CS2310 Multimedia Software Engineering Luca Lugini
Professor: Dr. Chang

Fingerprint Authentication for SIS-based Healthcare Systems

Project Report

Introduction

In many applications there is need for access control on certain sensitive data. This is especially true when it comes to health care environment, given the nature of medical data in which information is very sensitive. In this project for simplicity I will consider a hospital to be the health care facility. Access control mechanisms need to be in place so that each of the hospital's employees access patient information only relevant for performing their task: for example an employee working in the billing department only needs to know what treatments a patient receives in order to charge the insurance company, but he/she does not need to know whether the treatment was successful or not, or the medical history of the patient. Conversely, a doctor working in the hospital needs access to all of the patient's medical history, but not to the billing history of the patient.

The authentication mechanisms currently used in hospital are:

- knowledge-based: typically consists of passwords;
- token-based: consist of a unique token held by each user, typically an ID card.

These types of authentication have been used in the industry for a long time, though they have major disadvantages: password can be forgotten or stolen, ID cards can be lost or stolen. Recently biometric-based authentication has been gaining popularity because biometric sensors have become relatively small, cheap, and easy to deploy. Biometric authentication is based on an inherent characteristic that each person has, and if the biometric trait (e.g. fingerprint, iris, etc.) is unique enough this can be deployed successfully.

In this project I will explore the implementation of a fingerprint-based authentication component for healthcare system.

Proposed System

The following figure is a graphical illustration of the problem to solve:

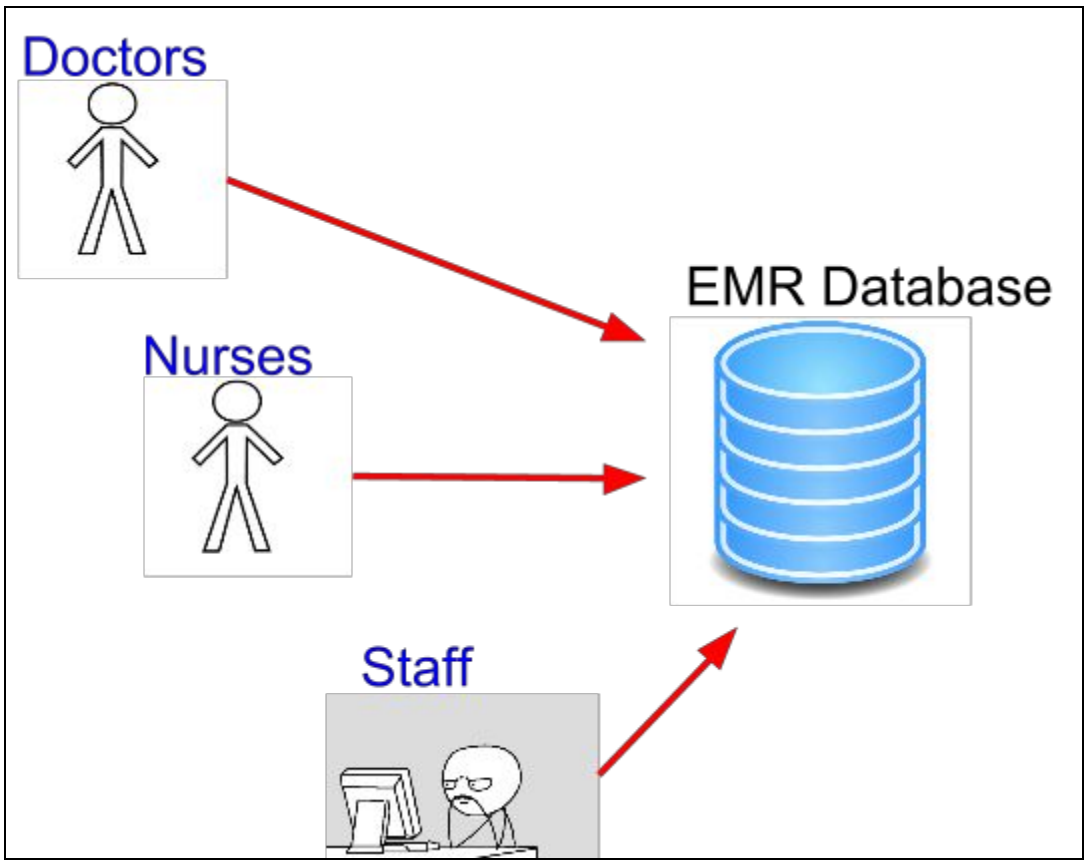


Figure 1: Illustration of access control problem

This project consists of implementing a fingerprint recognition system using software engineering design principles and the SIS software.

The fingerprint system will operate in identification mode: when a fingerprint is submitted as query, the system will check if the user is present in the current user list. It will not return the identity of the person who submitted the query. In other words, it will do a 1:many matching, not a 1:1. This is very useful for access control. The following is a simple scenario for which this fingerprint system would be useful.

Suppose only doctors are allowed to access to patients' medical records:

1. We can first enroll all the doctors in the fingerprint system (they will need to provide a first set of prints, as usual for all biometric systems);
2. Whenever anyone tries to access any medical record we ask for their fingerprint.
3. The fingerprint identification system will verify if the person is in the list of doctors enrolled (if he/she is a doctor in this case).
4. If the fingerprint system recognizes the person then the medical record system will release the patient's record, otherwise it will reject the request.

Figure 2 shows the architecture that could be used in a real-world healthcare system. Only some of the components are displayed, the complete system could have many more components.

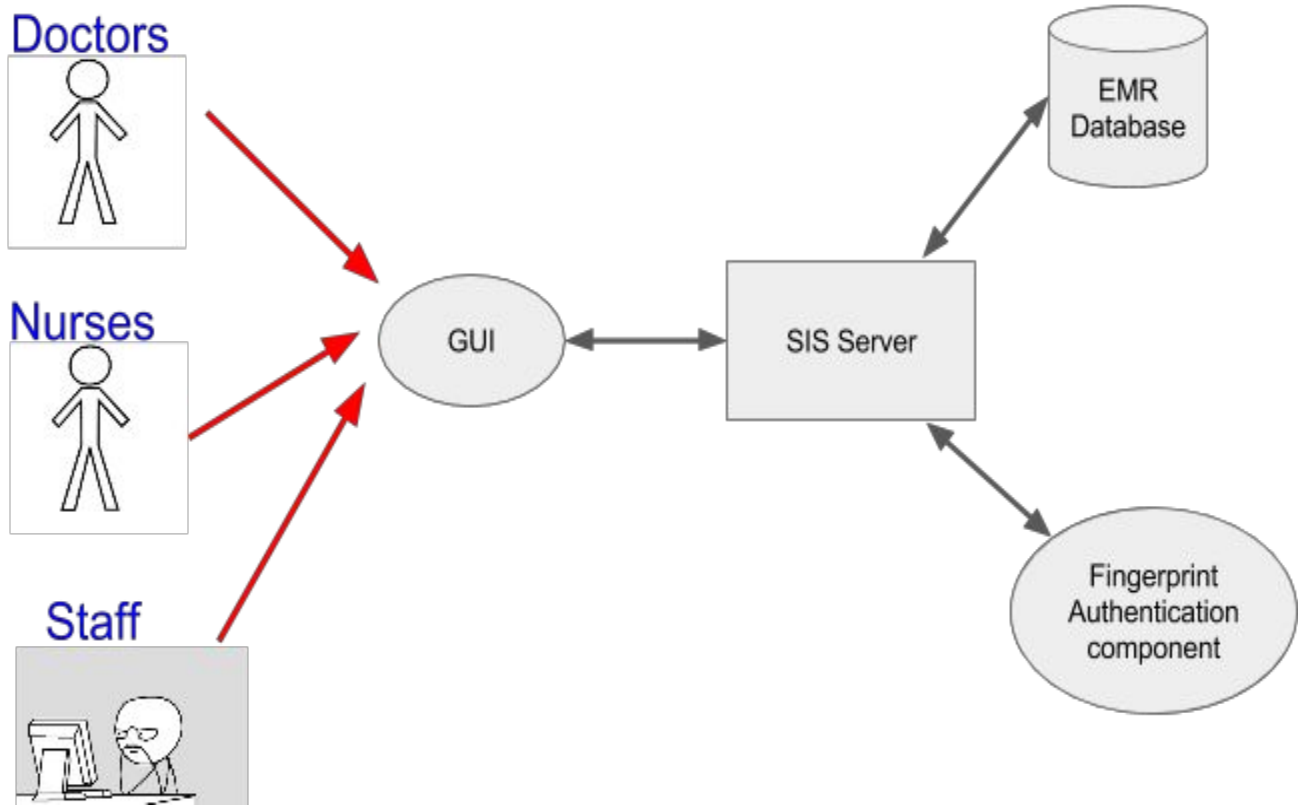
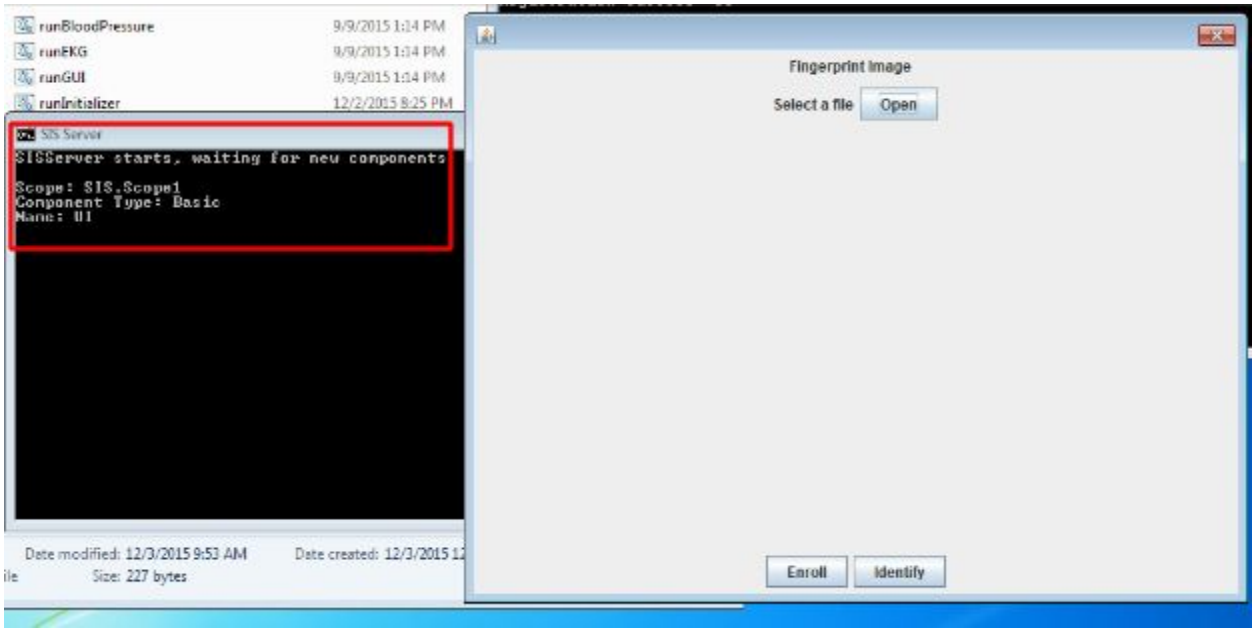


Figure 2: Healthcare system architecture

Typically in a real-world system all the users interact with the system through a Graphical User Interface. The GUI then is connected to the SIS server, and the SIS server is connected to the Fingerprint authentication component, database component, and all the other components. In this scenario if a user wants to access some data from the medical database, the GUI will ask for the user's fingerprint and send it to the SIS server. The SIS server then would forward the fingerprint to the fingerprint authentication component for matching, and depending on the outcome of the fingerprint match the SIS server will allow access to the user and retrieve the data from the database, or not allow access and report an error to the GUI.
For simplicity, in this project I will focus on the Fingerprint Authentication component (called UI in the code, for user interface) and the SIS server.
Since I do not have a fingerprint reader, I will simulate the input of fingerprints through a GUI I created for the fingerprint authentication module.
The fingerprint authentication component was completely implemented by me and it works as follows:

- at startup it registers with the server and then the GUI is brought up:

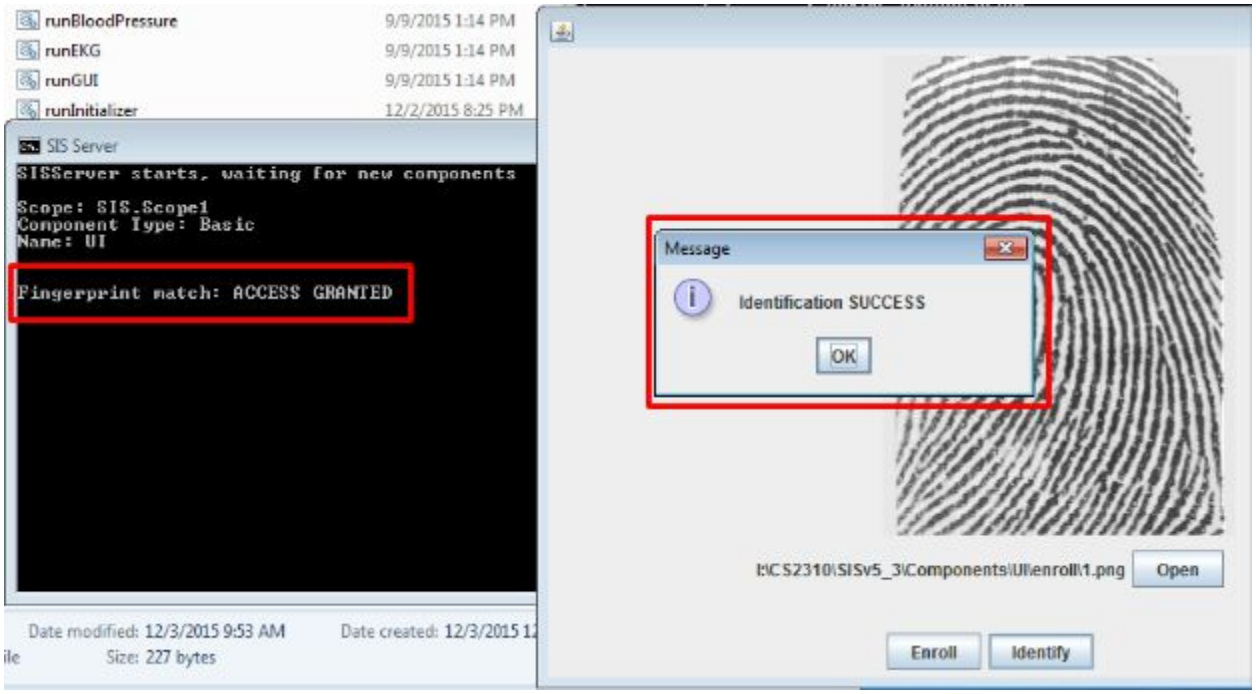


- To simulate a fingerprint scan we can clock the “Open” button and select a fingerprint file:

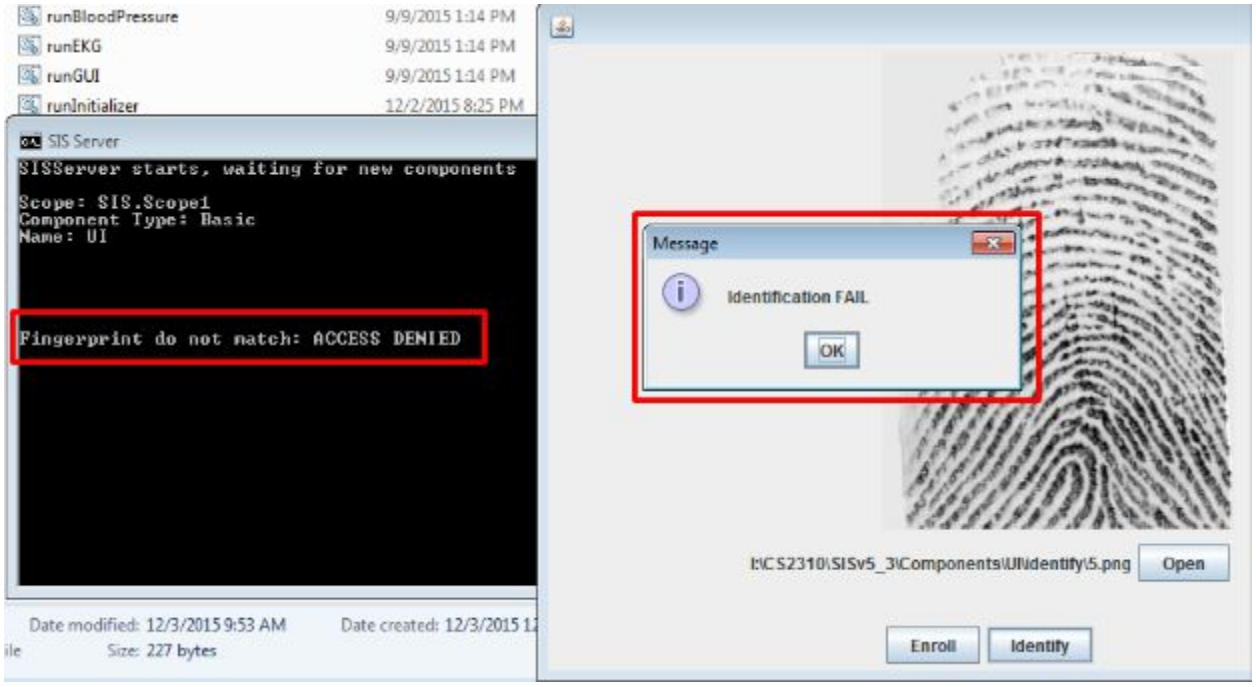
- The fingerprint will be displayed by the GUI and at this point we can clock on the “Enroll” button to register the user:

- We get confirmation that the user has been enrolled. This functionality needs to be used for all the users we want to register to the system so that they can provide a first set of their fingerprints.
- In an operational scenario, when the fingerprint authentication system is deployed, we can try to authenticate in the following way:
- Open a fingerprint image as before to simulate a fingerprint read from sensor:

- Click on the “Identify” button to try to authenticate the user. Two things can happen at this point:
    - The user is recognized, and the fingerprint module notifies the SIS server:



- The user is not recognized, and the fingerprint module alerts the SIS server of the unauthorized access:



- Based on whether the user is recognized or not, the SIS server can now decide whether to allow access to the user or not.

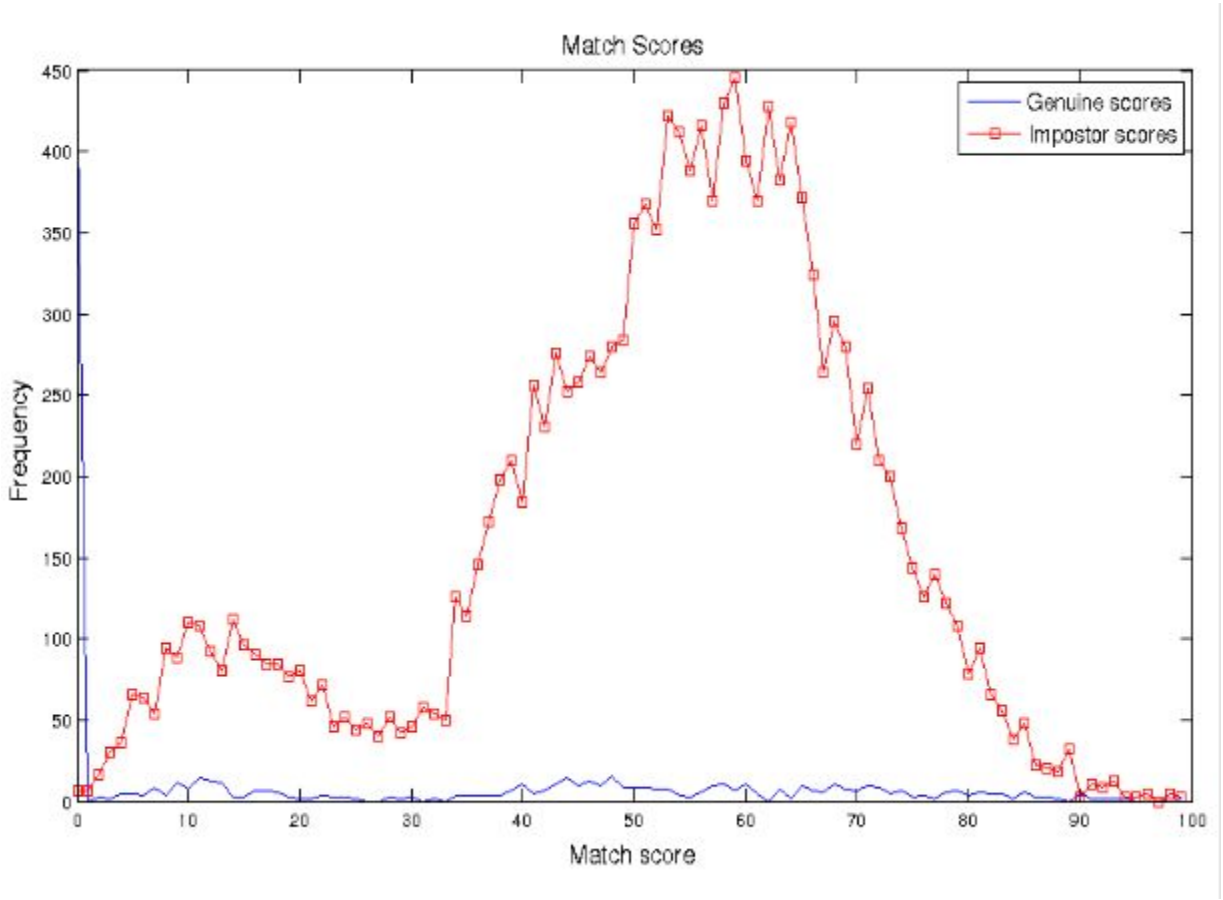
Note, for this project I did not use PrjRemote to simulate messages: the messages are sent directly by the fingerprint authentication component and the SIS server.
Additionally, I also modified the SIS server component to deal with the specific messages received from the fingerprint authentication component. In the base version of the fingerprint authentication component we are only verifying whether or not a particular user is present in the system, without distinguishing between different types of users. This means that every time we click the “Identify” button, the module matches the input fingerprint against all the users currently enrolled in the system. This can easily be extended to recognize user types by constraining the component to match the input fingerprint against only subsets of the currently enrolled users, representing certain categories. Because of time constraints I was not able to incorporate this functionality in this version of the component.

Fingerprint matching algorithm

At the core of the fingerprint authentication component is the fingerprint matching algorithms. Generally, we can divide fingerprint matching algorithms in two categories:

- texture-based: these algorithms take advantage of the textural appearance of fingerprints, which is inherently encoded in the grey-level image information;
- feature-based: these algorithms exploit human-defined features which are extracted from the image; these features are ridges, minutiae (ridge endings and bifurcations), fingerprint core, fingerprint alignment.

For a fingerprint system to be deployed in a real-world system, the matching rates need to be very high, therefore a combination of texture and feature information needs to be used for matching. For this class project, given the limited time, I opted for developing a texture-based fingerprint matcher, which operates on grey-levels of fingerprint images and tries to match the grey-level histograms of two fingerprints.
In order to assess the quality of this fingerprint matcher, I computed all the genuine match scores (scores obtained by matching each fingerprint of a user against all of his/her other fingerprints) and all the impostor match scores (obtained by matching a fingerprint of a user against all fingerprints from all other users). Although not perfect, the algorithm shows acceptable performance. The following is the Detection Error Tradeoff (DET) curve:



This figure shows the distribution of genuine match scores and impostor match scores: since the match score represents the distance between two fingerprints, a low match score indicates a match between two fingerprint, while a high match score indicates that the fingerprint do not belong to the same person. As we can see from the figure there is a separation between the genuine and impostor match score distributions, which is good, but some overlap between the two is still present. This means that the matching algorithm can be further improved. Alternatively, one could implement two version of the algorithm: a standard version to be used

as fast decision cycle and an improved version (which will include advanced image processing techniques and therefore be slower) as slower decision cycle.
The fingerprint I used for tests were obtained from "Biometrics Ideal Test, http://biometrics.idealtest.org/". Because of the license agreement I cannot redistribute the fingerprints, for that reason they are not included in the submission, but one can register to the website and download the available fingerprint datasets.

Conclusions

Through this project I gained some experience in developing components for an SIS-based system. I think the idea and the architecture of the SIS system make it much easier to develop components, taking away much of the communication and other complexity. This way the programmer can only focus on a module at a time which takes care of a specific task. Though the biometrics field is not new to me, I feel like it was a very valuable experience to integrate my knowledge about fingerprint matching in a Software Engineering framework.
A video with a demo of the project is available at the following link:
https://www.youtube.com/watch?v=214X6xQtCZ4&feature=youtu.be

Notes
Possible elements to consider for gems:

- Youtube video with live demo;
- Design of GUI;
- Performance analysis of the fingerprint matching algorithm;

References

[1] Biometrics Ideal Test, http://biometrics.idealtest.org/
[2] A Testbed for Personal HealthCare Slow Intelligence System, https://people.cs.pitt.edu/~chang/163/interface/SequenceSIS.htm