



THE CHINESE UNIVERSITY OF HONG KONG, SHENZHEN

CSC4001 PROJECT PROPOSAL

Web-based Medical Bot

Authors:

CHEN Yaqian

LIU Jiazhen

XIAO Shiting

XUE Chunyue

ZHOU Zheyuan

Student IDs:

117010032

117010168

116010243

116010255

117010423

March 28, 2020

Contents

1	Introduction	2
1.1	Motivation & Background	2
1.2	Requirement Analysis	2
2	Proposed Methodology	3
2.1	System Design	3
2.2	Implementation Strategies	5
2.2.1	Development Model	5
2.2.2	Dataset preprocessing	6
2.2.3	ML model training & prediction	6
2.2.4	Website development	6
2.3	Possible Advanced Development	6
2.3.1	ChatBot	6
2.3.2	Test for more types of disease	6
3	Software Testing	8
3.1	Test Planning	8
3.1.1	Test Assumptions	9
3.2	Test Design	9
3.2.1	Development Testing	9
3.2.2	Release testing	9
3.2.3	User testing	10
3.3	Test Execution	10
4	Work Division	10
4.1	Labor Division	10
4.2	Timeline	11
5	References	12

1 Introduction

1.1 Motivation & Background

Since ancient times, epidemic diagnosis and control have been the hot topics for human beings. Among all the epidemics, due to the frequent pandemics in recent years, influenza and pneumonia, which are mainly characterized by respiratory infections, have especially attracted widespread attention. Taking the recent novel coronavirus disease (COVID-19) as an example, up to May 2020, this disease has already caused more than 300,000 coronavirus cases and 13,000 deaths all over the world. This disease is extremely contagious, and once developing into a severe condition, it will pose a huge threat to human life. In the process of its continuous spread, people need to spend a lot of resources to diagnose and treat confirmed or suspected patients, which will cause the medical system to overload and even collapse. It will be difficult to effectively control the spread of the disease if effective means cannot be used to diagnose suspect patients quickly and in large quantities.

Fortunately, as the rapid development of medical image processing and analysis, these techniques offer an attractive potential method for rapid self-diagnosis of epidemics. In previous cases, F. Taher and R. Sammouda [3] develop a deep-learning diagnostic system for lung cancer detection in its early stages based on the analysis of the sputum color images, and M. B. A. Miah and M. A. Yousuf [2] also provide a lung cancer detection method using CT images instead, which can achieve an accuracy of 96.67%.

For COVID-19, besides the results of real-time reverse transcriptase PCR (rRT-PCR) testing, the noticeable features that distinguish it from flu and the common cold are the multiple small patchy shadows and interstitial changes at the early stages, and the multiple ground-glass shadows and infiltrates shadows at the developed stages on CT images of the lungs [1]. More than lung cancer, for these diseases similar to COVID-19, which are very characteristic on CT images of the lungs, medical image processing and analysis are undoubtedly full of the enormous potential for alleviating the above difficulties. This inspires us to build a Web-based Medical Bot, in which the users can upload their medical images and get an initial diagnosis of what they have.

1.2 Requirement Analysis

1. The user must upload his/her digital CT images of the lung in the certain format to the Medical Bot website, and the system shall diagnose and display possible lung diseases and their corresponding probabilities based on the provided images.
2. an advanced function, the user can further enter his/her main symptoms in the webpage ChatBot, then the system shall use the description as additional information to get the diagnosis together with CT images.
3. The system shall further provide additional information about the diseases such as symptoms and treatments based on the diagnosis.
4. The user can also enter the name of one disease in the webpage ChatBot, and the system shall return the information about the diseases including symptoms and treatments.
5. As an advanced function, for the user as a doctor, the system shall allow the user upload multiple patients CT images each with a tag (eg: patients name or id), and return separate reports for each patient.

2 Proposed Methodology

2.1 System Design

The Medical Image Detection System can be mainly divided into eight classes. During the implementation stage, training dataset and testing dataset will be constructed to optimize the system. After the completion of the core module part of the system, the web interface will be built up and moved to the testing and validation stage.

The mature system can satisfy the following user functions: Firstly, users can register an account in the system with all the personal information, which can promote the accuracy of examination, including but not limited to age, gender and etc.. Then, after successfully logging in, the user can upload the medical image and choose the image type, target torso part (i.e. organism) of the image respectively. In the next stage, the medical image detection system will analyze the figure with the corresponding algorithm and calculate the positive probabilities for possible disease. After the detection is finished, the report of the medical image detection result will be loaded and presented to user, user can choose to save, print or send the report according to personal needs. The test result data will be recorded in the profile of the user and the test result will be added into the training database for further optimization.

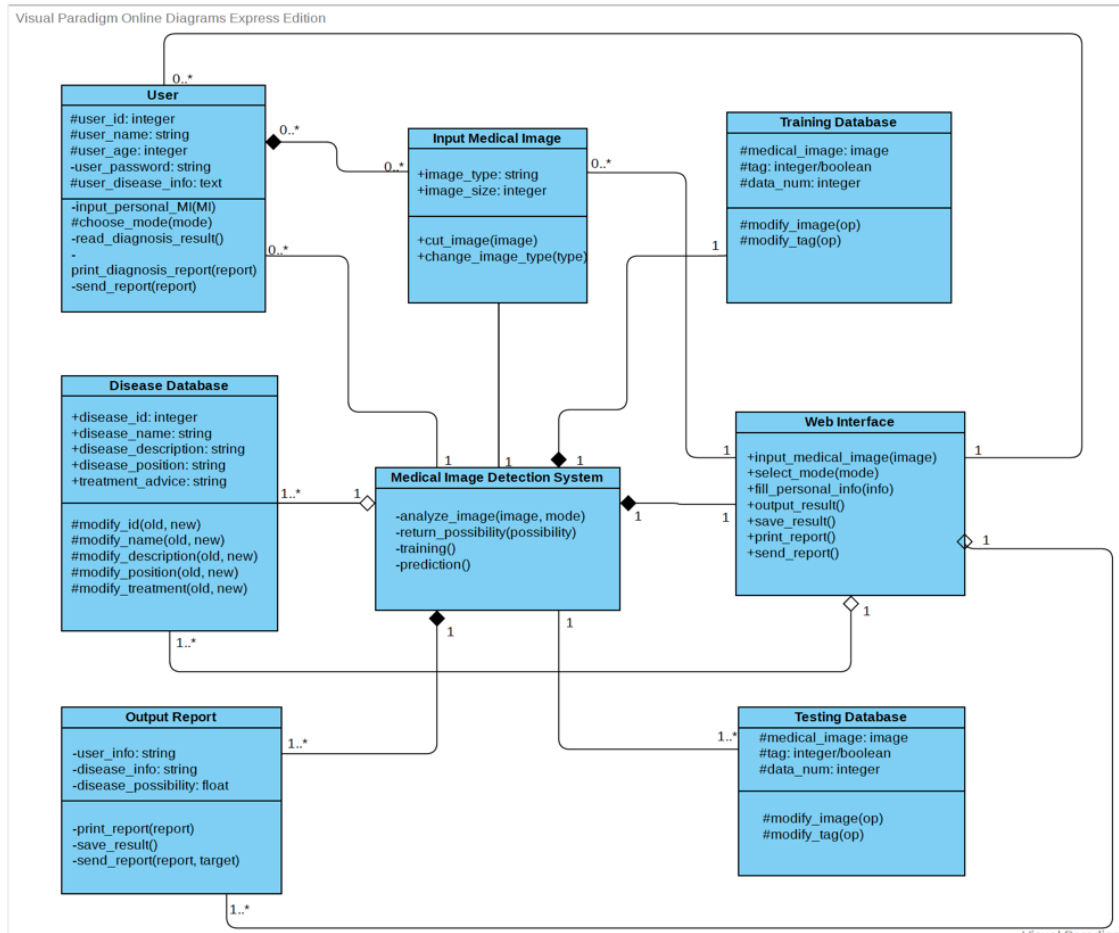


Figure 1: Class Diagram for Medical Bot System Baseline Situation

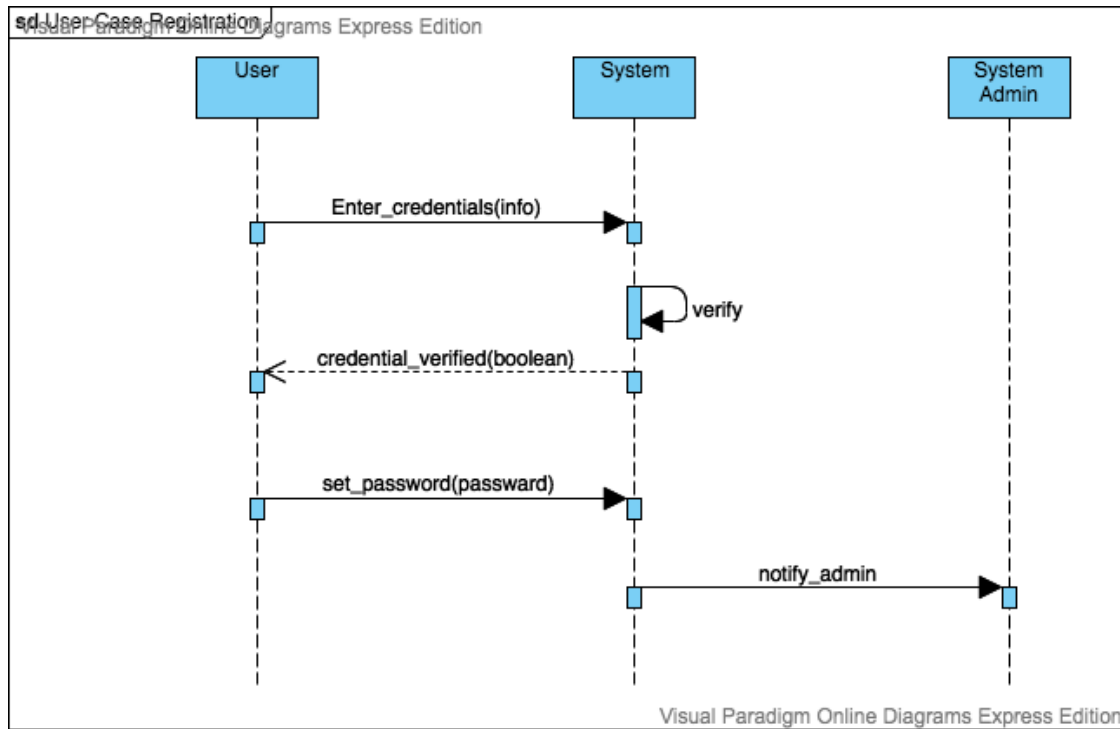


Figure 2: Sequence Diagram for User Case One - Registration

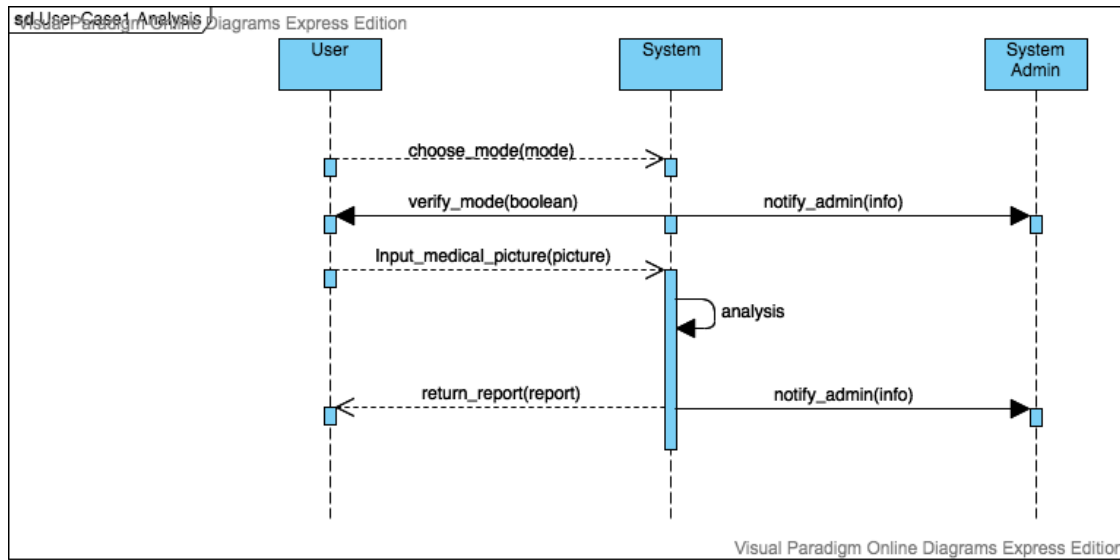


Figure 3: Sequence Diagram for User Case Two - Medical Image Analysis

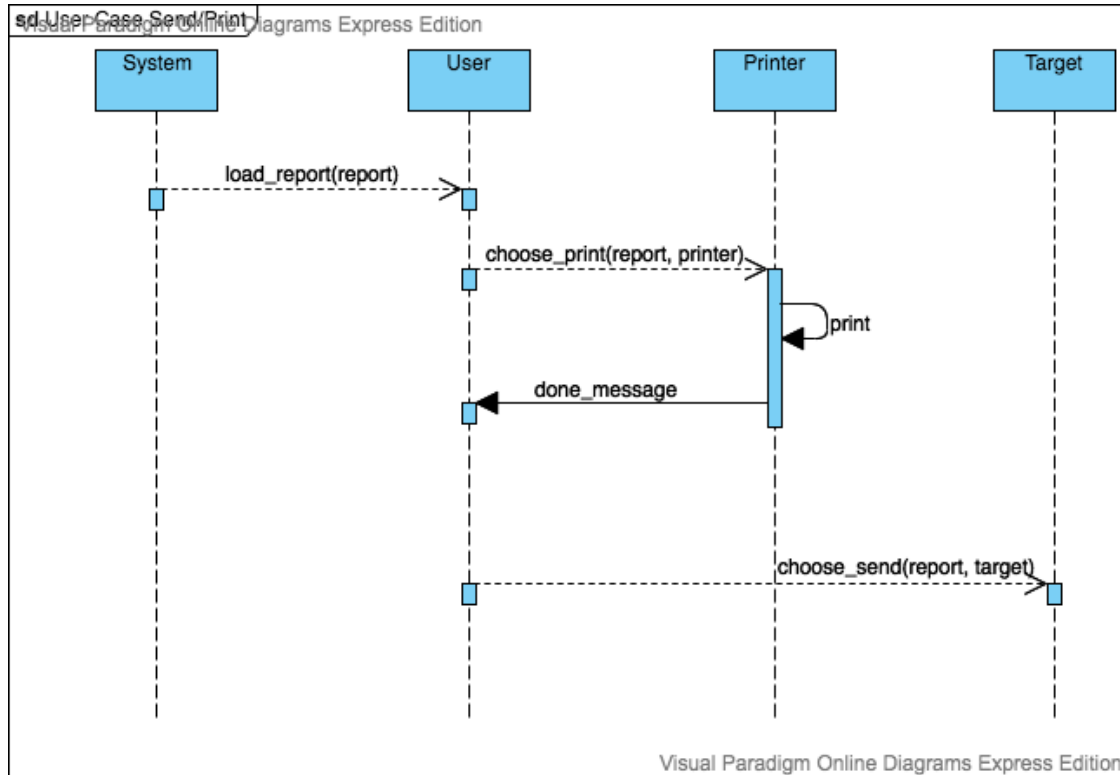


Figure 4: Sequence Diagram for User Case Three - Send/Print

2.2 Implementation Strategies

2.2.1 Development Model

The software process is determined to be combined of two approaches: plan-driven process and agile process.

The plan-driven process will be applied for baseline implementation: Pulmonary Nodules Detection Software, considering the writers have a complete and intact idea of frame, structure and prospective functions for this baseline, which has been described detailedly in the system design part. The waterfall model is used with five stages: requirement definition, system and software design, implementation and unit testing, integration and system testing and operation and maintenance, and each stage is explicitly described in the proposal.

The agile process will mainly be used for further implementation after realizing the baseline, considering optimizing the system would be relatively flexible and planning is incremental, it is easier to change the process and detailed functions with markets inconstant requirements. Integration and configuration model is more suitable for this project: systems integrated from existing components or COTS systems of different CAD (Computer Aid Diagnosis) fields and mature algorithms will be taken as reused elements, which, despite the baseline Pulmonary Nodule Detection system, will include but not limited to Artificial Intelligent ChatBot for disease consultation, Risk assessment of Alzheimer's disease by detecting retina neural, Stroke risk assessment from brain CT/MRI medical images.

2.2.2 Dataset preprocessing

We plan to preprocess the data using Python and its third-part library Numpy. A series of operations would be applied upon the submitted medical image, including the following:

1. Convert the input image, which is supposed to be of type CT, to .raw or .mhd format;
2. Denoising, coordinate transformation, resolution unification, etc;
3. 3D-patching, hard sample mining, data augmentation, etc.

For dataset, we plan to train our model on LUNA, which is a frequently used benchmark in the pneumonia diagnosis community.

2.2.3 ML model training & prediction

We will build our machine learning model using Pytorch in Python, because Pytorch is exactly the appropriate framework for structuring a small-scaled model, and it can fully make use of the speed-up of GPU to accelerate the computation so as to achieve better online interactive results. For pneumonia, the proposed model will mainly contain two parts, the first one of which is responsible for body part detection, while the second part takes as input the outputs of the first part and deals with the classification problem to decide whether the submitted image implies illness.

Based on the well-defined model, we will train and test it using cross-validation strategy, until it converges. This provides a set of optimal parameters which can be directly used for the real-time, online diagnosis.

2.2.4 Website development

The expected product we hope to achieve is a web-based software. For the front end, we plan to use css, HTML, and Javascript to build up the web-page. Node-js will be utilized for the service end. In order to build the backend database of the images, we plan to use MongoDB in Python.

2.3 Possible Advanced Development

2.3.1 ChatBot

An advanced version of our project may additionally contain a ChatBot. Two types of ChatBot may be implemented, with one facing the patients and the other one mainly being functional to the doctors. After acquiring the test results from the model, the patient can obtain some supplementary information by using the ChatBot. For instance, when the patient enters symptom description, the ChatBot may grasp the key words and provide some possible curing measures. As for the doctors, the ChatBot may be used as an encyclopedic reference, where they can obtain comprehensive information about the disease of interest.

2.3.2 Test for more types of disease

Apart from pneumonia, there are numerous kinds of diseases where computer vision techniques may also be instrumental. Alzheimer is a typical example of such malady where scientists have paid much efforts. Based on such consideration, it would be of great benefit if we can augment the functionality of our model, so that the input covers a wider range of medical image types, and a more diverse variety of diseases can be automatically diagnosed by utilizing our model. Considering

that Numpy alone may not be sufficient for data preprocessing when more diseases are accounted, we would also exploit Pandas and some other libraries when necessary.

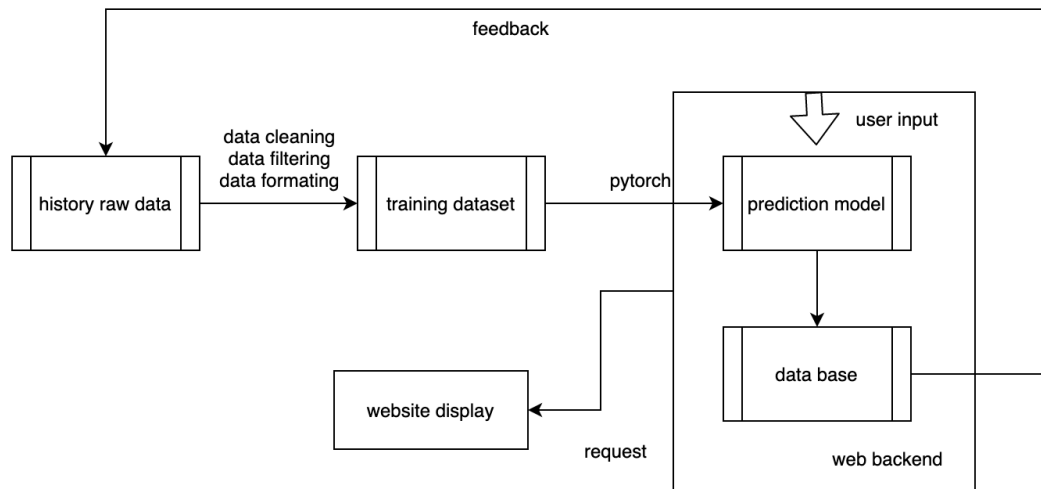


Figure 5: Data flow of the expected system (including the advanced functions)

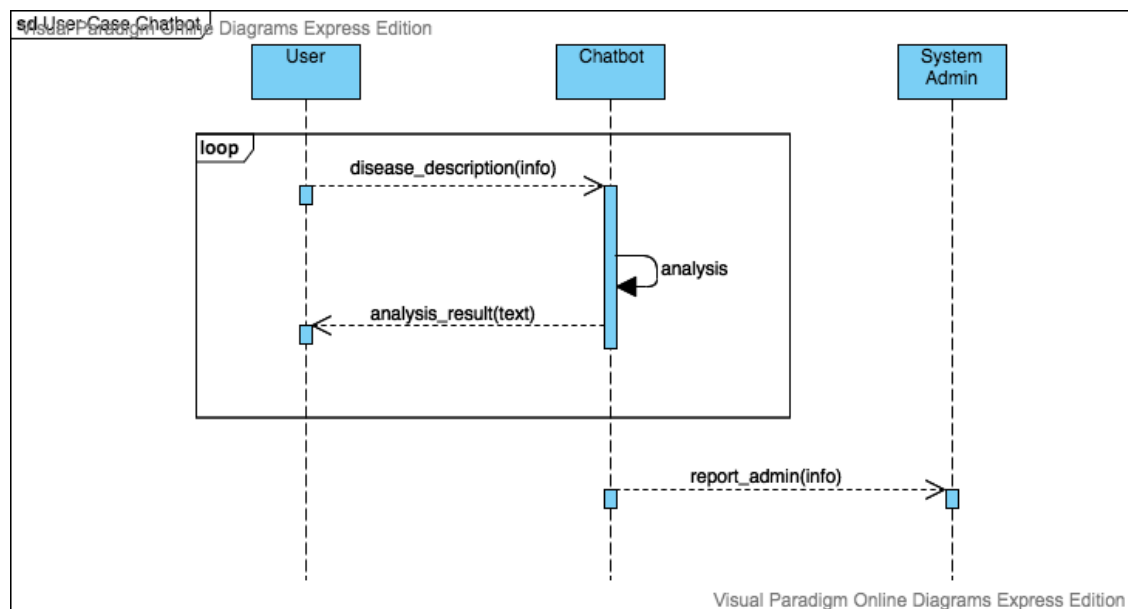


Figure 6: Sequence Diagram for User Case Four - Chatbot

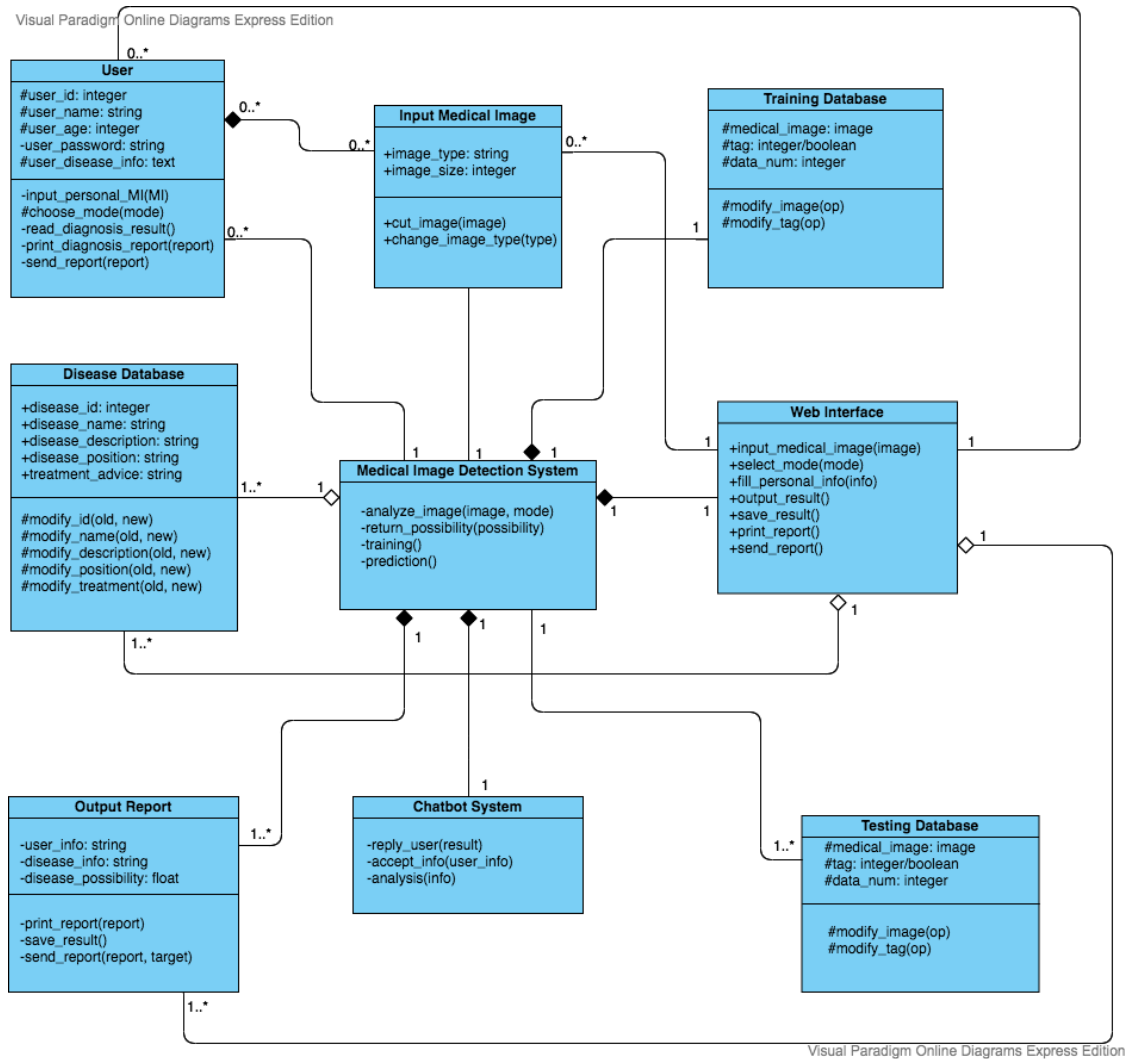


Figure 7: Class Diagram for Medical Bot System with the Advanced Functions

3 Software Testing

3.1 Test Planning

The objective of the test is to verify that the functionality of the Web-based Medical Bot works as the specifications are intended. Our test programs will execute and verify the test scripts, identifying errors, anomalies, undesirable features, or information about the software's non-functional attributes.

The final products of the test are the following two:

- A production-ready software;
- A set of reliable test scripts that can be reused for Unit and Functional test execution.

Figure (8) shows the model of the testing process ¹.

¹From textbook Fig 8.3

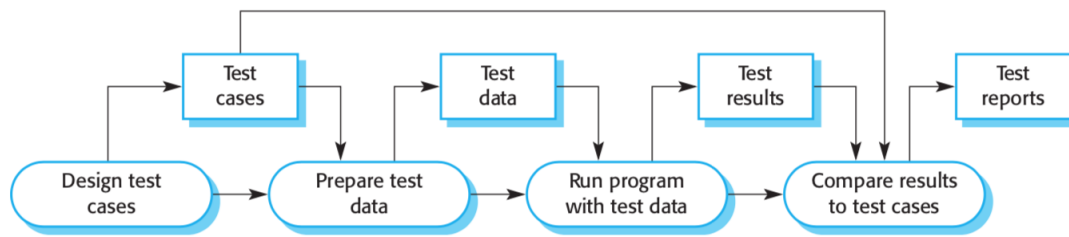


Figure 8: A model of the software testing process

3.1.1 Test Assumptions

- Pre-processed medical data required are available in the system prior to the start of ML model training.
- Performance Testing is considered for ML model training.
- Test environment will be set up in all team member's laptops.
- Team members will be paired up to review all the test deliverables.
- The system will be tested as a black box; if the information shows correctly online and in the reports, it will be assumed that the trained models are working properly.

3.2 Test Design

3.2.1 Development Testing

We will be carrying out the following three stages of testing

1. *Unit testing*: Unit tests will be carried out with the development of the software against individual function units or object classes to ensure every core functions' full functionality. Example: use prepared test data as input to see if the `predict()` function outputs valid results; input an image to check if function `read_image()/save_image()` reads/saves the image as intended;
2. *Component testing*: Tests against composite components comprising a group of individual units. Example: check camera component as a whole to see if its allowing people to take/upload medical images; check result-displaying component on the front end if its functioning as intended; object interfaces validation;
3. *System testing*: All components are integrated and the system is tested as a whole. The test focuses on validating the interactions between the components and objects, and case-based tests will be utilized. Example: test reusable components to check that they work as expected when they are integrated with new components; walk through the process view diagrams (Figures (2), (3) and (4)) and check the responses.

3.2.2 Release testing

We will test if our software meets all the requirements in this step. We are to develop several related tests:

1. Set up a patient account with no medical record. Log in. Upload photos, get a diagnosis report, and prescribe to doctoral suggestions if any.
2. Set up a patient account with medical records. Log in. Upload photos, get a diagnosis report, and prescribe to doctoral suggestions.

3.2.3 User testing

We will be doing a *Beta testing*, where a release of the software is made available to a group of users to allow them to experiment and to raise problems that they discover with the system developers.

3.3 Test Execution

Tests will be developed incrementally as the project is developing, and we will run regression tests to check that changes to the project have not introduced new bugs.

4 Work Division

4.1 Labor Division

- Implementation of CV and ML algorithms
- Data processing
- Model training and testing
- Web development
- Chatbot development

4.2 Timeline

Date	Activity 1	Member	Activity 2	Member
- March 14	1. Determine the project topic 2. Literature review 3. Determine the application requirement 4. Determine the machine learning model	Zheyuan Zhou Jiazhen Liu Chunyue Xue Shiting Xiao Yaqian Chen		
March 14 - 21	1. Write the midterm proposal 2. Prepare computer environment 3. Determine the ML algorithm and web framework	Zheyuan Zhou Jiazhen Liu Chunyue Xue Shiting Xiao Yaqian Chen		
March 21 - 29	Integrate the proposal components and slight modification	Zheyuan Zhou Jiazhen Liu Chunyue Xue Shiting Xiao Yaqian Chen		
March 29 - April 5	Learn the basic CV and ML algorithm	Jiazhen Liu Chunyue Xue Shiting Xiao	Learn the basic web implementation	Zheyuan Zhou Yaqian Chen
April 5 - 26	Develop the prediction model (including data pre-processing, training and testing) for basic application requirement (Tumor recognition and initial diagnosis)	Jiazhen Liu Chunyue Xue Shiting Xiao	Develop the website including database, web interface, web frontend for basic application requirement (Tumor recognition and initial diagnosis)	Zheyuan Zhou Yaqian Chen
April 26 - May 3	Test the basic web functions and conduct slight modification	Zheyuan Zhou Jiazhen Liu Chunyue Xue Shiting Xiao Yaqian Chen		
May 3 - 10	1. Search the relative psychology information for real-time monitoring of patients' mental health 2. Develop the online registration function	Jiazhen Liu Chunyue Xue Shiting Xiao	1. Develop the ChatBot function for patients to directly consult doctors 2. Include the consulting room function based on LAN which supports the online consulting	Zheyuan Zhou Yaqian Chen

Table 1: Timeline for the development process

References

- [1] China National Health Commission. Chinese Clinical Guidance for COVID-19 Pneumonia Diagnosis and Treatment (7th edition). 2020.
- [2] M. B. A. Miah and M. A. Yousuf. Detection of lung cancer from ct image using image processing and neural network. In *2015 International Conference on Electrical Engineering and Information Communication Technology (ICEEICT)*, pages 1–6, 2015.
- [3] F. Taher and R. Sammouda. Lung cancer detection by using artificial neural network and fuzzy clustering methods. In *2011 IEEE GCC Conference and Exhibition (GCC)*, pages 295–298, 2011.