - Definitions of technical or unusual terminology.  
    8.1.2 Acronyms and Abbreviations - Any abbreviated terms will be expanded here.  
  8.2 References - any external references necessary or helpful to understanding this document will be listed here.(这部分每个分工的人来改）

## 2.  Concept of Operations

### In the process of scoliosis diagnosis/analysis, it is necessary to store a large amount of patient information and pictures of the spine. The traditional manual operation method is prone to data loss, statistical errors and slow speed.

### Our purpose is to create two Table in the constructed MYSQL database to store the relevant information of patients and and the data results get by the algorithm team to realize data sharing and centralized control and the server team to render mentioned information services(the kernel of the operation is about insertion, deletion ,update and search) to the algorithm and client team.

### 2.1    System Context

**System Requirements:**

**Support Windows/Mac OS X/Linux，The following runtime environments need to be installed and configured in advance。This system mainly uses java/python as the back-end application development tool, MYSQL as the back -end database and Windows as the system platform.**

Windows:

Windows 10/8/7, Python 3.0+, Pytorch1.1.0+, OpenCV-Python

### 

### 2.2 System capabilities

After finishing the database construction, the server can fulfill the requirements raised by the algorithm including fetching the train&test set, customer’s new data and storing the analysis result into the database and the needs of the client team including register, upload new data, search&Request and unsubscribe.

(more functions under exploit)

## Use Cases

|  |
| --- |
| **Use Case 1:** Application Server Register new User |
| **Brief description:**  Application server request for the database the registration of one user to the platform. |
| **Primary Actors:**  Application Server |
| **Preconditions:**  The Data Access Object service (DAO) and the DBMS are running and the Application Server is authorized to communicate with the DAO. |
| **Main flow:**   1. The Application Server request for the DAO the registration of a new user, sending the authorization token and the following information about the user that it wants to register: name, patient note identification, birth date, place of birth, gender, national identification, father’s name and mother’s name; 2. The DAO verify if the Application Server is authorized; 3. If the Application Server is Authorized    1. the DAO register the new user in the DBMS, returning the confirmation of registration and the new user identification in the platform; 4. Else    1. the DAO returns a error message, with a unauthorized error; |
| **Postconditions:**   1. The new user is registered in DBMS. |
| **Alternative flows:**  None. |

|  |
| --- |
| **Use Case 2:** Application Server upload new x-ray |
| **Brief description:**  Application server store in the database a new X-ray film. |
| **Primary Actors:**  Application Server |
| **Preconditions:**  The Data Access Object service (DAO) and the DBMS are running and the Application Server is authorized to communicate with the DAO. |
| **Main flow:**   1. The Application Server request for the DAO the registration of a new user, sending the authorization token the x-ray image and the user note identification referred to the user that the x-ray belongs; 2. The DAO verify if the Application Server is authorized; 3. If the Application Server is Authorized    1. the DAO store the new x-ray in the DBMS, returning the confirmation of the x-ray store in the database; 4. Else    1. the DAO returns a error message, with a unauthorized error; |
| **Postconditions:**   1. The x-ray is registered in DBMS. |
| **Alternative flows:**  None. |

|  |
| --- |
| **Use Case 3:** Application Server Retrieve User’s Data |
| **Brief description:**  Application server request results for a user in a determined period. |
| **Primary Actors:**  Application Server |
| **Preconditions:**  The Data Access Object service (DAO) and the DBMS are running and the Application Server is authorized to communicate with the DAO. |
| **Main flow:**   1. The Application Server request for the DAO the results of a patient, sending the follow informations: user’s note identification, start date and end date of the period of request; 2. The DAO verify if the Application Server is authorized; 3. If the Application Server is Authorized    1. the DAO response with the images of x-ray for the patient and the prediction about the x-rays uploaded in the period sent by the Application Server existent in database; 4. Else    1. the DAO returns an error message, with a unauthorized error; |
| **Postconditions:**   1. The x-rays and predictions are returned for the Application Server. |
| **Alternative flows:**  None. |

|  |
| --- |
| **Use Case 4:** Application Server Delete User’s Data |
| **Brief description:**  Application Server delete user’s account, deleting all the data related to specified patient from the DBMS. |
| **Primary Actors:**  Application Server |
| **Preconditions:**  The Data Access Object service (DAO) and the DBMS are running and the Application Server is authorized to communicate with the DAO. |
| **Main flow:**   1. The Application Server request for the DAO for deleting all information about one specified user, sending the authorization token and the patient note identification; 2. The DAO verify if the Application Server is authorized; 3. If the Application Server is Authorized    1. the DAO delete all the data referencing the patient; 4. Else    1. the DAO returns an error message, with an unauthorized error; |
| **Postconditions:**   1. All the data related to the specified user is deleted of database. |
| **Alternative flows:**  None. |

|  |
| --- |
| **Use Case 5:** Algorithm want to get Train&Test set |
| **Brief description:**  The server is available with all the data stored on the database. |
| **Primary actors:**  None. |
| **Second actors:**  None. |
| **Preconditions:**  None. |
| **Main flow:**   1. The use case begins with the data stored on the database are available on the server. 2. The algorithm sends a request to the server to access the data. 3. The server returns the data for the algorithm. 4. The algorithm has access to the data and it’s now available to read and testing data. |
| **Postconditions:**  1.All the data are available to the algorithm. |
| **Alternative flows:**  None. |

|  |
| --- |
| **Use Case 6:** Algorithm gets a customer’s New Data |
| **Brief description:**  The customer inserts a new x-ray for prediction by the algorithm. |
| **Primary actors:**  Customer. |
| **Second actors:**  Server. |
| **Preconditions:**  1.The customer must be registered on the platform. |
| **Main flow:**  1.The use case starts when the customer select the option to add new data.  2.The customer add a new X-ray .  3.The server sends the new X-ray for algorithm for prediction. |
| **Postconditions:**  1. The algorithm start to predict. |
| **Alternative flows:**  None. |

|  |
| --- |
| **Use Case 7:** Store the Analysis Result from algorithm |
| **Brief description:**  The new result predicted by the algorithm is stored on the database. |
| **Primary actors:**  Algorithm. |
| **Second actors:**  Sever. |
| **Preconditions:**  the result of the algorithm will be updated on specific customer’s record. |
| **Main flow:**  1.The use case begins when the algorithm do the prediction.  2.If the prediciton was executed successfully  a.The algortihm sends a successfully message, sent the result to the server and is stored on the database.  3.Else  a.The algorithm returns an error message. |
| **Postconditions:**  1.The new result predicted is stored on the database, available for consulting. |
| **Alternative flows:**  None. |

## 3.    Behavioral Requirements

### 3.1 System Inputs and Outputs

#### 3.1.1 Inputs

Inputs to the database come from the user(patient or doctor input through the server ). The user can creat, read, update, delete data in the database. The inputs include the train and test data , the data inputted by doctor/patient ,the result returned from the algorithm.

  Inputs When Use The Database:  
      \*General: At least two inputs will be required for a new contact. One of those two inputs must be a name. There is no length limit on any of the fields below.  
      \* Name: A contact can have a first name, or last name, or both. The first and last names will be separate fields. The system will accept any characters in the first and last name fields.

\* Gender: Gender should be male or female .  
      \* Age: Age that are accepted as valid by the system must be digits. The only acceptable inputs for a valid input is numerical characters ranging 0-9. If characters is not digits that it won't be succeed input.  
      \* Cobb Angle: If the patient without a scoliosis the input will be none, else it will be the Cobb Angle.  
     \*X-ray image:The size of inputed image should be limited whithin 16MB, user can input several images in one time.  
Importing:  
    A file must be in a very specific format for importing. The file must be of .tsv extension type.In that way ,you can input batch data one time ,such as input the train set and test set .  
  
 Patient names will not be subject to restrictions of length or character type. The name will not be required to match the filename that be saved to/opened from.  
Searches will not be restricted as far as input is concerned.

Input From The Algorithm:

\*Status: Whether the algorithm runs successfully.

\*Marked Image: The input can be image that size limited whithin 16 MB ,also can be the storage address or the url of the marked image.

\*Cobb:The Cobb Angle value.

#### 3.1.2 Outputs

The outputs are sent to the server and through server to the users or algorithm.  
 Outputs to The User:

Output to the user include name, gender,age, cobb angle, and the X-ray images.Doctor can view the patients’ information and the patients only can view themselves data.After the users input the X-ray images the system will output the result from algorithm , and the result will be saved in the database.  
      Creat/Update Information:  
      \* If the system receives incorrect/invalid input from the user at the creat and update, the system will warn the user that their input is invaild and the information will not been insert to the database. The user will need to let the system know when the input has been changed by re-clicking the confirm button.  
  
      Viewing User data:  
      \* Doctor can read the patients information in the database and the patient only can read themselves information(now and past). If the data they want does not exist ,the system will send a ‘can’t find’ to the user. The user can reinput or exit.  
       
      Delete:  
      \* Only the doctor can perform the delete operation .The user will be warned if they want to delete the data.After the data be deleted ,system will return a ‘Delete Success’,else will return a error.  
  
     Importing:  
       \* If any of the patient in an import file don't fulfill the minimum requirements to create a new patient, the user will be informed of how many patient failed after the import has finished.

Outputs to The Algorithm:

\*X-ray image: The X-ray image that the user inputted. The output can be image also can be the path or url of the image.

\*Command: Selecting the option on the spinal X-ray image to be computed.

-keypoint:Obtaining the labelling of key points for obtaining all bone masses for a given spinal X-ray.

-locate:Obtaining the key points of all the bone fragments and marking the position of the bone fragments constituting Cobb Angle.

-cobb：Obtaining the key points of all the bone fragments, marking the position of the bone fragments constituting Cobb Angle and Cobb value.

### 3.2 Detailed Output Behavior

The output varies from user to user and opreation to opreation.The data that get from the algorithm and from doctor will be saved in different table. And the user can see the result that more reliable, if the doctor’s form have the result ,the output will be the doctor’s [diagnosis](javascript:;) else system will return the algorithm’s result.

The user creat ,delete and update opreation will output a creat/delete/update success or fall, the read opreation will show the information that in the database, user can view the data [from](javascript:;) [the](javascript:;) [past](javascript:;) [to](javascript:;) [the](javascript:;) present so that they can know the illness get better or worse.

The output to the algorithm will send through the server, and the server will change the data to the form that the algorithm need.

### 4   Quality Requirements

The application must be competitive with similar applications in regards to performance, security ,confidentiality, flexibility and so on.  
  
Performance: Responsiveness to user input  
      \* Standard actions that manipulate address books or their contents should not exceed 500ms execution time.  
      \* Searching on address books is a possible exception to the above standard, as search performance will depend on the size of the address book.

Security and Confidentiality:Proetect the database and the accounts

\* The key application server and database server are separated to prevent users from directly operating the database server and ensure the database security.

\* In the software system, the data protection and business operation permission are completed by identifying the user's identity and authority. Compared with the user passord, if it is the same, the system will assign the user's operation authority to the user, and then the user will operate the system according to the assigned authority. To prevent the reveal of the uers password, in the process of transmission and the password record field in the database should not use plaintext transmission and storage, should use effective mainstream technology for the plaintext password before the password is transmitted, encrypt the transmission data with the encryption algorithm described in the encryption part, and transmit it to the system after encryption. The system compares the encrypted password data submitted by the user with the encrypted password saved by the user, and the subsequent operation will be carried out if they are consistent.

\* In this application, Different businesses are handled by different people, and the data that can be accessed by different operators are different. In order to satisfy this requirement, when the user logs in, the system obtains the user's permission after the user's authentication, and displays the corresponding function menu according to the user's permission. When the user browses the data after reading, writing and deleting, the system judges the user's access rights to the data and determines whether the operation is allowed.

Flexibility: Function flexibility and Operation flexibility

     \* The database is designed for more requirements in the future. To meet these probable functions, the database need to adopt a scalable and modifiable method to develop. When some new requirements adding to the system, we don't have to rebuild the whole database, but just to add a few interfaces and instances to realize the functions.

\* The database system should be compatible with different kinds of operation modes, ranging from common user to large-scale interface call. Different operation modes should be run properly in this database system, and the system is enable to extend more available operation modes in the future.

## 5.    Expected Subsets

L0:  
-Basic GUI with the ability to send data to and recall data from the backend storage  
-Ability to add address to address book  
-Ability to view an entry in an address book  
  
L1:  
-Address book capable of: storing complete entries, sorting, editting entries, and saving  
-Complete GUI for access to address book  
-Ability to import and export address books  
-Open recent

## 6.    Fundamental Assumptions

In order to make the application run properly, there are some advanced issues to take into consideration.

Environment support: the database system should run properly in prevalent operation systems and software environments, such as Linux, Windows 7++, IOS, Android.

Interface Principle: the database system should provide some interfaces for convenient large amounts of data analysis. The interface principles should be designed appropriately.

## 7.    Expected Changes

   Features to Add:  
      Import and Exporting Sets of Addresses  
      User Defined Fields  
      Links to Social Media  
      Deletion of Many Contacts at Once  
      Importing a Photo for Each Contact  
      View Contacts By Photo  
      Search Contacts By Photo  
      Voice-To-Text Searching  
      Ability to "Turn Off" Yellow Highlighting for Invalid Fields

   Future Platforms:  
      Write Front End For Andriod  
      Port Application to iOS

## 

## 8.    Appendices

### 8.1    Definitions and acronyms

#### 8.1.1    Definitions

|  |  |
| --- | --- |
| **Keyword** | **Definitions** |
|  |  |
|  |  |
|  |  |
|  |  |

#### 8.1.2    Acronyms and abbreviations

|  |  |
| --- | --- |
| **Acronym or**  **Abbreviation** | **Definitions** |
| GUI | Graphical User Interface |
|  |  |
|  |  |

### 8.2    References

《Beginning Database Design》

《Database design requirements analysis》

 这里应该还有client/algorithm那边的需求分析