|  |  |  |
| --- | --- | --- |
| **Protein Analysis Using PCA** | | |
| Team Information | 1. Seth Tucker 2. Ted Green 3. Daxtyn Hiestand 4. Zach Bockskopf 5. Joseph Hallmark | |
| Document Type: Software Requirement Specification v1.0 | | *Last updated on*: 09-14-2017 |

**Document Approval**

This Software Requirements Specification has been accepted and approved by the following stakeholders:

|  |  |  |
| --- | --- | --- |
| **Printed Name** | **Title** | **Signature and Date** |
|  |  |  |
|  |  |  |
| R. IQBAL | Customer/Course Instructor |  |

**Table of Contents**

[**1. Introduction**](#_30j0zll) **4**

[1.1 Purpose](#_1fob9te) 4

[1.2 Scope](#_3znysh7) 4

[1.3 Definitions, Acronyms, and Abbreviations](#_2et92p0) 4

[1.4 References](#_tyjcwt) 5

[1.5 Overview](#_3dy6vkm) 5

[**2. General Description**](#_1t3h5sf) **5**

[2.1 Product Perspective](#_4d34og8) 5

[2.2 Product Functions](#_2s8eyo1) 6

[2.3 User Characteristics](#_17dp8vu) 6

[2.4 General Constraints](#_3rdcrjn) 6

[2.5 Assumptions and Dependencies](#_26in1rg) 6

[**3. Specific Requirements**](#_lnxbz9) **7**

[3.1 Functional Requirements](#_1ksv4uv) 7

[FR.1 Image acquisition](#_lkaimy6qyizd) 7

[FR.1.1 Student mode](#_kyy547g75ase) 7

[FR.1.1 Professor mode](#_fzqssg4bnnc8) 8

[FR.1.1 Monochrome or color](#_u6l4ishowqph) 8

[FR.2 Image Parsing](#_oivoe76gfpoy) 8

[FR.2.1 Area selection](#_30xzczo1a25o) 8

[FR.3 Data Processing: Reference File](#_dhn4g4n0zyem) 9

[FR.4 Reference File Importing](#_l18wmdhtojj0) 10

[FR.5 Data Processing: Student Input](#_cylpivs3g6ma) 11

[3.2 Non-Functional Requirements](#_2jxsxqh) 12

[NFR.1 Time limits](#_5h4a4mepmqy1) 12

[3.3 Design Constraints](#_z337ya) 13

[3.4 External Interface Requirements](#_3j2qqm3) 13

[3.4.1 User Interfaces](#_1y810tw) 13

[EIR.1 Android Devices](#_fph02477xsgc) 13

[3.4.2 Hardware Interfaces](#_4i7ojhp) 13

[EIR.2 Mobile Device Camera](#_yfrpwvn64g25) 13

[3.4.3 Software Interfaces](#_vc6chcg80dlo) 13

EIR.3 Mobile OS 13

3.4.4 Communications Interfaces 14

EIR.4 Cloud Platform 14

[3.5](#_zf3q68xxdm6o) Logical Database Requirements

LDR.1 N/A

3.6 Other Requirements 14

[4. Change Management Process](#_2bn6wsx) 14

[Appendices](#_qgd80d7n02p) 14

# 1. Introduction

## 1.1 Purpose

This SRS document is intended to inform the developers about the requirements, properties, and goals that our group has created. This document is subjected to change with any future ideas that the group comes up with. This document is intended to inform the developers about the requirements of the product in sufficient detail.

## 1.2 Scope

The product will allow users to take images of protein trays and use Principal Component Analysis (PCA) to reduce the data to allow the results to be viewed in 3 dimensions. The color of each well on the protein tray will be averaged and used as a data point in an element representing the image.

Instructors will be able to use multiple images to create a reference file that will then be downloadable by students for comparison to data they gather. The product will give the closest match and a percentage confidence in match metric based upon distance between input and closest reference point. The reference file and results of user input can be graphed and viewed in 3D.

## 1.3 Definitions, Acronyms, and Abbreviations

The Definitions, Acronyms, and Abbreviations for this SRS document are:

* Principal Component Analysis (PCA) - A linear algebra method that can reduce data sets of higher dimensions into 3D while preserving a great amount of useful information.
* ALGLIB - A cross platform open source numerical analysis and data processing library.
* Eigenvectors - A linear algebra non-zero vector whose direction does not change when a linear transformation is applied to it.

## 1.4 References

There are no references for this document at this time.

## 1.5 Overview

The rest of the SRS contains the following:

* A General Description of the general factors that will affect the product and it’s requirements these include:
  + Product Perspective
  + Product functions
  + User Characteristics
  + General Constraints
  + Assumptions and Dependencies
  + Function and Non Functional requirements
  + Design Constraints
  + External Interface Requirements
  + Other Requirements
  + Change Managements Process
  + Appendices

# 2. General Description

This product is intended as an alternative to bulky lab equipment used in chemical analysis. It is intended to be a less accurate, but more accessible alternative, as the spectral graph machine is expensive, where as most people will already have smartphones.

## 2.1 Product Perspective

The spectral graph machine is the only mainstream alternative. It is large, and expensive, often outside of the budget that a high school can spare.

## 2.2 Product Functions

The product will have the following functions:

* The program will take pictures of protein trays. Using these images, the program will allow the user to select areas to gather information. Then an area’s color will be averaged and used as a data point in an element representing the image.
* Instructors will use multiple images to create a reference file.
* Students will be able to import the reference files created by the instructors.
* The program will allow students to compare images they take against the reference file. A closest match will be given with a percentage confidence in match metric based upon distance between input and closest reference point.
* The program will allow the user input data and the reference data to be graphed and viewed in 3D.

## 2.3 User Characteristics

Users include: High School students with limited technical and mathematical knowledge and chemistry instructors. Users will be limited to smartphone devices.

* High School Students
  + Able to access basic functions such as taking pictures and creating result files
* Professors
  + Full access to entire program
  + Be able to create and upload reference files.

## 2.4 General Constraints

Some general constraints pertaining to this project include:

* A limited amount of time is allocated to complete this project.
* The product software must be smartphone compatible.
* A major limitation is the fact that the program will require a significant amount of accuracy by the user in taking the pictures. Things such as lighting, shadows, and improper alignment could potentially disrupt the program and cause it to return false or incorrect data. The user will need to know how to use the program correctly.
* When selected areas for comparisons, users must make sure that they select their reference areas in the same manner for each picture
  + This means areas must be the same size and selected in the same order to get a clean comparison
* Student Mode Constraints
  + Limited to only exporting reference files
  + Limited to making only one comparison before
* Professor mode Constraints
  + Must take 3 or more pictures to create a reference file
  + Must take at least 3 areas per image
  + Each image must have the same number of areas
  + Must make a label for each picture taken so that comparisons can be easier to identify

## 2.5 Assumptions and Dependencies

The assumptions and dependencies of this project are as follows:

* Mobile Device (Android OS).
* Devices will be able to take pictures.
* ALGLIB compatibility.

# 3. Specific Requirements

# 3.1 Functional Requirements

### FR.1 The System shall be able to take pictures. ( Image acquisition)

This is a fundamental feature of the program, and therefore is a priority of 1

### **FR.1.1 The** system shall have a mode for students. (**Student mode)**

In student mode, the program should allow the user to take one picture using the device's camera.

**Source:** Customer

**Priority:** 1

**Introduction:**

**Inputs:** Camera Data

**Processing:** The program should take the data from the camera and form a bitmap

image from it.

**Outputs:** Image Data

**Error Handling:** If the app is unable to take an image, then a message box

should warn the user of the error.

### FR.1.2 The system shall have a mode for professors. (Professor mode)

In professor mode, the program should allow the user to take multiple pictures using the device's camera.

**Source:** Customer

**Priority:** 1

**Introduction:**

**Inputs:** Camera Data

**Processing:** The program should take the data from the camera and form a bitmap image from it.

**Outputs:** Image Data

**Error Handling:** If the app is unable to take an image, then a message box should warn the user of the error.

### ~~FR.1.3a Monochromatic Processing~~

~~The system shall allow the user to select the mode in which the image will be processed in monochrome~~

**~~Source:~~** ~~Other parts of the system~~

**~~Priority:~~** ~~3~~

**~~Introduction:~~**

**~~Inputs:~~** ~~Image Data~~

**~~Processing:~~** ~~This should be a method with a boolean and if true, Program will convert colored images take to monochrome (grayscale)~~

**~~Outputs:~~** ~~Image Data~~

**~~Error Handling:~~**

**~~FR.1.3b Color Processing~~**

~~The system shall allow the user to select the mode in which the image will be processed in color~~

**~~Source:~~** ~~Other parts of the system~~

**~~Priority:~~** ~~3~~

**~~Introduction:~~**

**~~Inputs:~~** ~~Image Data~~

**~~Processing:~~** ~~This should be a method with a boolean and if false, Program will will leave the image as is~~

**~~Outputs:~~** ~~Image Data~~

**~~Error Handling:~~**

### FR.2 The system will parse the image. (Image Parsing)

This is a fundamental feature of the program, and therefore is a priority of 1

### FR.2.1 The system will allow user to choose choose areas. (Area selection)

Once an image is taken, the user should be allowed to select areas on the image they wish to take data from. Each of these areas will be averaged and turned into a data point.

**Source:** Team

**Priority:** 1

**Introduction:** This will take multiple areas defined by the user and average the color values found within that area. This will create a set of 1 or more data points for an image.

**Inputs:** Image Data, user input (selection of areas)

**Processing:** This should take the input image and a set of areas. The areas will then be given a result that is the average color inside of that area. This will be done for every area defined

**Outputs:** List of data points (numbers)

**Error Handling:** Each image after the first one should have the same number of areas. The student mode should also have the same number of areas as defined by the reference file. Tuples need to be of at least four (4) dimensions or greater for the PCA core to function.

**FR.2.2 The system will put data into tuples. (Tuple formation)**

When the areas are selected, and each area’s color has been averaged, the data points will be considered a tuple. The user can define the order of the data points so the analysis will remain accurate.

**Source:** Other Parts of the Program

**Priority:** 1

**Introduction:** This is a place for a warning to the users that the ordering of the tuple needs to be the same throughout the entire data set.

**Inputs:** Data points, user input (order of the points)

**Processing:** This requirement will reorder the data points and form them as a tuple.

**Outputs:** Tuple data of the image

Error Handling: If the reference file is found, or in professor mode, the number of data points should be the same for all tuples.

**FR.2.2 The system will allow user to put multiple tuples into a set. (Set Formation)**

In professor mode the user has the option to have created multiple tuples, if there are multiple tuples they are allowed (and encouraged) to be able to label each one.

**Source:** Other Parts of the Program

**Priority:** 1

**Introduction:** This is a place to warn the user (professor) that it is highly encouraged to name each tuple so that when the results screen is used by students they can have meaningful feedback.

**Inputs:** Tuples, user input(names for each tuple)

**Processing:** This requirement will attach a label to each tuple.

**Outputs:** Tuple data of the image

**Error Handling:** If the reference file is found, or in professor mode, the number of data points should be the same for all tuples.

### FR.3 The system will perform data processing. (Data Processing: Reference File)

This is a fundamental feature of the program, and therefore is a priority of 1. However, it is unknown at this time if we will be using the device to do the calculations or not. We intend to explore using a networked server to do the calculations.

**FR.3.1 The system will process the professor data set. Process Professor Data Set**

Once the user has created a set of data points, this data will be passed to the PCA core.

**Source:** Other Parts of the Program

**Priority:** 1

**Introduction:** This is the requirement of passing the user created data points to the PCA core. The core will then produce internal data used to create the reference file.

**Inputs:** Data set from user.

**Processing:** The PCA core will perform Principal Component Analysis(PCA).

**Outputs:** Internal Data for next requirement

**Error Handling:** If the reference file is found, or in professor mode, the number of data points should be the same for all tuples.

**FR.3.2 The system will form a reference file. Reference File Formation**

After the PCA core finishes processing it will be ready to form a reference file.

**Source:** Other Parts of the Program

**Priority:** 1

**Introduction:** The PCA core will form the reference file for a passed data set. It’s format will be

The 3 eigenvectors/Principal Components

The set translated via the principal components

the color mode the set was created in.

**Inputs:** PCA core internal data

**Processing:** Formatting the data into a text file(or other file type)

**Outputs:** Reference file

**Error Handling:**

**FR.3.3 The system will have the ability to save reference files. Reference File Saving**

Once the reference file has been returned to the user(professor) they need to be able to save it. This will allow them to distribute it to their student users.

**Source:** Other Parts of the Program

**Priority:** 1

**Introduction:**

**Inputs:** Reference File from PCA core

**Processing:** The device has a file system that can be accessed.

**Outputs:** Reference file as file on user device.

**Error Handling:** If the file fails to save alert the user to try again.

### FR.4 The system will allow reference file importing. Reference File Importing

The professors reference file will be imported in student mode. The reference file will be used to determine two factors in the student mode. The number of data points per tuple, and the color mode of the image parsing.

**FR.4.1 Reference File Importing**

The user (student) needs to import a reference file before they can take an image. This file is the reference file created and provided by their professor. Thus they need to be able to search for and import the file from the devices file system.

**Source:** Other Parts of the Program

**Priority:** 1

**Introduction:**

**Inputs:** Reference file provided by professor

**Processing:** Access the devices file system

**Outputs:** Internal data of the reference file

**Error Handling:** If the file fails to load alert the user to try again.

**FR.4.2 Enforce Consistency With Reference File**

Once the reference file is imported by the user (student), the picture mode needs to be forced into the color mode specified by the file. Also when selecting areas, they will have to select the same number as the reference file. (both the student tuple and the reference file tuples will have the same number of dimensions.

**Source:** Other Parts of the Program

**Priority:** 1

**Introduction:** This is also a place to warn the users (both professor and student) that they should try and keep as much consistent as possible, both in how the pictures are taken and how the areas are selected.

**Inputs:** Reference file provided by professor

**Processing:** Match the color mode and enforce a number of areas needed.

**Outputs:** Internal data of the reference file

**Error Handling:** If the file fails to load alert the user to try again.

### FR.5 Data Processing: Student Input

Once the user (student) has both imported a reference file and selected the areas of their image to form a tuple, the system is ready to create their results screen for their input.

**FR.5.1 Combine Information From Student Input and Reference File**

Once the student has created their tuple from an image, it's data along with the the reference file will be sent to the PCA core.

**Source:** Other Parts of the Program

**Priority:** 1

**Introduction:**

**Inputs:** Reference file and user (student) selected areas

**Processing:** Send to PCA core

**Outputs:** Internal data in the PCA core

**Error Handling:**

**FR.5.2 Generate Results File**

With the information from the reference file and the user input, the PCA core can generate a results file for the user (student).

**Source:** Other Parts of the Program

**Priority:** 1

**Introduction:** The PCA core will translate the user input via the three principal components defined in the reference file. It will then then file the closest match in the reference data set and give a closest match result with % of distance away/confidence.

**Inputs:** Reference file and user (student) selected areas

**Processing:** PCA translation of user input and measuring distance to reference data set.

**Outputs:** Results file

**Error Handling:**

### ~~FR.6 Student Results~~

~~With the information from the reference file and the user input, the PCA core can generate a results file from user (student)~~

**~~Source:~~** ~~Other parts of the program~~

**~~Priority:~~** ~~1~~

**~~Introduction:~~**

**~~Inputs:~~** ~~Results file~~

**~~Processing:~~** ~~display the data in the user file~~

**~~Outputs:~~** ~~Display~~

**~~Error Handling:~~**

### ~~FR.7 Results and Visualizations~~

~~With the reference file the data set should be able to be graphed in 3D space. If a results file also exists it can be graphed as well.~~

**~~FR.6.1 Student Results~~**

~~With the information from the reference file and the user input, the PCA core can generate a results file for the user (student).~~

**~~Source:~~** ~~Customer~~

**~~Priority:~~** ~~1~~

**~~Introduction:~~**

**~~Inputs:~~** ~~Results file~~

**~~Processing:~~** ~~Display the data in the results file~~

**~~Outputs:~~** ~~Display~~

**~~Error Handling:~~**

## 3.2 Non-Functional Requirements

### NFR.1 Time limits

These are the time limits each of the major functions of the program. they have to complete their respective tasks before an error message is displayed due to going over the time limit.

**NFR.1.1 Student Result Time**

Once a picture has been taken by the user(student), the program shall return a result file to the user(student) in no more than 20 seconds.

**Source:** Other parts of the program

**Priority:** 1

**Introduction:**

**Inputs:** Picture taken by user

**Processing:** Scans picture taken with reference files to find a match

**Outputs:** Result with name of protein arrangement

**Error Handling:** If file fails to process within the time limit, prompts user to try again

**NFR.1.2 Reference Upload Time**

When uploading reference file to the program, it will take no more than 20 seconds to complete the upload.

**Source**:

**Priority:** 1

**Introduction:**

I**nputs:** Reference file

**Processing:** Takes and uploads a reference file created by a professor

**Outputs:** Reference file is now loaded onto the program for use

**Error Handling:** If file fails to process within the time limit, prompts user to try again.

## 3.3 Design Constraints

* Program will be developed in the C# language
* Program will be developed using the Xamarin libraries and platform
* Program will be designed using Visual Studio
* Using an android emulator, the app shall be tested before being finalized.

## 3.4 External Interface Requirements

### 3.4.1 User Interfaces

### EIR.1 Android Devices

**Source:** Customer

**Priority:** 1

**Introduction:** The program will be developed for the android device using Xamarin.

**Inputs:** n/a

**Processing:** n/a

**Outputs:** n/a

**Error Handling:**

### 3.4.2 Hardware Interfaces

### EIR.2 Mobile Device Camera

**Source:** Customer

**Priority:** 1

**Introduction:** The software will use mobile devices camera.

**Inputs:** n/a

**Processing:** n/a

**Output:** Picture of the data

**Error Handling:** n/a

### 3.4.3 Software Interfaces

**ERI.3 Mobile Device OS**

**Source:** Customer

**Priority:** 1

**Introduction:** The operating system of the user’s mobile device.

**Inputs:** n/a

**Processing:** n/a

**Outputs:** n/a

**Error Handling:** n/a

### ~~3.4.4 Communication Interfaces~~

**~~ERI.4 Cloud Platform~~**

~~The system shall be hosted on the cloud platform~~

**~~Source:~~** ~~Team~~

**~~Priority:~~** ~~5~~

**~~Introduction:~~** ~~interface with a PCA core hosted on a remote server~~

**~~Inputs:~~** ~~Set of data points, or a reference file and a single tuple~~

**~~Processing:~~** ~~PCA~~

**~~Outputs:~~** ~~reference file or results file~~

**~~Error Handling:~~** ~~n/a~~

## 3.5 Logical Database Requirements

LDR.1 N/A

## 3.6 Other Requirements

### ~~3.4.3 Cloud Computing~~

**~~ERI.5 Cloud hosted PCA core~~**

~~The system shall host the PCA core on the cloud~~

**~~Source:~~** ~~Customer~~

**~~Priority:~~** ~~5~~

**~~Introduction:~~** ~~The PCA core on a cloud server (Stronger CPU than mobile device) to increase the speed of processing~~

**~~Inputs:~~** ~~n/a~~

**~~Processing:~~** ~~n/a~~

**~~Outputs:~~** ~~n/a~~

**~~Error Handling:~~** ~~n/a~~

# 4. Change Management Process

# When it comes to making potential changes, the group will discuss what needs to be changed in SRS document via Slack which will then be voted and approved upon before being implemented officially.

# Appendices

PCA = Principal Component Algorithm