Architecture design for an Artificial Intelligence based medical and healthcare system

Group Number: 65

Group member:

55171121 冯文显

55171122 刘畅

55171123 刘世喆

55171124 赵国璋

55171125 赵岩

Context

1 INTRODUCTION

1.1 Current status and analysis

The application of artificial intelligence in the medical field can be divided into six aspects - virtual assistant, medical record and literature analysis, medical image-assisted diagnosis, diagnosis and treatment result prediction, drug development, and gene sequencing. The software system we designed is mainly for the field of medical image assisted diagnosis.

Medical imaging is the most important source of information for disease screening and diagnosis. The acquisition and interpretation of medical imaging data has become a heavy and challenging task in clinical diagnosis. In some cases, it is difficult to manually interpret images to provide sufficient information for diagnosis, and there are some subjective differences in manual interpretation of images.

Therefore, we are considering the development of a medical image processing system based on artificial intelligence to undertake the cumbersome screening of lesions. It can quickly extract valuable information related to diagnosis from massive data to assist or replace doctors in disease diagnosis.

1.2 Overview

This article mainly describes the framework of the "artificial intelligence-based medical image processing system" designed by us. We first outline our software system architecture and give the basis and method for our design architecture. Then we describe the functions and implementation of each module in our software architecture, and introduce the data flow and cooperation between modules. In addition, we also use the UML view to describe the entire system architecture in more detail.

1.3 Scope

All personnel involved in system analysis and design must read this document carefully and follow the principles described in this document. Those responsible for software system integration and configuration management also need to read this document for system integration configuration environment and system integration settings.

2 ARCHITECTURAL GOALS AND CONSTRAINTS

2.1 Goals

I. Reliability： The system should have high accuracy and stability, while ensuring stable storage for large amounts of image data.
II. Security: The system should be designed to ensure the accuracy and safety of patient information.

III. Specialty: The design of the system should comply with national standards and requirements.
IV. Maintainability: Provide system maintenance personnel with convenient data backup and daily security management methods.

2.2 Constraints

I. The system must ensure that unauthorized data access is blocked
II. Timing evaluates the system image processing results and adjusts and feedbacks the system.
III. The transmission of information between different modules of the system must be accurate.

3 System Architecture

3.1 Architecture Overview

The software is positioned based on an artificial intelligence medical image processing system, which functions to process medical images and use artificial intelligence to identify and detect medical images, thereby diagnosing the patient's condition.

We briefly analyzed the use case. The software system has the following main responsibilities:

1) Obtain patient medical images collected by medical image acquisition machines (CT, BT, CR, etc.)
2) Processing medical images (edge detection cropping, region growth segmentation, image sharpening, etc.)
3) Recognition and detection of images by neural network
4) Analyze the recognition results and give a diagnosis of the condition
5) Store patient medical images and diagnostic information and remote access via the network

Based on the results of the software functional analysis, we present our software architecture. In our design, we divide the medical image processing system based on artificial intelligence into subsystems such as user interface, image acquisition processing system, controller, database, network communication system and auxiliary diagnosis system.

The image acquisition processing system further includes a medical image acquisition module, a medical image processing module, and a neural network detection and diagnosis module. The system architecture design is shown in Figure 1.

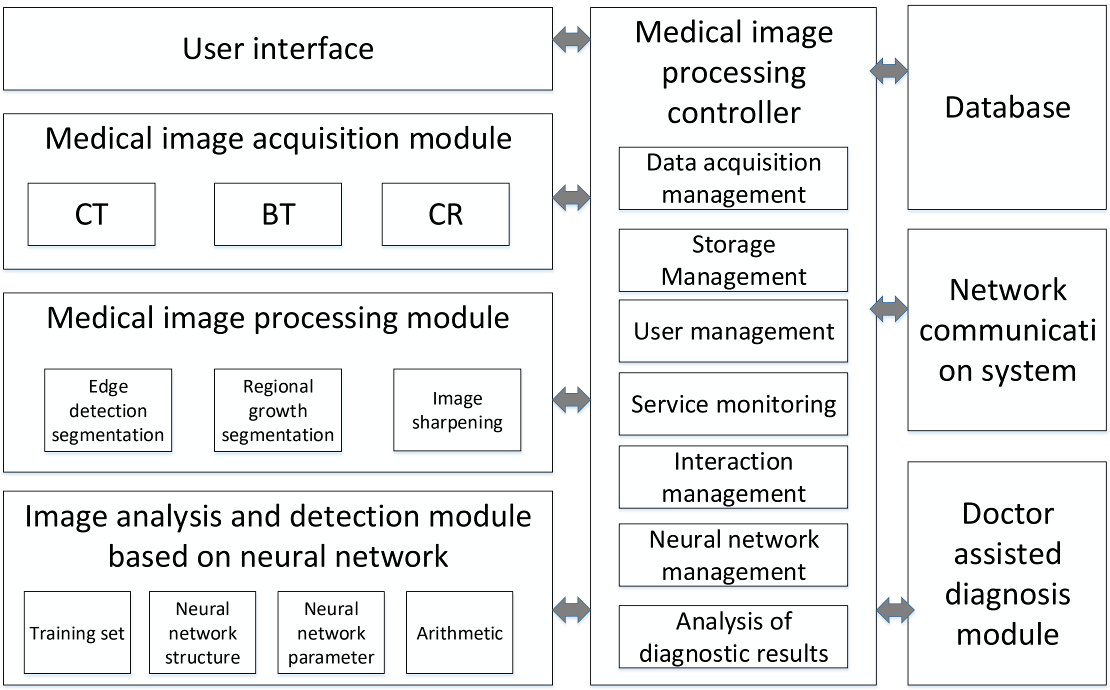

Fig1: System architecture diagram

The architectural pattern we adopt is layered. The functions of main module are as follows:

The medical image acquisition module is responsible for acquiring the medical image of the patient collected by the medical image acquisition machine (CT, BT, CR, etc.) and transmitting the image to the medical image preprocessing module.

The medical image preprocessing module is responsible for processing medical images (edge detection cropping, region growing segmentation, image sharpening, etc.). The processed result will be transmitted to the neural network image analysis detection module.

The neural network image analysis detection module is responsible for identifying and detecting the processed medical image, and sending the result to the doctor auxiliary diagnosis module.

The doctor-assisted diagnosis module comprehensively analyzes the medical image information processed by the neural network to give a diagnosis of the condition.

The control manager is responsible for scheduling the operation of each module, and the user controls the operation of each module through the controller. The main functions of the controller include (user management, data acquisition management, data storage management, service monitoring, interaction management, neural network management, network management, diagnostic result analysis management).

3.2 System main workflow

Firstly, the image acquisition module of the system acquires the patient medical image through a plurality of external medical devices, and then the image processing module performs edge detection segmentation, region growth segmentation, and the like on the image, and then the image data is detected by the neural network module, and the detection result is detected by the doctor. The auxiliary diagnostic module performs the diagnosis and produces the results. The doctor operates the system through the graphical interface through the graphical interface, and the diagnostic results are stored in the database.

The medical image processing flow chart is shown in Figure 2.

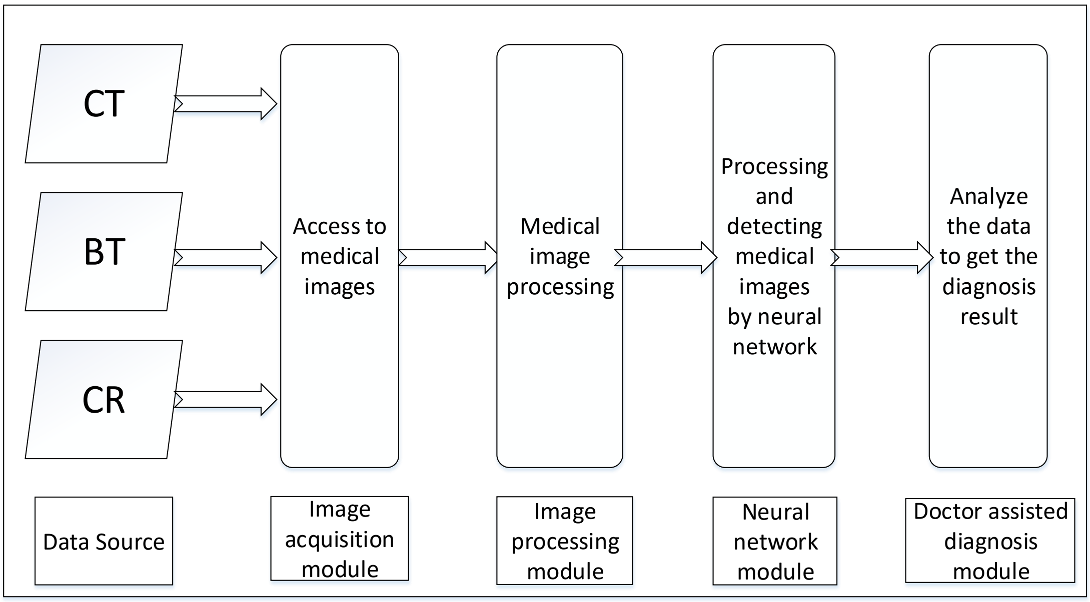

Fig2: Medical image processing flow chart

3.3 How does an artificial intelligence neural network work?

We use artificial neural networks such as convolutional neural networks to identify and classify medical images, and then to detect the degree and type of patients, and to achieve medical intelligence. The structure of the neural network is manually selected, and its node parameters can be obtained by training the labeled medical image training set, and the feedback between the actual result and the predicted result is adjusted. Figure 3 shows how an artificial intelligence neural network works.

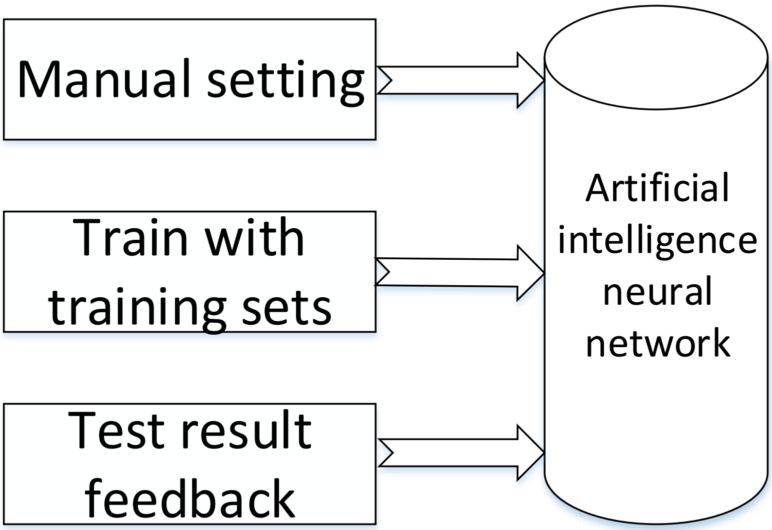

Fig3: Artificial intelligence neural network work diagram

In the following, we will use the UML diagrams to analyze the system architecture in many aspects.

4 USE CASE VIEW

4.1 Requirement analysis

The medical image-assisted diagnosis system is handed over to the hospital to help the doctor identify the patient's medical image information. The system analyzes medical images, determines the patient's disease, and gives the final analysis.

According to the requirement, we analyze the main participants of the system are doctors.

4.2 Actors of the system

Doctor: who be able to use this system to process medical image information.

4.3 Use case diagram

When a doctor first uses a medical image-assisted diagnosis system, registration is required. After successful registration, doctor can login the system to obtain the analysis result of a patient image.

According to the actual requirement, the doctor should be able to view the medical image of a patient, obtain the results of the system processing image information, modify the parameters of the image processing algorithm (neural network algorithm), and store the results of the medical image information after analysis. Figure 4 shows all the use cases of the doctor.

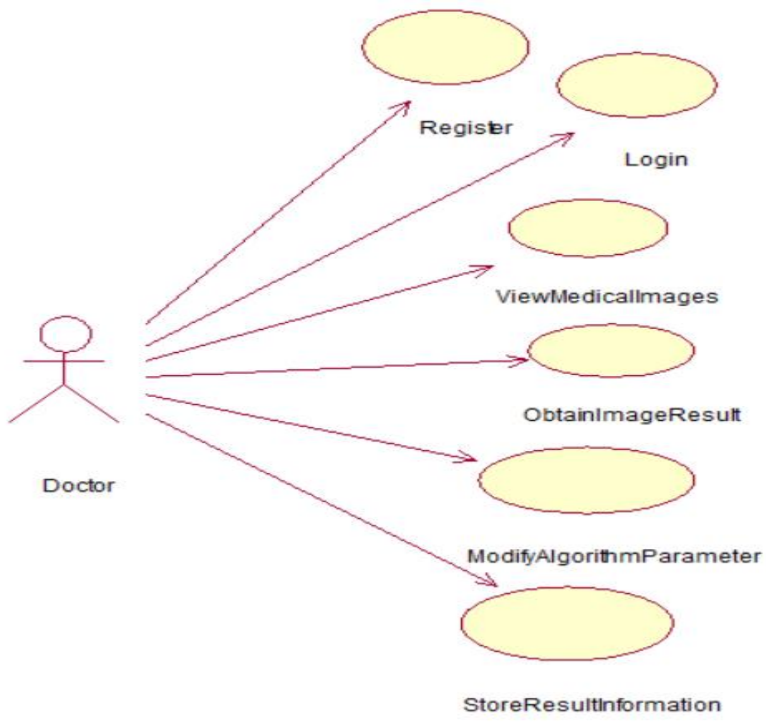

Fig4: Doctor Use Case Diagram

3.4 Use case description

<1>Use Case: Register
Participants: Doctor
Brief description: It describes how a user register to become the System-User
Main flow:
1. User enters the registration screen, input their personal profile.
2. The system validates the input information and return feedback that the user has registered successfully.
Alternate flow: When the personal information is incomplete or exists errors, the system return feedback that the information doesn't meet the norms and the user need to examine his/her information.

<2>Use Case: Login
Participants: Doctor
Brief description: It describes how a doctor to enter the medical image processing system
Main flow:
1.Doctor inputs his or her account information
2.The system validates the input information and jump to the corresponding page
Alternate flow: When the account information doesn't exist, the system prompt error message

<3>Use Case: ViewMedicalImages
Participants: Doctor
Brief description: It describes that a doctor View patient medical image
Main flow:
1.Doctor selects one image and clicks "View" button 2. The system shows the image

<4>Use Case: ObtainImageResult
Participants: Doctor
Brief description: It describes how a doctor to get the image analysis processing result
Main flow:
1.Doctor selects one image and clicks "GetResult" button
2. System begins to analysis and process the image 3. System shows the result to doctor in screen

<5>Use Case: ModifyAlgorithmParameter
Participants: Doctor
Brief description: It describes how a doctor to modify the neural network algorithm parameter
Main flow:
1.Doctor clicks "Modify" button and then input a number to replace modify the algorithm parameter
2.The system changes the neural network algorithm parameter
3.The system shows the message "modify successfully"
Alternate flow: When the information that doctor input is false, the system prompt error message.

<6>Use Case: StoreResultInformation
Participants: Doctor

Brief description: This use case describes how a doctor to store Result Information

Main flow:

1.Doctor selects some results and clicks “Store” button 2.The system stores these results in database 3.The system shows the message “store successfully” in screen

5 LOGIC VIEW AND PROGRESS VIEW

In this part, three sequence diagrams will be showed and its detailed description also be explained. In addition, the class diagram of the entire system and activity diagram will also be shown here.

5.1 Sequence diagram

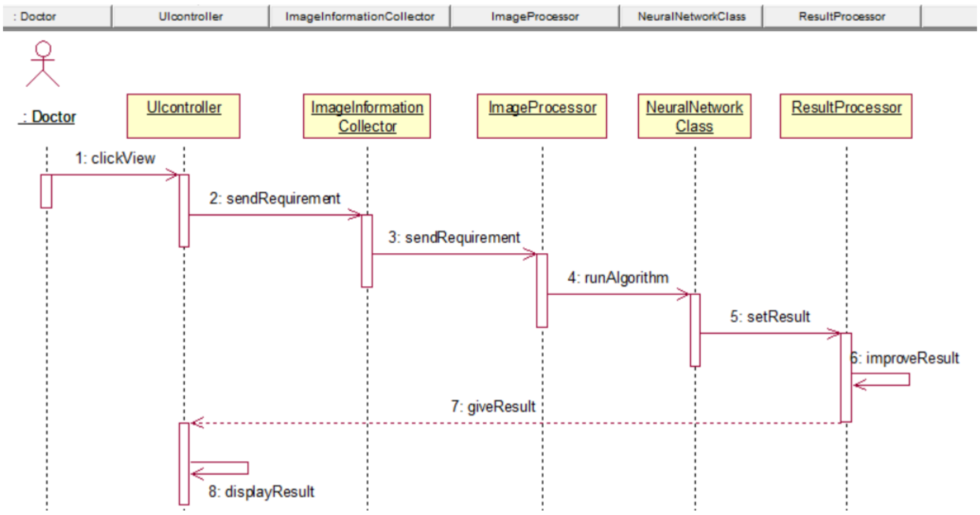

Fig5: “ObtainImageResult” sequence diagram

Explanation: When the doctor clicks the "View" button, the image information collector sends a signal request to the image processor class to process the related image, then the image processor calls the neuron network class, and finally passes the neural network class to the result processor for more Accurate result processing and display the exact results of the processing on the screen.

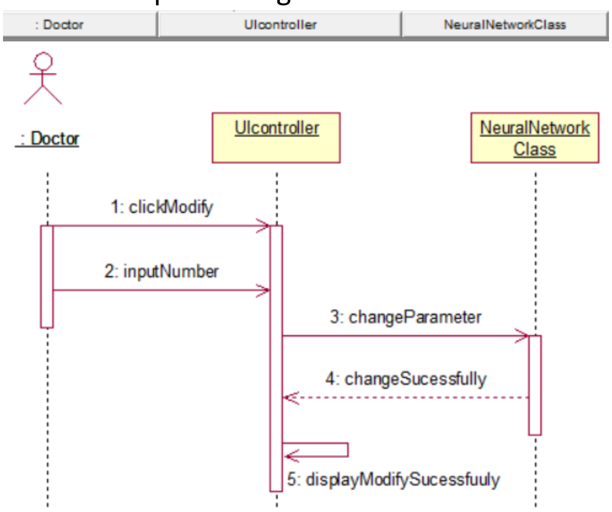

Fig6: “ModifyAlgorithmParameter” sequence diagram

Explanation: The doctor clicks the "Modify" button, and then enters the value of the modified parameter. After the modification is successful, the system interaction interface will inform the doctor.

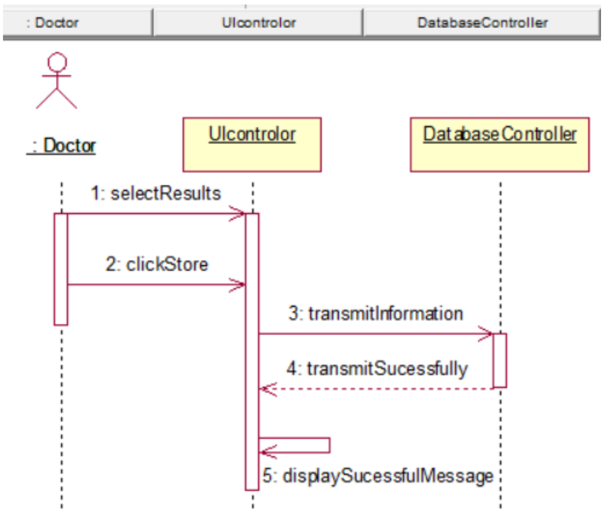

Fig7: “StoreResultInformation” sequence diagram

Explanation: The doctor selects some medical image processing results, click on the "Store" button, and the results will be transferred to the database for storage.

5.2 Class diagram

According to the sequence diagrams above, it is easy to separate the classes of the system.

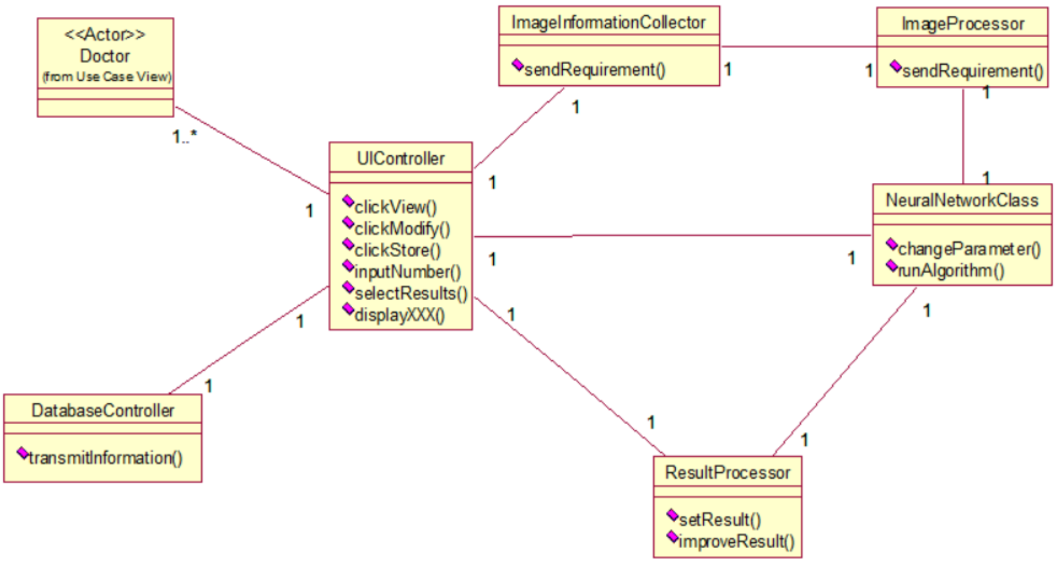

Fig8: Class diagram of the entire system

5.3 Activity diagram

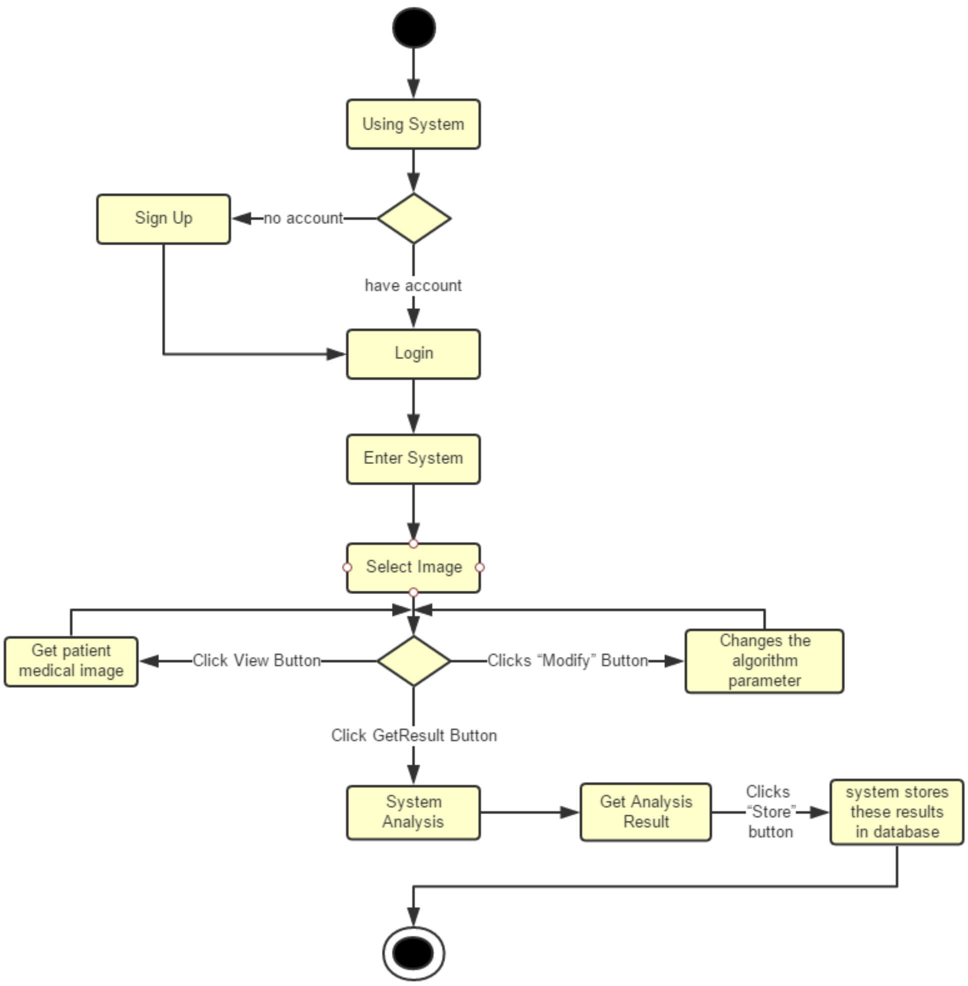

Fig9: Doctor's activity diagram

6 COMPONENT VIEW AND DEPLOYMENT VIEW

6.1 Component diagram

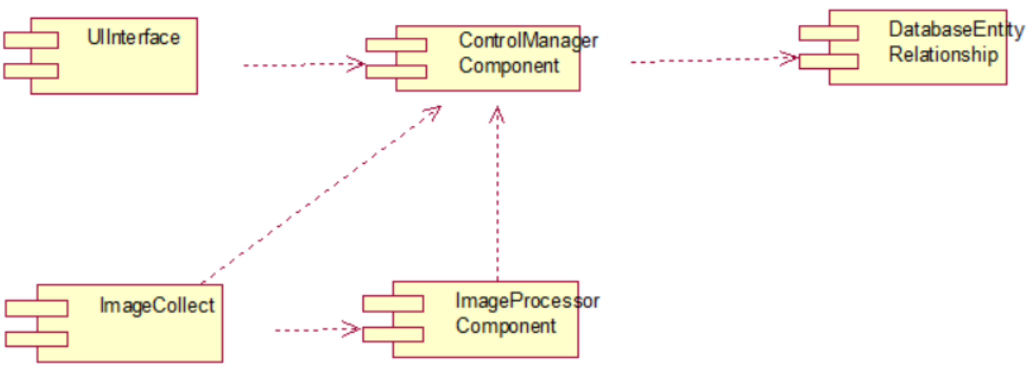

Fig10: Component diagram

The component diagram shows the physical composition and the dependency relationship among those components of the system.

6.2 Deployment diagram

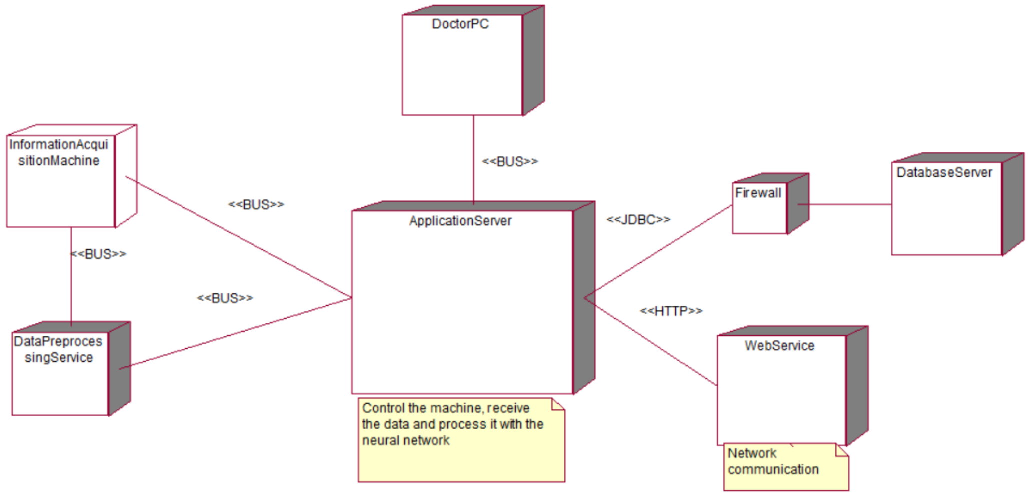

Fig11: Deployment diagram

Deployment diagram of the architecture describes the various physical nodes for the most typical platform configurations. Also, it describes the allocation of tasks (from the Process View) to the physical nodes.

1. DoctorPC: Interacts with ApplicationServer through the bus, shows what users want to do.
2. InformationAcquisitionMachine: Collection of patient's body information, then the data is transmitted to the DataPreprocessingService through the bus.
3. DataPreprocessingService: Receives data from InformationAcquisitionMachine, the data is simply processed by some algorithms and then transmitted to the ApplicationServer through the bus.
4. Firewall: Protects database.
5. DatabaseServer: Receives data from ApplicationServer and stores all the information.

7 CONCLUSIONS

In this article, we designed the appropriate architecture for our software and described it in great detail. The whole architecture included all the essential parts to describe how the system work. And the architectural pattern we adopt is layered. Requirement is analyzed in use-case view. What’s more, several sequence diagrams are provided to explain some important use-cases more exactly. After that, the class diagram has illustrated all the essential classes and their relationships. Then in activity diagrams, probable states of an item and the transformation among them are shown in a proper way. Finally, we give a component diagram to describe the main functions of a system from a software architecture perspective and a deployment diagram to map between software and hardware.

We have gained a lot from this project. We have gone through a process of creating a system coming from nothing. From requirement an analysis to design, we discussed a lot of problems that we had never thought of before. In the process of working hard on our project, everyone in our team tried his best to make a better effort to our project. The important factors that makes us finish the architecture builder successfully are the teacher's guidance and the students' solidarity and cooperation.

Thank you very much!

8 REFERENCES

[1] Hill J, Szewczyk R, Woo A, et al. System architecture directions for networked sensors[C]//ACM SIGOPS operating systems review. ACM, 2000, 34(5): 93-104.
[2] Borthakur D. The hadoop distributed file system: Architecture and design[J]. Hadoop Project Website, 2007, 11(2007): 21.
[3] Engler D R, Kaashoek M F. Exokernel: An operating system architecture for application-level resource management[M]. ACM, 1995.
[4] Luckham D C, Kenney J J, Augustin L M, et al. Specification and analysis of system architecture using Rapide[J]. IEEE Transactions on Software Engineering, 1995, 21(4): 336-354.
[5] Loening A M, Gambhir S S. AMIDE: a free software tool for multimodality medical image analysis[J]. Molecular imaging, 2003, 2(3): 15353500200303133.