

**Design Manual**



**Student Name: Wen Liu**

**ID Number: C00105088**

**Project: Fingerprint Recognition**

**Supervisor: Nigel Whyte (M.Sc. MIEEE)**

**Date: 10/01/2010**

**Contents**

1. **Introduction ……………………………………………………………… 2**
2. **High Level Design …………………………………………………….. 3**

**2.1** Data structure design **…………………………………………… 3**

2.2 Database design **………………………………………………... 5**

2.3 Architecture design **…………………………………………… 7**

2.4 Interface design **……………………………………………... 9**

2.5 Modular design **……………………………………………...... 13**

1. **Low Level Design ………………………………………………................ 15**

**3.1 Use case diagram …………………………………………........ 15**

3.2 Use cases **……………………………………………............... 16**

3.3 System sequence diagram**…………………………………...... 18**

1. **Application Test Scheme ………………………………………………......... 21**
2. **Conclusion ………………………………………………............…………. 23**
3. **Introduction**

This paper is the design manual of Fingerprint Recognition project – FingerprintSpy. The main goal of this project is to build a fingerprint recognition application based on biometric identification technology, which can help user to match two fingerprint images for verification or identification and output the matching result with comments.

The FingerprintSpy contains five main modules:

* Graphic User Interface
* Fingerprint Enrollment Module
* Fingerprint Matching Module
* Enrolled Fingerprint Database
* Database Management Module

In this paper, those modules will be explained in detail in High level design section. I will also mention some Low level design elements like user cases, domain module etc.

This design manual shows the interaction of modules in the application and ideas of how I am going to develop this project and meet the goals of this project.

1. **High Level Design**

The High Level Design shows how the application is built, indicates main components and modules of the application and their communications and interactions.

***2.1 Data Structure Design***

The data structure design for the application is to hold data for fingerprint image enrollment and image database update. There are two struct: Minutiae, FingerprintImage.

Minutiae struct holds minutiae data. Each minutia will create a Minutiae instance. The following table shows elements of Minutiae struct:

|  |  |  |  |
| --- | --- | --- | --- |
| **Element** | **Data type** | **Sample Input** | **Comments** |
| min\_type | string | “ending” | Type of the minutiae |
| min\_id | int | 1001 | The id of the minutiae |
| X\_coor | int | 201 | The row index of pixel where the minutiae at. |
| Y\_coor | int | 87 | The column index of pixel where the minutiae at. |
| angle | int | 45 | The rotation degree of the minutiae |
| fingerprint\_id | string | “011004\_09\_15” | The name of fingerprint image which the minutiae in. |

( Table 1. Elements of Minutiae data structure)

With this data structure, all requisite data of a minutia can be stored for future use and each instance of Minutiae structure holds data of a minutia.

Similar with Minutiae struct, FingerprintImage structure holds essential data of a processed, minutiae extracted and ready-to-enroll fingerprint image. There is only one instance of FingerprintImage structure for a opened fingerprint image.

The table2 shows the components of FingerprintImage structure:

|  |  |  |  |
| --- | --- | --- | --- |
| **Element** | **Data type** | **Sample Input** | **Comments** |
| Image\_id | string | “F001\_08\_01” | The name of original fingerprint image |
| fingerprint\_id | string | “011004\_09\_15” | The id of the processed image, same with “fingerprint\_id” element in minutiae struct |
| No\_of\_min | int | 34 | The number of minutiae in this image. |
| comments | string | “middle quality” | The comments about processed image. |
| angle | int | 45 | The rotation degree of the image |

( Table 2. Elements of FingerprintImage data structure)

***2.2 Database Design***

The database separated into two parts in this project: **image database** and **main database**. These databases connected with each other by a key – the name of processed fingerprint image.

Image database, very simple and straightforward, is a folder which holds the processed fingerprint image. See figure 1 for an example.

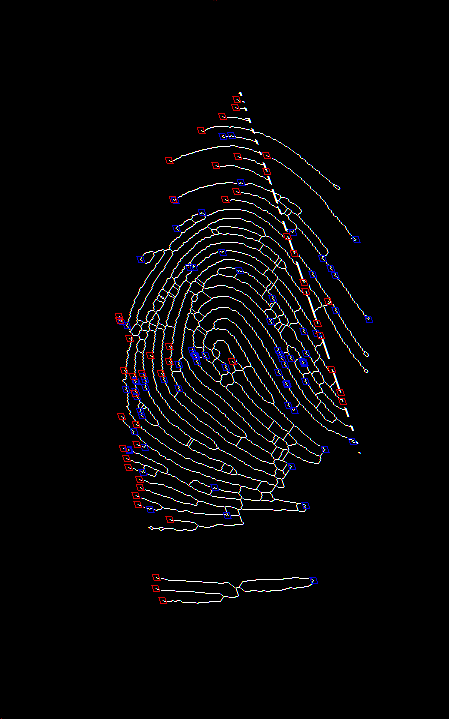
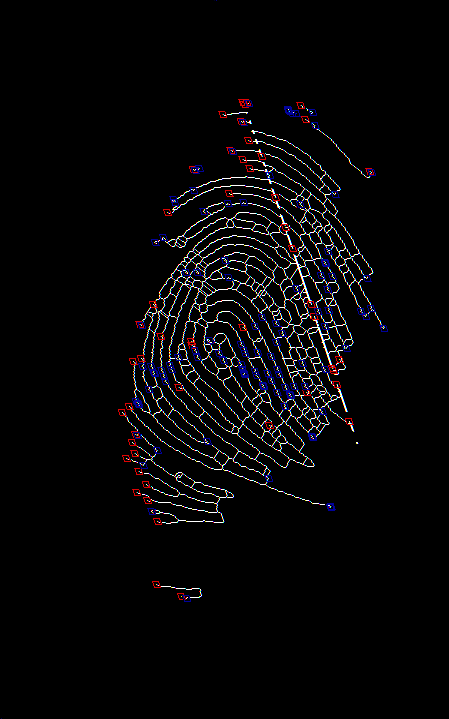


Figure Two processed fingerprint images (expect effect) in the image database of the application

Every image in the database has a unique filename as an identifier. The filename is also the connection key between image database and main database.

The main database contains two tables connected with each other by a primary key value. The data these two tables hold are inputted from those two data structures: FingerprintImage struct and Minutiae struct. Every element of those tables holds the value of corresponding element in those two data structures. The following figure shows the structure of main database.

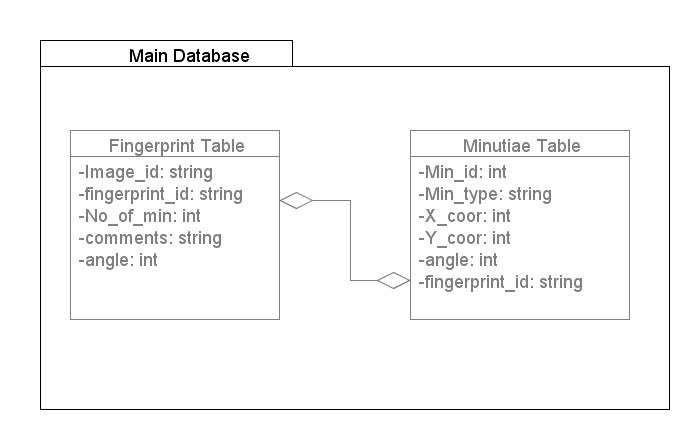


Figure Main database structure

With the database, all essential information of fingerprint image and processed image can be recorded. Matching module can also read these information from database for recognition.

***2.3 Architecture Design***



**GUI**

****

User Interface

File

Operation

Fingerprints

Matching

**Domain**

Database

Management

Image

Preprocessing

Image

Enrollment

Database

Hard Disk

Figure 3 Architecture diagram of the project

**Service**

****

Figure 3 shows the overview of application architecture. The functionality of each component is:

* User interface:

It is the interface between user and other function modules. It takes user’s actions and delivers those actions to corresponding event handlers. It also shows the visible results of calculation and event to user directly.

* Image preprocessing:

The responsibility of this component is to process input fingerprint image with several techniques and algorithms, such like: Soble edge detection, Zhang-Suen thinning algorithm, minutiae extraction and elimination techniques, etc.

* Image enrollment:

It writes processed data of image into main database and saves the corresponding processed image file into image database.

* Fingerprint matching:

In matching section, it allows user to open two fingerprint images from database and running image matching technique on those two images with essential data got from database. It also delivers the matching result to user interface.

* File Operation:

It handles all image file writing and reading operations and other file operation which cause actions of second storage (hard disk).

* Database management:

It handles all database action events. For example: modification, deletion, upgrade, etc.

* Database:

As I mentioned in last section, database of this application is composed of two parts: image database and main database.

***2.4 User Interface Design***

As I mentioned in previous documents, one of the goals of this project is easy- to-use. To achieve this goal, a well designed, straightforward graphic user interface is necessary.

I tried to use several existent fingerprint recognition applications. Most of them are commercial application and contains high accuracy of matching result. However, there are many buttons and options in those applications that make manipulation very difficult, or the final product / result is given without showing steps of how to get this.

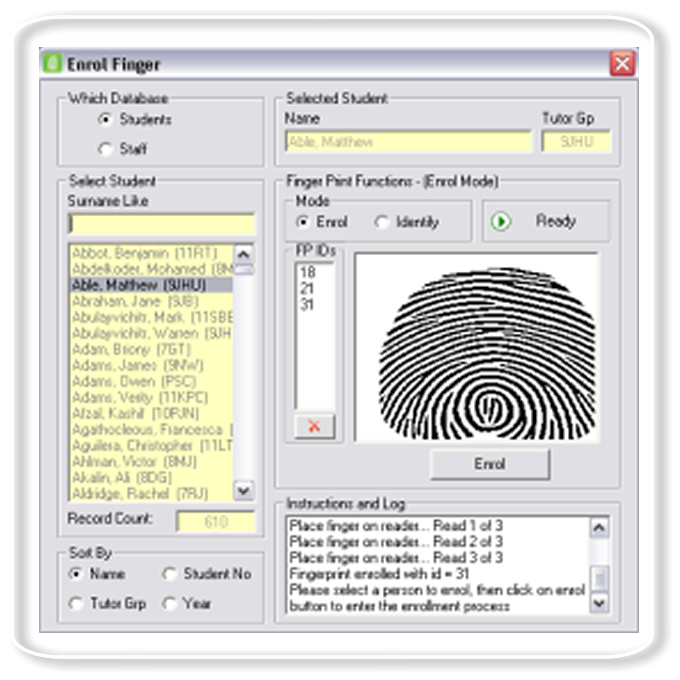


Figure 4 The screenshot of a commercial fingerprint application –TouchRegistration.

In my own project, I finally decided to build up a GUI without redundant buttons, lists and components. The GUI of FingerprintSpy is focus on clear, straightforward, easy-to-use and every step of image preprocessing will be shown in image to show the changes after each function.

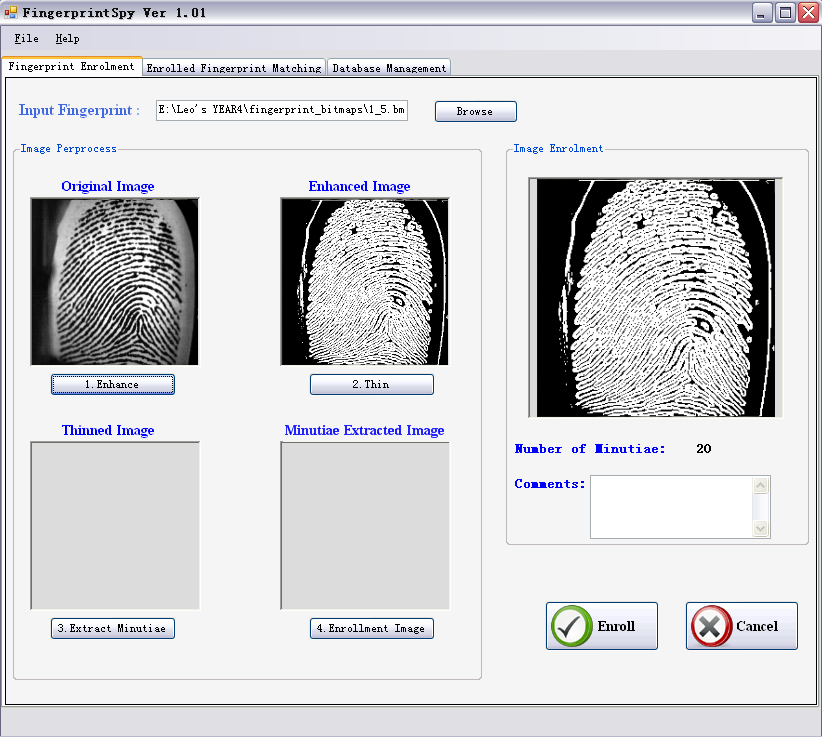


Figure 5 The screenshot of fingerprint enrollment section of FingerprintSpy

The figure 5 shows the GUI of a main functions section of my project – fingerprint enrollment. User click  button to choose an ordinary fingerprint image. The image will be shown in the “original image” box. After clicking those four buttons with function name and number in sequence, those product of each process will be shown in corresponding image boxes. The final product after all image preprocess steps will be shown in a bigger image box on the right. This image and its relevant data will be saved into image database and main database after user click  button.

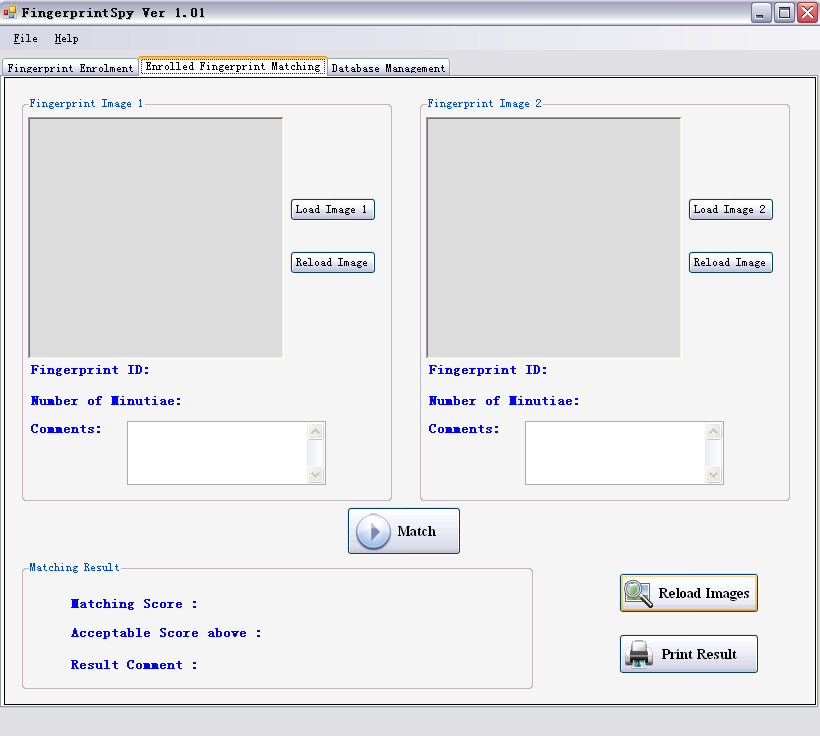


Figure 6 The screenshot of enrolled fingerprint matching section of FingerprintSpy

Same with “Fingerprint enrollment” section, user only needs to select two enrolled image from image database by click “Load Image 1/2” button, the images and their relevant data will be shown in two separate groups.

After selection, user click  button to run fingerprint recognition algorithm on those two images. The result of matching will be shown in “Matching Result” group area in detail with comments.

Two additional functions: “Reload Images” and “Print Result” are also available.

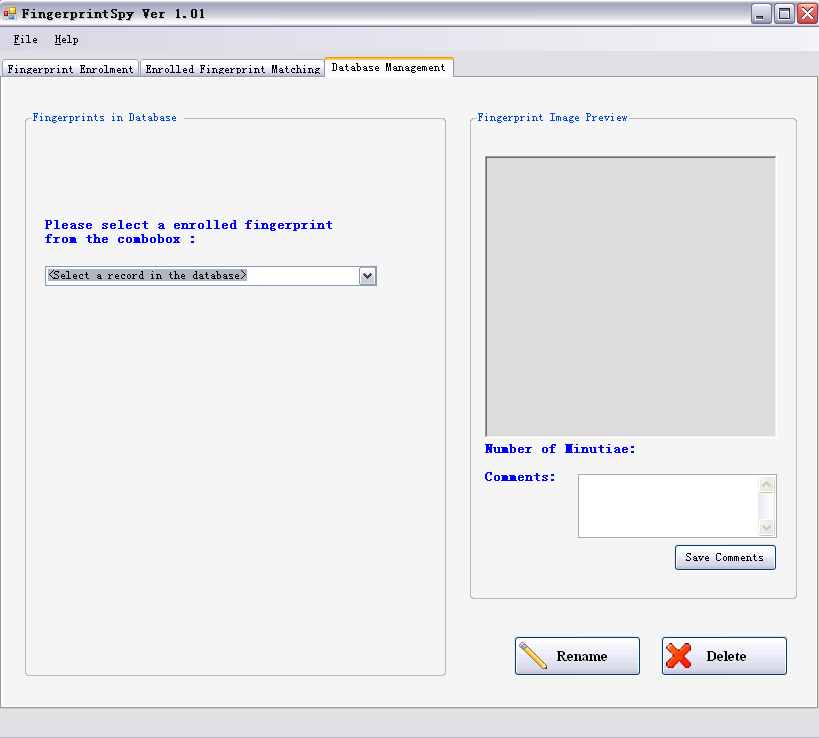
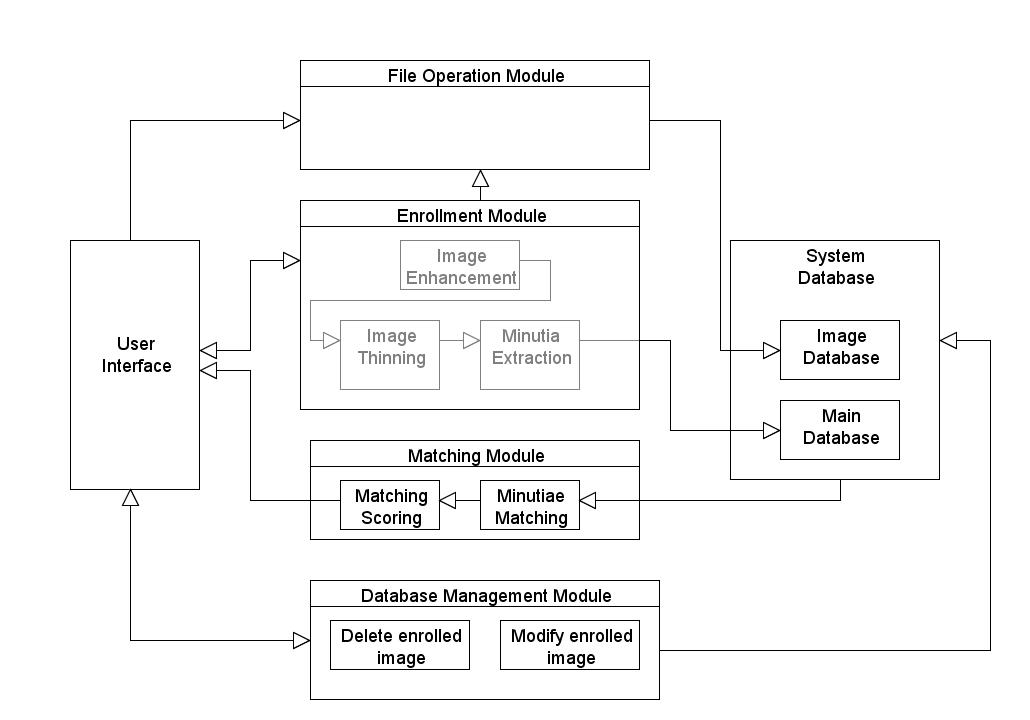
******

Figure 7 The screenshot of database management section of FingerprintSpy

The last but not least section is Database Management. The layout of GUI of this section is also easy to manipulation. The names of all enrolled fingerprint images are in the list on the left. User selects one he/she wants to modify and the enrolled image and its relevant information will be showed in the “Fingerprint Image Preview”. User clicks  button to rename the record and image or clicks  button to remove the enrolled image and its data record from image database and main database.

***2.5 Modular Design***

* User interface:

Figure 8 Modular design diagram of FingerprintSpy system

The User interface can access every module and modules work behind user interface. It is an interface of communications between user and other functional modules.

* Image enhancement:

It is the first component of enrollment module. It enhances the ordinary image and runs edge detection function on the image. User cannot move on to the next step without image enhancement.

* Image thinning:

It is the second component of enrollment module. The enhanced image will be thinned and all ridges will be coming 1 pixel breadth. It is the precondition of minutiae extraction.

* Minutiae extraction:

It is the last component of enrollment module. It extracts all minutiae from the thinned image and eliminated all fake minutiae to improve the accuracy of matching.

* File operation module:

File operation module connect with image database. The functions within the module handle all image file writing and reading operations and other file operation which cause actions of hard disk.

* Main database:

It communicates with minutiae extraction and minutiae matching. It handles literal data of enrolled fingerprint image. It contents functions writing those data into or reading them from main database.

* Image database:

The image database communicates with file operation module. It handles image data of enrolled fingerprint image. It contains functions writing those data into or reading them from image database.

* Minutiae matching:

It contains several complex functions of enrolled fingerprint matching scheme. It connected with system database module and gets image’s data from databases.

* Matching scoring:

The result of minutiae matching will be passed to it. It generates a matching score to show the result of fingerprint recognition. The score is based on the similarity of two images. It gives comments based on the score to increase system’s humanization.

* Modify enrolled image

The responsibility of it is handling enrolled image modification action from users.

* Delete enrolled image

Similar with modify enrolled image, delete enrolled image hands enrolled image modification action from users.

1. **Low Level Design**

The low level design is focus on domain layer design. In this section, I will mention some main terms in low level design view, such as the use case diagram, use cases and system sequence diagram.

***3.1 Use case diagram***

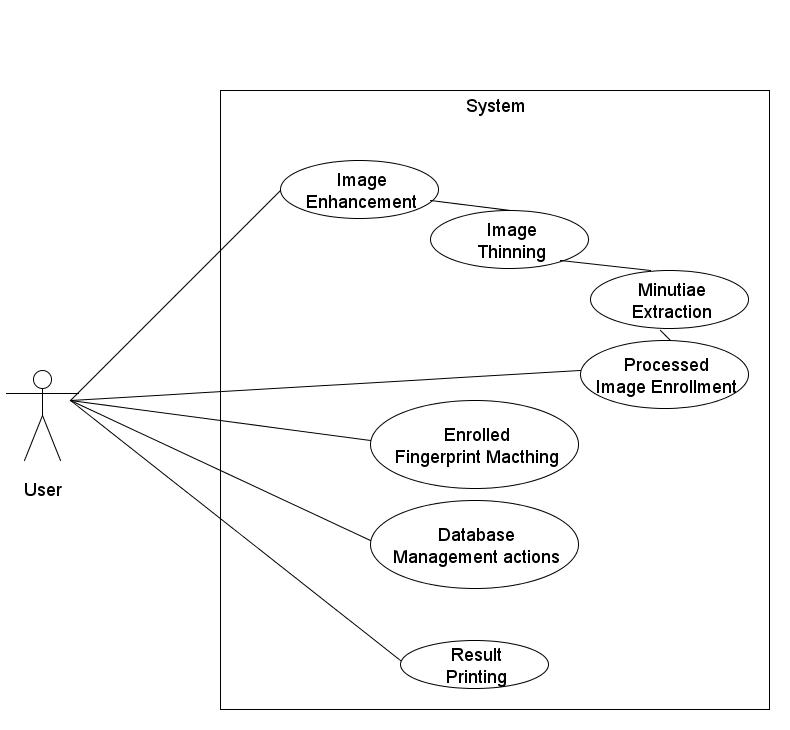


Figure 9 Use case diagram of FingerprintSpy system

***3.2 Use cases***

Use case: Image Enhancement

Actor: User

Description:

User clicks “Browse” button to select an ordinary fingerprint image. The image will be displayed in “Original Image” image box. User then clicks “1. Enhance” button to enhance (run thresholding and edge detection) the image. The processed image will be displayed in “Enhanced Image” image box.

Use case: Image Thinning

Actor: User

Description:

After image enhancement process, user clicks “2. Thin” button to run Zhang-Suen thinning algorithm on the image. The processed image will be displayed in “Thinned Image” image box.

Use case: Minutiae Extraction

Actor: User

Description:

After image thinning process, user clicks “3.Extract minutiae” button to extract all minutiae in the image and eliminate fake minutiae. Every minutia will generate an instance of minutiae struct to hold data of that minutia and an instance of fingerprintimage struct will also be created to hold image data. The processed image will be displayed in “Minutiae Extracted Image” image box.

Use case: Processed Image Enrollment

Actor: User

Description:

After all image processes, user clicks “4.Enrollment Image” button to generate final image which ready to be enrolled. User can input comments about this image and click “Enroll” button and give a name for it to write this image and its relevant data into databases or click “Cancel” button to clear all works did so far.

Use case: Enrolled Fingerprint Matching

Actor: User

Description:

User clicks “Load Image 1” button to select first fingerprint from image database, its relevant detailed information will be loaded from main database and displayed in “Fingerprint Image 1” group area. User iterates this action to load the other enrolled fingerprint.

After select two images, user clicks “Match” button in the middle to start matching scheme. The score of matching will be displayed and a threshold score value will also be gave to show what an accepted mark is. This information and a comment based on them will be displayed in “Matching Result” group area.

Use case: Database Management Actions

Actor: User

Description:

User clicks and selects a name of enrolled fingerprint image from the drop-down list box on the left first. The preview of the selected fingerprint image and its relevant information will be showed in “Fingerprint Preview” area on the right.

After select image selection, user can click “Rename” button to give a new name to the image. User can also remove this image by click “Delete” button. To change the comment about this image, user just types the new comment in the text box under the image and clicks “save comment” button.

***3.3 System sequence diagram***

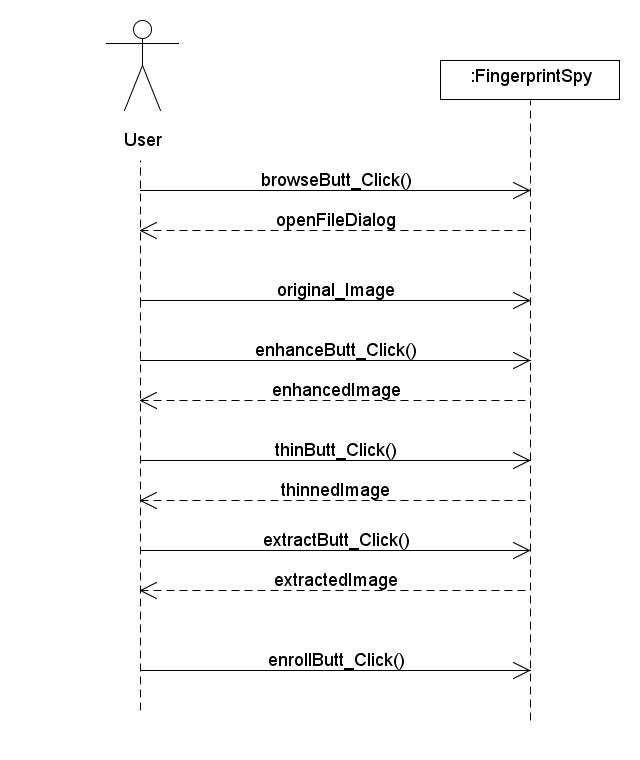


Figure 10 System sequence diagram for image enrollment

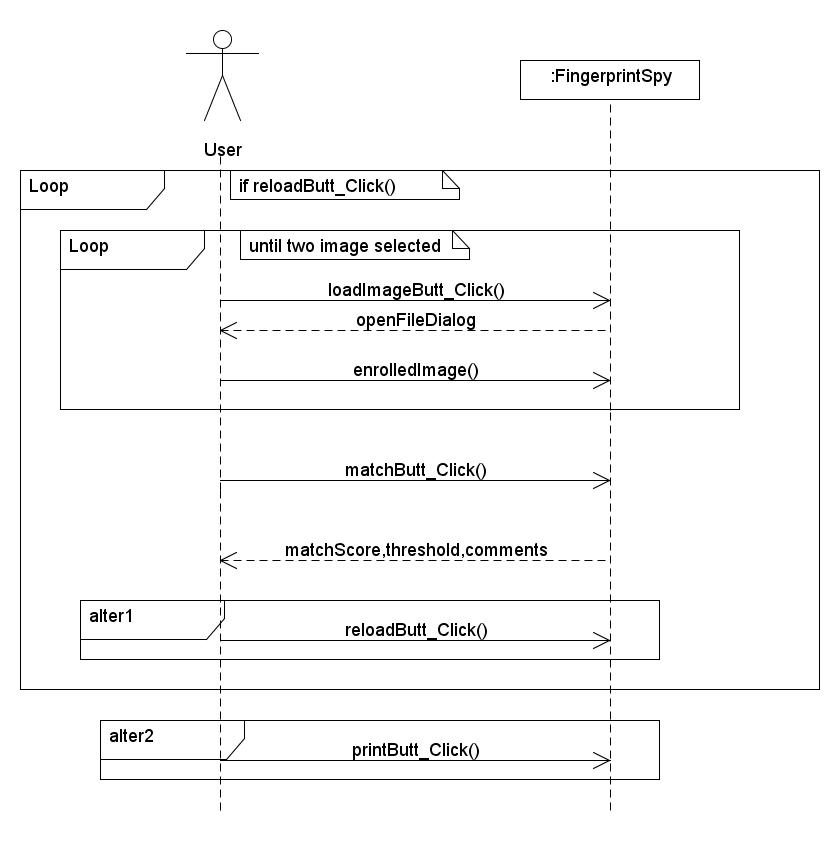


Figure 11 System sequence diagram for fingerprint matching

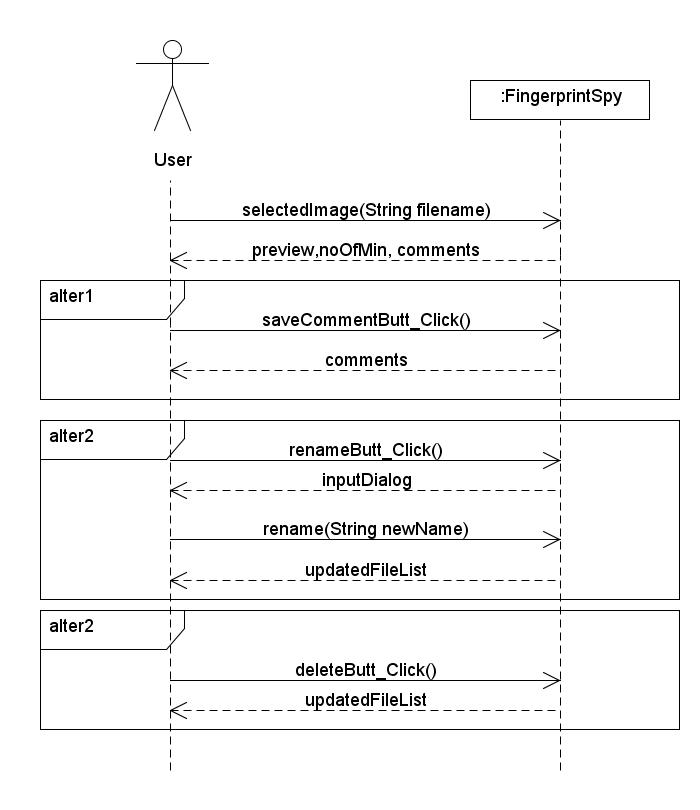


Figure 12 System sequence diagram for database management

1. **Application Testing Scheme**

Testing is an important process of project development. The purpose of testing is to make sure every function modules are working properly as expected, such as:

* If the GUI is connected with other modules correctly.
* If all active components (buttons, menu items, lists, text output areas, etc.) of GUI are working well.
* If input fingerprint image has been processed (enhanced, thinned, minutiae extracted, etc.) correctly.
* If the processed fingerprint image can be saved into image database and its relevant data can be saved into main database without errors.
* If the enrolled image its relevant data can be loaded and successfully from databases.
* If the fingerprints matching algorithm is working correctly and the score, comments are generated properly.
* If the items in databases can be managed from GUI properly.

I designed a testing scheme to achieve these tasks:

***GUI testing***

I will click all buttons and other active components to test their actions and activations (click a button to enable or disable another active component). Check output display components by using different materials.

***Image processing testing***

As I mentioned before, there are three image processes work on an input fingerprint image. I will test these processes separately.

In the enhancement test, I will use some different pictures in different colors and grayscales to check the result of enhancement.

In the thinning test, the images I will use are different in complexity. Some image may be a single character such as “H” or “W” and some image may be very complex such like an image of landscape.

For minutiae extraction, I will not use a 256\*256 pixels fingerprint image because it is too big to locate the coordinate of a minutia by hand. What I will use is a tiny (13\*13 pixels, for example) enhanced and thinned image which content is a single character (such as “H” or “W”). The minutiae coordinate of this image can be easily located by hand and I will compare the result I got with result generated by my algorithm to check the correctness. (See figure 13)



Figure 13 An example of pattern to test minutiae extraction algorithm. \* in red color are minutiae of this pattern. (7\*8 pixels)

The coordinate of minutiae in figure 13 are (0,0), (3,0), (6,0), (0,7),(3,7),(6,7).

***Image enrollment testing***

I will enroll some different processed images to check the image enrollment process and check the correctness of items in main database and their connection.

***Fingerprint matching testing***

I will enroll some different copies of a fingerprint with different qualities and rotations and running several matching between every two of these images to check the accuracy of matching result and its comments.

***Database management testing***

The connection between GUI and database will be tested in previous tests. Database management testing is focus on if the database can be upgraded without errors. I will do some unusual operation about database (such as rename the enrolled image and then click “Delete” button and so on) to check the correctness to avoid system crash.

1. **Conclusion**

The design manual is an essential document which must be done before coding. With a proper design document, coding will be easier and straightforward and the testing, debugging and maintaining of the application will be easier to perform in the future.

In this paper, I mentioned all standard items which should be contained in a design manual. During the design, I get a clear view of the relations and interaction between main functional modules of my project. The ideas of testing are also got during the design.

During the process of application development, I may change some scheme or design I mentioned in this paper to achieve a better result – the future is always uncertain. If there is any change, I will mention it in the documents coming later.