# Software Requirements

Version 51, last updated by [rylandm](https://uocis.assembla.com/profile/rylandm" \t "_blank) at 2018-02-02

# Software Requirements Specification (SRS)

Revision History:

|  |  |  |
| --- | --- | --- |
| Date | Author | Description |
| 18-9-20 | Kaiyue Shi & Lingling Li | Adding/Editing Use Cases |
| 18-9-20 | Jiangying Xue | Behavior Requirements - Inputs |
| 18-9-20 | Kaiyue Shi & Lingling Li | Behavior Requirements - Outputs |
| 18-9-20 | Yan Zhao | Detailed output behaviors |
| 18-9-20 | Jinghan Lu | Quality requirements |
| 18-9-20 | Kaiyue Shi | Capabilities, Context |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Contents

[Software Requirements 1](#_Toc51163622)

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

[1.  Introduction 2](#_Toc51163624)

[1.1    Intended Audience and Purpose 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: Server submit the X-Ray photograph 4](#_Toc51163631)

[Case 2: Server Wants to do an angle calculation 5](#_Toc51163632)

[Case 3: Server Wants to classify the case of scoliosis 5](#_Toc51163633)

[Case 4: Server wants to migrate the model 6](#_Toc51163634)

[3.    Behavioral Requirements 11](#_Toc51163646)

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

[3.2 Detailed Output Behavior 13](#_Toc51163648)

[4.   Quality Requirements 13](#_Toc51163649)

## 1.  Introduction

### 1.1    Intended Audience and Purpose

This document is intended to provided information guiding the installation and development process, ensuring that all system requirements are met. The following entities may find the document useful:  
Primary Customer - This page will detail all of the application requirements as understood by the production team. The customer should be able to determine that their requirements will be correctly reflected in the final product through the information found on this page.  
User - A prospective user will be able to use this document to identify the main functionailty included in the application. Furthermore, the application will have a set of system requirements before the application can be run. Details regarding these requirements can be found here.  
Development Team - Details of specific requirements that the final software build must include will be located here. Developers can use this document to ensure the software addresses each of these requirements.  
QA Team - By developing testing procedures founded in the system requirements, the QA Team can create a comprehensive testing regimen that will guarantee requirements are met.

### 1.2    How to use the document

Table of Contents:  
  
1. Introduction  
2. Concept of Operations - broad description of the purpose of the application  
  2.1 System Context - details any specific system requirements the application will require to run  
  2.2 System Capabilities - description in prose of all capabilities available to the user in the address book  
  2.3 Use cases - A detailed look at each functional requirement, describing the application context both before and after an action is taken  
3. Behavioral Requirements - How the application will interact with a user  
  3.1 Input and output requirments - 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. Appendicies - Details aiding the understanding of this document  
  8.1 Definitions and acronyms - Any technical terms or abbreviations will be spelled out here for ease of use of the document  
    8.1 Definitions - Definitions of technical or unusual terminology  
    8.1.2 Acronyms and Abreviations - Any abreviated terms will be expanded here  
  8.2 References - any external references necessary or helpful to understanding this document will be listed here

## 2.  Concept of Operations

  The goal is to create a user friendly, standalone address book application. It will allow its users to save multiple address books each with contacts' names, addresses, emails, and phone numbers. A user must have the application installed and Java installed on their machine. The application uses drop down menus, text boxes, and file system navigation windows to interact with the user. For more details on the usage and capabilities of the application read the section, [System Capabilities](https://uocis.assembla.com/spaces/cis422w18-team2/wiki/Software_Requirements#System%20Capabilities).

### 2.1    System Context

**System Requirements:**  
We are responsible for the algorithm part of the whole system, which requires the call from the Server of the system, and sends back results to Server.

The calculation part of our algorithm is built on a deep learning based network, and requires computers with high computing performance.

### 2.2 System capabilities

 The algorithm implements three functions:

- angle calculation

- scoliosis classification

- module migration

## 3.  Use Cases

### Case 1: Server wants to process the X-Ray photograph.

**Players:** Server

**Goals:** The algorithm process and identify the X-Ray photograph.

**Preconditions:** The server transfer the photograph as parameters to the algorithm.

**Case:**

* 1. The algorithm module checks if the format is correct。
  2. The algorithm module changes the image of X-Ray photograph to a digital version .
  3. The algorithm module alters the image to adapt to the standard size.

**Alternative Flows:**

1.1 The photograph is failed to receive.

1.1.1 The format of X-Ray photograph is incorrect.

1.1.2 An empty photograph is received.

**Postconditions:**

A qualified digital image is prepared.

### Case 2: Server wants to do an angle calculations

Players: Server

Goals: The Server wants to do an angle calculation.

Preconditions: The algorithm module is ready and a digital X-Ray image is available.

Case:

* 1. The algorithm module extracts lines of bones from the image.
  2. The algorithm module marks every feature bones.
  3. The algorithm module calculates the angle of them and return the cobb angle to the Server.

Alternative Flows:

2.1.1 The algorithm module cannot identify any bones from the image.

2.1.1.1 The algorithm module returns with error and report it to the Server.

Postconditions:

The Server gets the cobb angle and marked bones from the X-Ray image.

### Case 3: Server wants to classify the case of scoliosis.

Players: Server

Goals: The Server wants to classify the case of scoliosis.

Preconditions: The photograph has been processed and results of cobb angles is available.

Case:

* 1. The algorithm module analyzes the cobb angles and classify the case of scoliosis.
  2. The algorithm module tells the Server whether this case is “C”, “S” or normal shape.

Alternative Flows:

3.1.1 the feature of cobb angles is irregular.

Postconditions:

The Server is now aware of the classification of scoliosis in this case.

### Case 4: Server wants to migrate the model.

Players: Server

Goals: The Server wants to migrate the model.

Preconditions: The algorithm module is working and a network model is available.

Case:

* 1. The algorithm module extracts the model of this case and output to a file.

Alternative Flows:

* + 1. The output file is empty.

4.1.1.1 The algorithm module returns with error and report it to the Server.

Postconditions:

The Server gets the model file of this case.

## 3.    Behavioral Requirements

### 3.1 System Inputs and Outputs

#### 3.1.1 Inputs

Inputs to the algorithm come from the server.

Inputs When server transfer the X-Ray photograph:  
      \*image: a X-Ray photograph from server  
 Inputs When server calls an angle calculation:  
      \*bones: two of the bones which from the image from server

 Inputs When Server wants to classify the case of scoliosis:  
      \*signal: classify the case of scoliosis signal from server

 Inputs When User wants to migrate the model:  
      \*model:  migrate the model signal

#### 3.1.2 Outputs

The whole algorithm module returns mainly three outputs to the server, and another outputs between cases of the algorithm.

   Outputs to other cases from case1:  
      Digital image of X-Ray photograph with format that meets algorithm requirements.

Outputs to Server:

      Figure of cobb angles:

\* If the algorithm module calculates the expected results of cobb angle, then returns a figure to server to show how cobb angles of bones distributed.  
  
      Class of scoliosis:  
      \* If the algorithm module is able to classify the cobb angle features into known type(C, S or normal), returns the class of scoliosis to server.

      Network Module:  
      \* If the server wants to migrate the module our algorithm has built, pack it to a file and returns to server.

Exception warning:  
      \* If any input of each case is not fulfilled the minimum requirements to execute the following calculation, throw an exception to server.

### 3.2 Detailed Output Behavior

An angle calculation：

From server,we can get the X-Ray image from server,scan processing and calculation,at last,output the results to the server

classify the case of scoliosis:

first,get the X-Ray image from server,then use the angle calculation to scan processing and calculation,give the calculation results,Compare the calculated value with the standard value and output the result to the server.

migrate the model:

after the model migration signal is obtained from the server,The algorithm outputs the trained network into a model file and returns the file to the server.

### 4.2   Quality Requirements

The algorithm must be competitive with similar algorithms in regards to performance, reliability, consistency, and scalability.  
  
Performance: Responsiveness to user input  
      \* Standard actions that scan the X-ray photograph, gets the image of X-Ray photograph in digital version from the scanner,angle calculation,scoliosis classification and model migration should not exceed 500ms execution time.

Reliability: Actions taken will not cause errors and the accuracy of angle calculation and scoliosis classification should be up to 90 percent.

      \* User’s input error should not cause effect to the whole algorithm progress.

      \* Any modifications to the model base should produce a lasting change that persists through any following series of actions taken by the user.

\*The scale of the training set must be big enough.

\*The error rate compared to other algorithms should be low to 5 percent.

Consistency: Persistent data in model base

      \*Model should be modifiable after being migrated in such a way that target fields can be changed without affecting data in other fields.

Scalability: Ease of extending algorithm capabilities  
      \* The algorithm should be able to extend other capabilities or solve the problem in other situations.For example,the image is not clear enough that the algorithm can’t calculate the angle correctly or some new kind of scoliosis appears,the algorithm can also classify the scoliosis correctly.