# Software Requirements

Version 1.7, last updated by Yue Hong at 11-07-2020

# Software Requirements Specification (SRS)

Revision History:

|  |  |  |
| --- | --- | --- |
| Date | Author | Description |
| 9-18-20 | Yue Hong | Adding/Editing Use Cases, and Gantt graph |
| 9-18-20 | Guangpeng Li | Updating Gautt graph |
| 9-30-20 | Yue Hong | Modifying function details after the first RA’s meeting |
| 10-07-20 | Yue Hong | Adding use case diagram |
| 10-14-20 | Yue Hong | Modifying use case diagram |
| 10-30-20 | Yue Hong | Adding new requirements |
| 11-06-20 | Yue Hong | Final modifying and checking details |

Contents

[Software Requirements 1](#_Toc51365726)

[Software Requirements Specification (SRS) 1](#_Toc51365727)

[1.  Introduction 3](#_Toc51365728)

[1.1    Intended Audience and Purpose 3](#_Toc51365729)

[1.2    How to use the document 3](#_Toc51365730)

[2.  Concept of Operations 4](#_Toc51365731)

[2.1    System Context 4](#_Toc51365732)

[2.2 System capabilities 5](#_Toc51365733)

[3.  Use Cases 5](#_Toc51365734)

[Case 1: User wants to scan the X-Ray photograph 5](#_Toc51365735)

[Case 2: User wants to do an angle calculation 5](#_Toc51365736)

[Case 3: User wants to classify the case of scoliosis. 6](#_Toc51365737)

[Case 4: User wants to migrate the model. 6](#_Toc51365738)

[Case 5: User wants to compare the results with other algorithms. 6](#_Toc51365739)

[**3.1 System Inputs and Outputs** 7](#_Toc51365740)

[**3.1.1 Inputs** 7](#_Toc51365741)

[**3.1.2 Outputs** 7](#_Toc51365742)

[**3.2 Detailed Output Behavior** 7](#_Toc51365743)

[4   Quality Requirements 7](#_Toc51365744)

[5.    Fundamental Assumptions 7](#_Toc51365745)

[6.    Expected Changes 7](#_Toc51365746)

[7    References 8](#_Toc51365747)

## 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:

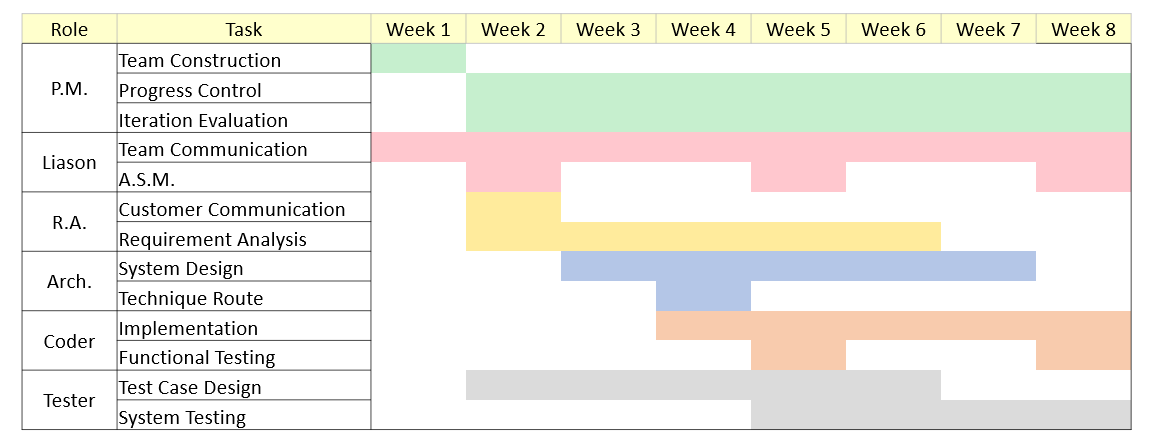
Primary Customer (mainly orthopedists, or potential patients who need self-diagnosis) - This page will detail all of the application requirements as understood by the production team. The customer should be able to determine that their requirements will be correctly reflected in the final product through the information found on this page.

User - A prospective user will be able to use this document to identify the main functionality 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.

Coders - Details of specific requirements that the final software build must include will be located here. Coders can use this document to ensure the software addresses each of these requirements.

Testers - By developing testing procedures founded in the system requirements, testers can create a comprehensive testing regimen that will guarantee requirements are met.

Estimated Gantt Graph:



### 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  
   3.Use cases - A detailed look at each functional requirement, describing the application context both before and after an action is taken  
   4. Quality Requirements - Requirements not pertaining to the function of the application will be listed here  
   5. Fundamental Assumptions - Some specifics about input, output, or behavior upon which other requirements are founded will be listed here  
   6. Expected Changes - Future features and directions the project is expected to take  
   7. References - any external references necessary or helpful to understanding this document will be listed here

## 2.  Concept of Operations

The goal is to create an effective algorithm to diagnose scoliosis using artificial intelligence. Through the process of ct images analysis, the algorithm will calculate the cobb angel and provide a valid diagnosis of high accuracy. There are strict requirements on the format of the input.

### 2.1    System Context

**System Requirements:**  
Requires a system with a GUI display because all of the operations are performed through a GUI. The application is in Java so users must have an updated version of Java installed on their machine to use the application.

Windows:

* Windows 10 (8u51 and above)
* Windows 8.x (Desktop)
* Windows 7 SP1
* Windows Vista SP2
* Windows Server 2008 R2 SP1 (64-bit)
* Windows Server 2012 and 2012 R2 (64-bit)
* RAM: 1G (Minimum)
* Disk space: 4G (Minimum, for data and others)
* Processor: Minimum Pentium 2 266 MHz processor

Android:

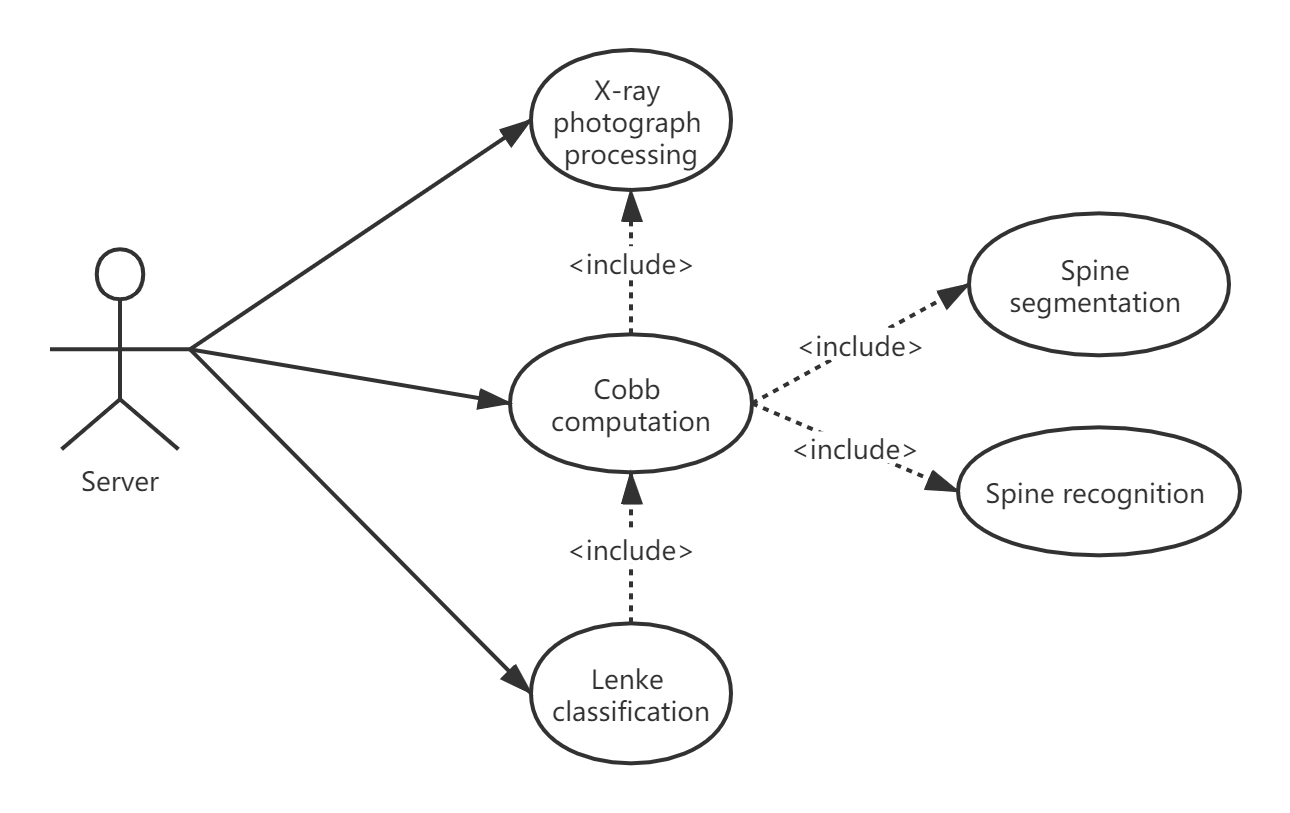
* SDK: API 4 (Minimum)

### 2.2 System capabilities

Call this program, one can get the diagnose results of the ct image. This program will provide suggestions on whether the patient’s condition belongs to scoliosis, and the calcification of it, the angle of scoliosis, and other useful information.

## 3.  Use Cases

Use case diagram:



### Case 1: Server wants to do X-ray photograph processing

**Players**:

Server

**Goals**:

Input the photograph successfully from the server and ensure it is ready for further processing.

**Preconditions**:

The server transfers the X-ray photograph to the algorithm.

**Case**:

* 1. Check the format of the photograph, whether it is acceptable.
  2. Do the preprocessing of the photograph.

**Alternative Flows:**

* 1. Report receiving error to the server if the algorithm fails to receive the photograph.
  2. Report format error if the format of the photograph is unacceptable.
  3. Report processing error if there is anything wrong during the preprocessing.

**Postconditions:**

The algorithm successfully opens a digital image corresponding to the X-Ray photograph. And it is ready for further operations.

### Case 2: Server wants to calculate the cobb angel

**Players**:

Server

**Goals**:

Return the result of cobb angle calculation to the server.

**Preconditions**:

Preprocessing of the photograph goes well.

**Case**:

1. Do spine segmentation.

2. Extracts lines of bones in the photograph.

3. Do spine recognition and identification.

4. Calculate the cobb angel according to the rule.

5. Return the result to the server.

**Alternative Flows**:

1.Report recognition error if recognition fails.

2.Report calculation error if the result of calculation is unacceptable.

**Postconditions:**

The server receives the expected cobb angle and the marked photograph.

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

**Players**:

Server

**Goals**:

Returning the scoliosis classification result to the server.

**Preconditions**:

Cobb angel is successfully calculated, and the segmentation and recognition go well.

**Case**:

1. Analyze the cobb angel and classify the case of scoliosis according to the rules.

2. Returning the result of the classification (“C”,” S”, or normal) to the server

**Alternative Flows**:

1. Report classification error to the server if the cobb angel is irregular and the result is unacceptable.

**Postconditions**:

The server receives the classification of scoliosis in this case.

**3.1 System Inputs and Outputs**

**3.1.1****Inputs**

The input to the algorithm comes from the server.

Input: when the server transmits the x-ray photo.

Image: the x-ray photo from the server.

Input When the server invokes the angle calculation.

Bones: two bones from the server's image.

Input When the server wants to classify a scoliosis condition.

Signal: the scoliosis classification signal from the server.

Input When the user wants to migrate the model.

**3.1.2 Outputs**

The entire algorithm module returns two main outputs to the server and one output to the algorithm between cases.

Output from case 1:

A digital image of the x-ray photograph in a format that meets the algorithm's requirements.

Plot of Cobb angle:

If the algorithm module calculates the expected result for the cobalt angle, it returns a number to the server showing the distribution of the cobalt angle of the skeleton.

Scoliosis classification:

If the algorithm module is able to classify the cobb angle feature as a known type (C, S, or normal), the grade of scoliosis is returned to the server.

Exception Warning:

Throws an exception to the server if any of the inputs in each case do not meet the minimum requirements to perform the following calculations.

**3.2 Detailed Output Behavior**

Cobb angle:

We can acquire X-ray images from the server, process the scans and calculations, and finally output the results to the server.

Scoliosis classification:

First, the X-ray image is acquired from the server, then the scan is processed and calculated using angle calculations, the results are given, the calculated values are compared with the standard values, and the results are output to the server.

### 4   Quality Requirements

The product will fill a gap in the market, therefore there is no competition exists.

The algorithm must be competitive with Team A1 and Team A2’s algorithms in terms of performance, reliability, consistency, and scalability.

Since the problems involved in this program are relatively new, the requirements of this program are mainly for reference.

Performance: Responsiveness to user input

 \* Standard actions that should not exceed 9000ms execution time.

## 5.    Fundamental Assumptions

There are strict requirements on the format of the input image.

The training set of this program is known.

The caller of this program will meet all the hardware requirements of this program.

The caller of this program uses the same token and semantic system as this program.

## 6.    Expected Changes

   Features to Add:

   Future Platforms:  
     Port Application to iOS