Syracuse Hospital Management System

A secure and practical design

Ziqi Wang Xiaojun Zhang Yufei Fang Yue Zhao
Chenyang Du Huiyuan Li Hanyi Li
Haiyang Zhang

May 2018

Contents

1 Introduction **5**
- 1.1 Introduction . . . 5
- 1.2 The significance of development . . . 5

2 Development Environment **6**
- 2.1 Introduction to the Development Environment . . . 6
- 2.2 Introduction to Development Tools . . . 6

3 System Design **7**
- 3.1 Summary Design . . . 7
  - 3.1.1 System Function Analysis . . . 7
  - 3.1.2 Detailed Design Method . . . 9

4 Database Design **10**
- 4.1 Introduction of Database . . . 10
- 4.2 Requirement Analysis . . . 11

5 System Implementation **15**
- 5.1 System login module implementation . . . 15
  - 5.1.1 Flow Chart . . . 15
  - 5.1.2 User Interface . . . 16
- 5.2 The realization of the main interface . . . 18
  - 5.2.1 Administrator main interface . . . 18
  - 5.2.2 Reception main interface . . . 19
  - 5.2.3 Doctor's main interface . . . 21
  - 5.2.4 Warehouse keeper home screen . . . 22

6 Security Analysis **23**
- 6.1 SQL Injection . . . 23
  - 6.1.1 Introduction . . . 23
  - 6.1.2 Attack Simulation . . . 23
  - 6.1.3 Countermeasure . . . 24
- 6.2 CSRF attack . . . 26
  - 6.2.1 Introduction . . . 26
  - 6.2.2 Attack Simulation . . . 26
  - 6.2.3 Countermeasure . . . 28
- 6.3 XSS attack . . . 29

6.3.1 Introduction 29
6.3.2 Attack Simulation 30
6.3.3 Countermeasure 31
6.4 Password Protection 33
6.4.1 Hidden Password 33
6.4.2 Hashed Password 33

7 System Evaluation and Analysis 34
7.1 Test Case 34
7.2 Main functionality 40
7.2.1 Main functionality 40
7.2.2 Details 40
7.3 Features 41
7.3.1 Log Management 41
7.3.2 Coherence and streamlined 41
7.3.3 Detail oriented 43

8 Conclusion and Future Plan 45
8.1 Conclusion 45
8.2 Future Plan 46

Abstract

Computer science continues to develop, to provide a great convenience to various fields. At present, the patients become more and more and medical institutions accounts more and more miscellaneous. Therefore, it is necessary to develop an electronic Hospital Management Information Systems with a simple, convenient, high reliability, high storage capacity, high security, long time preservation, and low cost.

This report firstly introduces development significance of the development of electronic medical record system and related technology. Then the system design tasks and detailed design methods are analyzed according to the needs, and focuses on the database structure design, and the main function modules of the real and related functions. Finally, it summarizes the system's functional characteristics, the existing problems, and the corresponding improvement program. This system mainly uses the current relatively popular in the JSP to develop, adopt suitable for small to medium sized projects the MySQL database. The system is mainly divided into three functional modules: user management module, patient management module and the warehouse management module, according to the different roles to show the corresponding menu, the main functions of the system have a personnel management, records management, warehouse management, and log view, etc.

1 Introduction

1.1 Introduction

The hospital information management system is an indispensable information management system for today's medical institutions. It can provide sufficient space and a really quick way to search prescription for each medical institution to manage patient and physician information, so that it is greatly convenient for medical institutions. With the proper management of managers, hospital information management system is very useful for the managers of medical institutions.

With the rapid development of science and technology, computer science has also become more and more mature, almost all offices are benefit from computers. Computers have occupied a very high position in human life and work. In the past, prescriptions were mostly handwritten, complicated and time-consuming, and after the prescriptions were generated, it was not easy to save them, which required a lot of manpower and material resources, also it was inconvenient to find them. Different from traditional manual input, electronic prescriptions are simple and fast to use, have high reliability, large storage capacity, good confidentiality, long life, and low cost. It is conducive for the management of patient's basic data and tracking. At the same time, it can also be used to find information about the patient's past medical treatment and the corresponding departments and doctors, which can simplify the process of seeking medical treatment again. These advantages can greatly improve the efficiency of patient and physician management.

1.2 The significance of development

Nowadays, the popularity of computers has been really high. At the same time, its performance has also been qualitatively improved, thus it is used in various fields, which has been indispensable in our learning and working life. Nowadays, with the improvements of hospital's scale and scope of business, simple labor control can no longer meet the needs of current medical institutions. At the same time, manual operations will inevitably lead to errors. As far as the current medical resources are concerned, the ordinary manpower cannot satisfy the growing number of patients. Thus it can be seen that the previous manual mode has not been adapted to the development of the times. It not only wastes a lot of manpower and material resources, but also delays

the treatment of patients. Therefore, this traditional management method will be replaced by information management is an inevitable result.

As a team from Computer Science major, we hope that through our own efforts to develop an electronic hospital information management system that can solve the above problems, to strengthen the hospital management, improve the quality of medical care, and provide a reliable solution for the hospital which can safely and efficiently store patient information over the years. When there is a demand, we can quickly find out all kinds of information for patients and physicians. This can effectively help medical institutions manage patients and physicians' data.

2 Development Environment

2.1 Introduction to the Development Environment

Hardware system	PC, 1G graphics card Intel i3 CPU, 4G physical memory
Software System	Windows 7 Operating System Myeclipse 10.0 Tomcat 7.0 server
Database Software	MySQL, MySQL workbench

2.2 Introduction to Development Tools

Myeclipse10.0 is a mainstream development tool for Web applications. It adds a lot of Web-related components based on eclipse. For example, it integrates some kinds of mainstream frameworks such as Spring, Struts. Also other mainstream servers such as Tomcat and JBoss can also be well supported. It also integrates maven, which is widely used project management tool. It provides a large number of plug-ins. Through these plug-ins, Myeclipse can perfectly support the development of web projects such as JSP and Servlet as well as various complicated interface designs and rapid implementation of various functions.

At the same time, it also integrates three major frameworks and can be quickly implemented. By using these plug-ins, our coding tasks will be quite simple and the design of the interface will take us less time. By using Myeclipse, the efficiency of project development and the reliability of program

operation have been significantly improved.

MySQL workbench is a graphical management tool of MySQL database from Webyog company. The software is simple and easy to use. At the same time, the interface is concise. So the code operation in the CMD command window can be operated through a visual window in the software, such as building a table, inserting Data, modify the table structure and other operations can be quickly completed by using the software. At the same time, the backup and restore of the database and the batch operation of the SQL script also can be quickly completed with the software. There're usually three ways for a JSP system to access the database:

1. Access the database by using the basic JDBC API
2. Use the data access variables provided by JSP to access the database;
3. Access ODBC API functions by using the interface provided by ODBC.

There are three main data access APIs: data access objects, remote data objects, and ActiveX data objects. We mainly used JDBC way to access the database in our system. Similar to the role of the filter in other applications, the main function of the filter in JAVAWEB is to intercept all user requests and then perform related filtering operations. The most commonly used functions are character set filters and permissions. check.

3 System Design

3.1 Summary Design

3.1.1 System Function Analysis

Based on the information we collected from hospital management needs currently, the following functions of the management system are expected to be achieved:

1. User Management Module
   (a) The administrator can create/modify/delete the details of other owners, including the password.
   (b) Non-administrators can modify their own relevant information.

2. Receptionist module

   (a) The reception receives appointment from patient, then they add the appointment into the system, message will be sent to doctor automatically.

   (b) After doctor determines which medicine patient should take, they will send the result to reception by system, pricing and charging will be done and the patient can pick up their medicine in the warehouse.

3. Warehouse keeper Module

   (a) The warehouse module can add, modify, delete, and query detailed drug information.

   (b) Warehouse manager adds the storage information before the goods are purchased in order to make sure that warehouse always have enough.

According to the detailed analysis of the above functional modules, the entire electronic management system is decomposed into a module structure diagram as Figure 1.

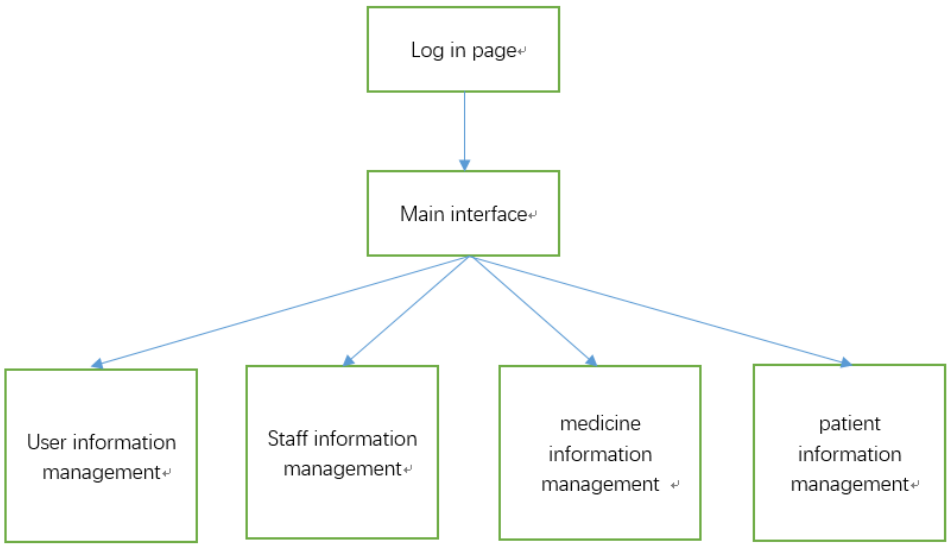
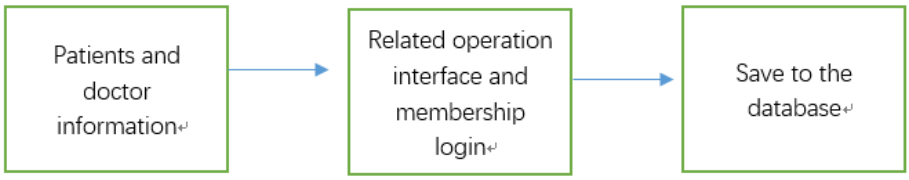


Figure 1: Module Structure Diagram

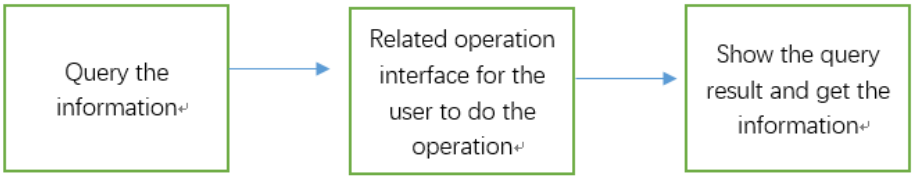
3.1.2 Detailed Design Method

The detailed design method is divided into the most common program flowcharts, N-S box diagrams, and the IPO diagrams we will use below. Through these graphical modes, the process of our system design and development is much more clear.

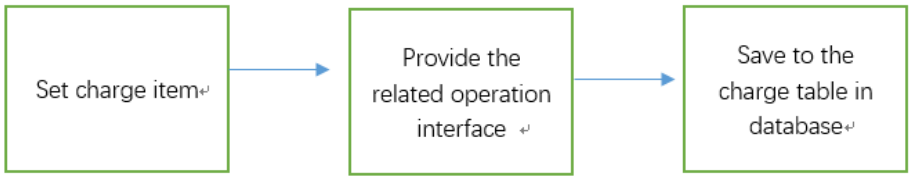
1. patients and doctors management model IPO



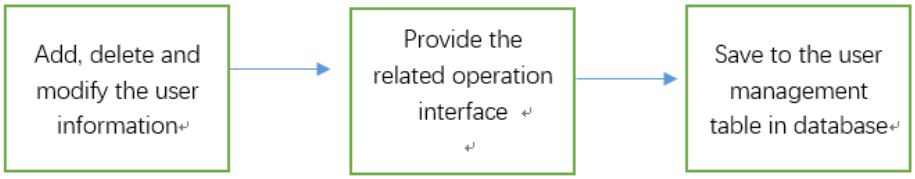
2. query model IPO



3. charge management model IPO



4. user management model IPO



4 Database Design

4.1 Introduction of Database

The definition of a database: A database generally refers to the consolidation of a large amount of organized and simultaneously sharable data stored on a computer for a long time. The basic characteristics of the database:

1. Data is usually organized and stored according to a certain model.
2. The database can be shared by various users and multiple applications.
3. Little redundancy between data.
4. The independence between data is high.
5. easy to expand.

4.2 Requirement Analysis

According to the investigation of the actual situation of the patients and physicians in the relevant hospitals, the four most important roles in the medical record management system are summarized:

- Administrator: the main responsibility is to manage all users.
- Receptionist: It is mainly responsible for receiving patients at the front desk, registering patients, and handling prescription-related matters, including price allocation, payment, and medication.
- Doctor: It is mainly responsible for giving patients medical treatment, filling out medical records, and prescribing prescriptions
- Warehouse keeper: Mainly responsible for some of the drug management, including drug information collection, drug storage, drug sales.

1. In the system user table, there are mainly four types of users in this system.

   - System administrator: Its main function is to manage all other user's information, but also includes adding users, viewing all information of other users, and modifying their own relevant information.
   - Receptionist: Can help patients to register, manage prescriptions, modify their own relevant information.
   - Doctor: Can fill in the case and prescribe the patient, modify their own relevant information
   - Warehouse keeper: Relevant information for the management of drugs in warehouses, including additions and deletions of drugs, inventory of drugs, inventory of drugs, and modification of their own relevant information.

   Based on the above analysis, we get the table structure of the table as shown in Table 2.

2. The inventory table manages all drug information, including the time of the drug's storage, the storage data, the storage batch, and the total price. This table is mainly performed by the warehouse keeper

name	type	length	Primary key	meaning
id	int	11	primary key	User id
uname	varchar2	255		username
upass	varchar2	255		password
Utype	varchar2	255		User type
tname	varchar2	255		User real name
Sex	varchar2	255		Gender
Age	varchar2	255		age
Tel	varchar2	255		tel
addrs	varchar2	255		address
Filename	varchar2	255		photo

Figure 2: System user table

for related additions, deletions, modifications, and query operations. Other users do not have the right to manage inventory information, as shown in Table 3.

name	type	length	Primary key	meaning
id	int	11	Primary key	Inventory id
name	varchar2	255		Medicine name
rkdate	varchar2	255		nventory data
tnum	varchar2	255		quantity
batch	varchar2	255		batch
totprice	varchar2	255		Total price

Figure 3: Warehouse table

3. The health history table contains the main relevant information of the medical record, such as ID, medical record number, relevant patient ID number, etc., which are mainly related to the doctor to add, delete, modify, query operations. The table structure is shown in Table 4.

4. A charge table, which manages all charge information, including the medical record number, drug information, total price, and related status of the charge amount. The table is mainly related to the window personnel to increase, modify, query operations, other people can not

name	type	length	Primary key	meaning
id	int	11	Primary key	Health history id
blno	varchar2	255		Health history number
pname	varchar2	255		Patient name
ssn	varchar2	255		Social Security Number
Sex	varchar2	255		Sex
birth	varchar2	255		Birth date
sym	varchar2	255		Symptom

Figure 4: Warehouse table

name	type	length	Primary key	meaning
id	int	11	Primary key	charge id
blno	varchar2	255		Health history number
mname	varchar2	255		Medicine name
totprice	varchar2	255		Total price
status	varchar2	255		status

Figure 5: charge table

perform related operations on the table, the table structure shown in Table 5.

5. Medicine table, with the continuous increase of drugs, electronic information management has become very necessary. The drug table manages all relevant information of drugs, including drug names, drug manufacturers, drugs to adapt to symptoms, and drug related taboos. , the unit price of drugs and the price of medical insurance, as well as some other information. This form is filled in by the library manager. Other personnel cannot change the relevant information of the table. The table structure is shown in Table 6.

6. Prescribe information table manages the medication information that the doctor prescribes to the patient, including the relevant medical record number, drug information, and related status. The table is managed by a doctor. Other users do not have the right to modify the relevant information of the table. The table structure is shown in Table 7.

name	type	length	Primary key	meaning
id	int	11	Primary key	Medicine id
mname	varchar2	255		Medicine name
factory	varchar2	255		Medicine factory
sym	varchar2	255		Symptom
se	varchar2	255		Side effect
price	varchar2	255		Unit price
member	varchar2	255		Member or not
mbprice	varchar2	255		Member price
com	varchar2	255		comment
Filename	varchar2	255		picture

Figure 6: Medicine table

name	type	length	Primary key	meaning
id	int	11	Primary key	Medicine id
blno	varchar2	255		Health history number
med	varchar2	255		Medicine name
num	varchar2	255		quantity
status	varchar2	255		status

Figure 7: Prescribe information table

name	type	length	Primary key	meaning
id	int	11	Primary key	Log id
userid	int	11		User id
Uname	varchar2	255		username
blno	varchar2	255		Health history number
Createtime	varchar2	255		Operation time
ssn	varchar2	255		user ssn

Figure 8: log table

7. log table, in any management system, log information is an important part of the system, this system is no exception. In this system, the log information records the user's related operations, such as login, exit information, the doctor's increase in medical records, modify operations, etc. It is of great significance to the related operation and maintenance in the future. The log system is automatically created by the system when the user performs the related operations. At the same time, only the system administrator can log the system and view related log information. Based on the above analysis, The table structure of this table is shown in Table 8.

5 System Implementation

5.1 System login module implementation

5.1.1 Flow Chart

The Flow chart is shown as Figure 9

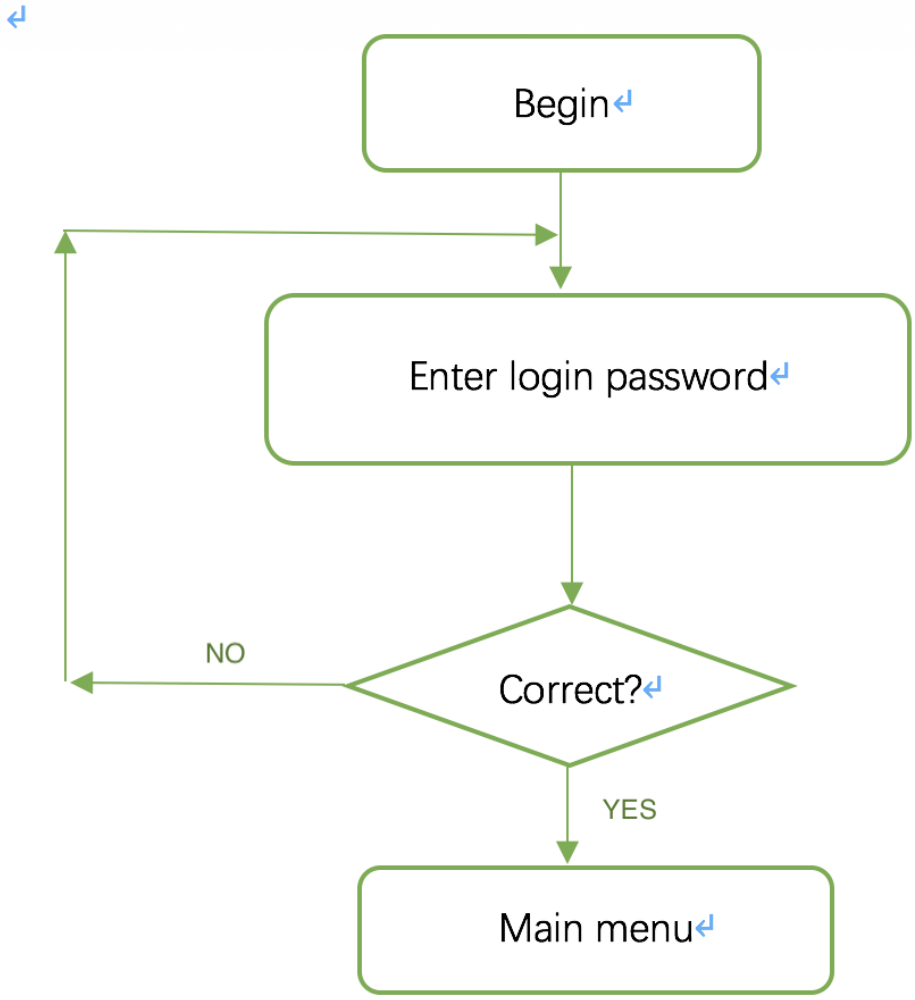


Figure 9: Flow Chart

5.1.2 User Interface

The User Interface is shown as Figure 10

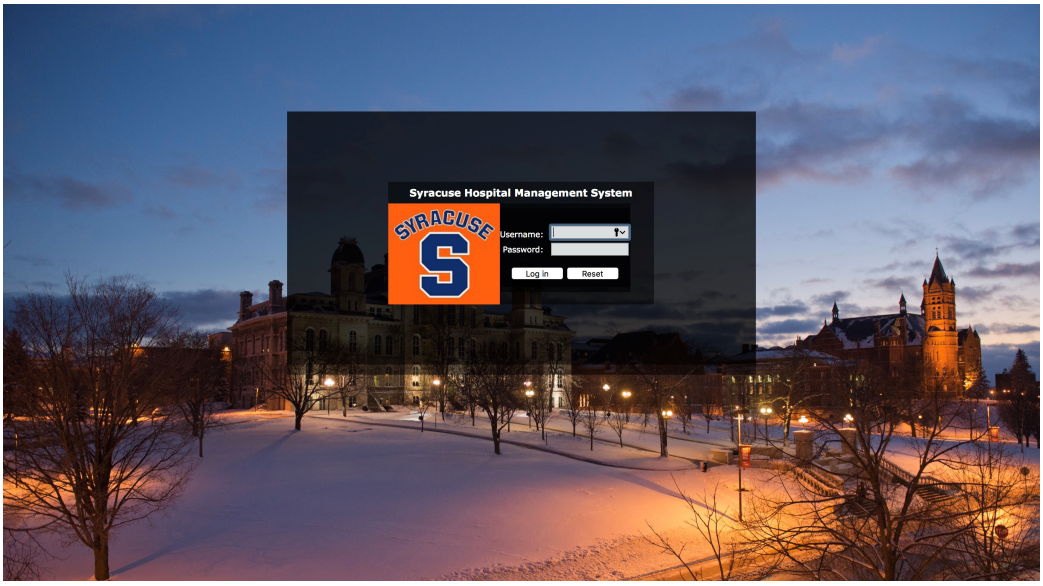


Figure 10: Interface

Relative functions:

1. As we can see, this interface consists of two input boxes and two buttons. After the user enters his own information into the corresponding input box, click the login button to submit the data in the input box.
    - If the user enters the correct information, the system will let the client jump to the main page.
    - If there is any inconsistency, the corresponding error message is given.
2. Input: user name and password.
3. Relevant processing:
    - First, the validity of the user name and password is checked through the check of the foreground JS, mainly to prevent the user from entering illegal characters.
    - After the verification of the front desk JS, the system will pass the user submitted data the day after tomorrow, let the server to check whether the user name exists or the password is correct.

4. Output:

   - If the user login is successful, he enters the user's system home page, and displays the related menu interface and provides the corresponding role operation according to the user's related type utype.
   - If the login is unsuccessful, an error information page is displayed.

5.2 The realization of the main interface

5.2.1 Administrator main interface

The administrator's main interface is expressed in the form of a menu, as shown in figure 11.

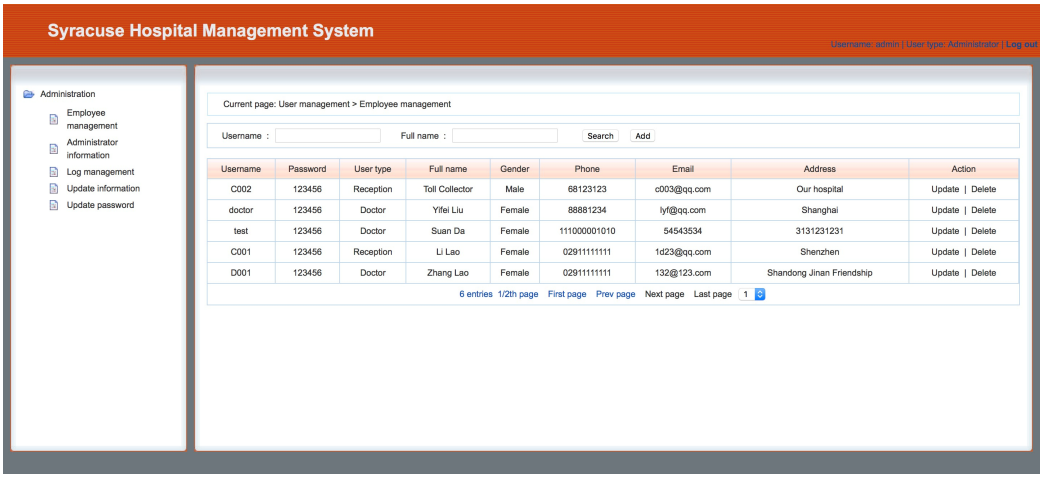


Figure 11: Administrator interface

The main functions are as follows:

1. Staff management: The management of other roles except the system administrator, including the addition, modification, and deletion of their passwords and other detailed information, can also be performed on the user based on the user name and name.

2. Administrator user information: In addition to managing general employee information, administrators can also manage administrator related information, such as adding administrators.

3. Log Management: Administrators can view and search the log information in the system, including searching according to the operator, and searching according to the operation record number. In this system, the log information is used in related operations, the system automatically in the database The relevant log information is inserted, where the log information is mainly divided into two types, one is an ordinary login operation, and the other is a new modification operation of the medical record, and the log records the operator, operation time and other related information.

4. Modify personal information: Administrators can make some modifications to their own details.

5. Modify login password: According to the existing password administrator, you can update your own password.

5.2.2 Reception main interface

The window staff main interface is represented by a menu, as shown in Figure 12.

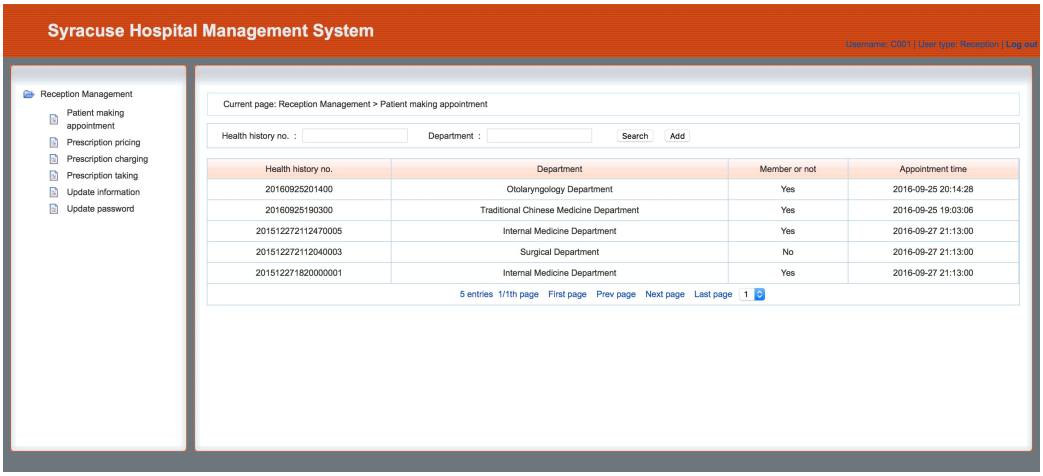


Figure 12: Reception interface

The main functions are as follows:

1. Patient Registration: After the patient arrives at the hospital, the patient is registered at the window. The window personnel will register the relevant information and inform the patient of the patient's medical record number. The patient can find the doctor in the relevant department with the patient's medical record number.

2. Prescription markup: When the doctor fills out the medical record and opens the benefits, the patient comes to the window with a good case for the prescription price.

3. Prescription payment: After the prescription price is completed, the patient will perform the payment operation and calculate different according to whether the patient has medical insurance.

4. Prescription medicine: After the patient completes the doctor's fee payment operation, he can go to the window to take medicine. However, if the patient has not paid, the window person cannot find the patient's medicine information in the medicine picking list.

5. Modify Personal Information: The window personnel can modify their own details after logging in to the system.

6. Modify login password: After logging into the system, the window personnel can update their own password based on the existing password.

5.2.3 Doctor's main interface

The doctor's main interface is represented by a menu, as shown in Figure 13.

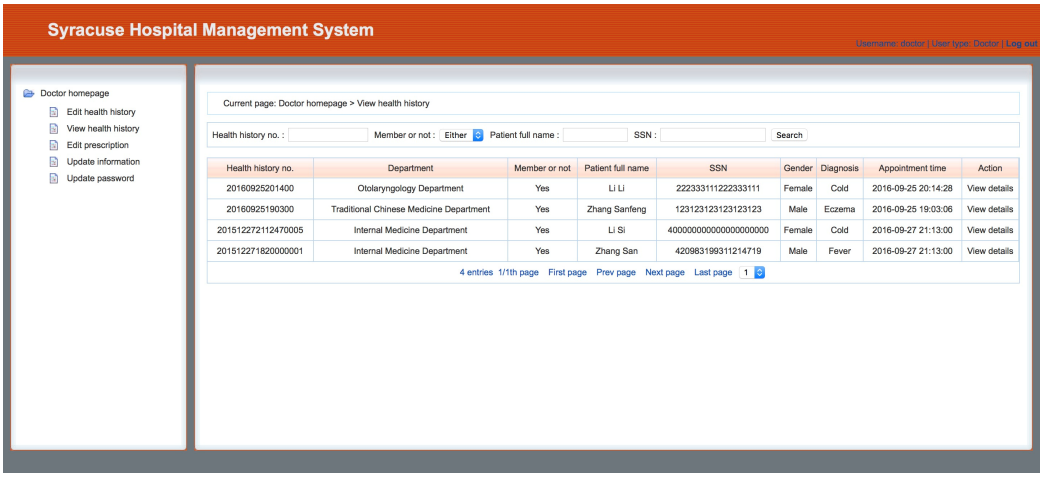


Figure 13: Doctor interface

The main functions are as follows:

1. The patient: The doctor can find the patient's medical record according to the patient's medical record number and fill in the relevant information of the medical record.

2. medical records view: doctors can search for relevant medical records based on different information of patients, including patient's name, ID number, medical record number and other relevant information.

3. Prescription management: After the doctor fills out the patient's medical records, he can prescribe drugs to the patient. In this interface, the doctor can add prescription information, view the prescription information and find relevant prescriptions based on the medical record number and drug name, modify and delete. Related prescription information.

4. modify personal information: After the doctor logs on the system can modify their own detailed personal information.

5. modify the login password: Like other users, doctors can also follow their own password to keep up with their own password.

5.2.4 Warehouse keeper home screen

The drug administrator is represented by a menu, as shown in Figure 14.

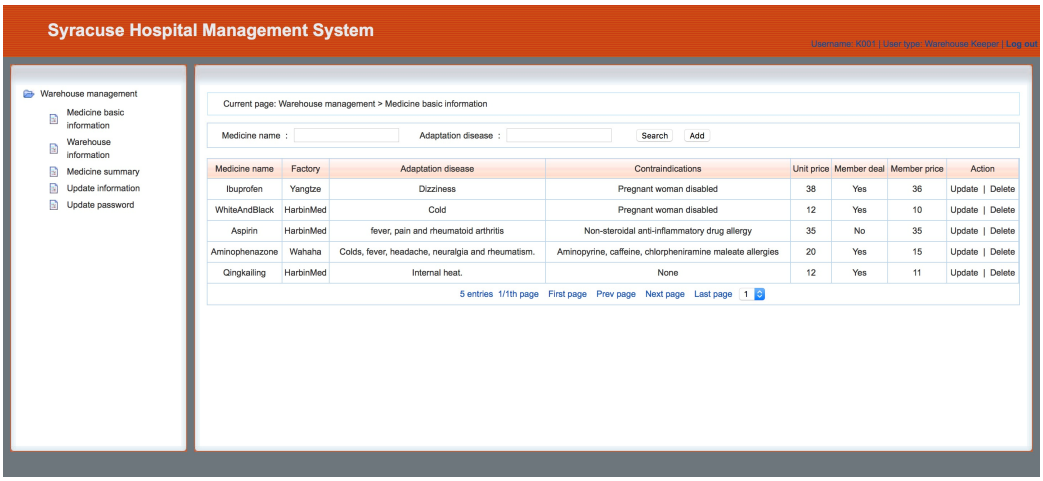


Figure 14: Warehouse keeper interface

The main functions are as follows:

1. Drug basic information management: After the warehouse administrator logs in the system, he can add new drugs or modify existing drug information through the form, and can also query the corresponding drugs based on the drug name and applicable symptoms.

2. Management of drug storage information: The library management personnel can add information on the storage of drugs, including information on time of storage, batch number of storage, etc., and can modify the relevant information of storage, and can also check according to the drug name and storage time. Related storage information.

3. Drug inventory: The warehouse personnel can view the inventory information of the drug, including the inbound quantity, current inventory, total purchase amount and total sales, and purchase the goods according to the current inventory and sales.

4. modify personal information: inventory personnel can modify their own details after logging in the system.

5. modify the login password: inventory personnel can log in to the system can update their own password based on the existing password.

6 Security Analysis

In the section, we would like to discuss the potential attacks for our system, and our practical countermeasures to eliminate these risks.

6.1 SQL Injection

6.1.1 Introduction

SQL injection is a code injection technique, used to attack data-driven applications, in which nefarious SQL statements are inserted into an entry field for execution (e.g. to dump the database contents to the attacker).

Usually, it attacks by inserting an SQL statement into a data field, where only a number or string is wanted, to change the condition in where clauses or others. As a result, it may bypass some conditions to visit some pages or to view some table entries which should not be accessed by the attacker.

6.1.2 Attack Simulation

This is extremely dangerous for our hospital management system, because an attacker may easily login as whatever role he wants, i.e. administrator, reception, doctor or warehouse. Consequently, all secrete data like health history records, employee personal information will be stolen, and the data in the database may be modified or deleted.

Here is an example of how an attacker login as administrator using SQL injection. We insert such information in the login page.

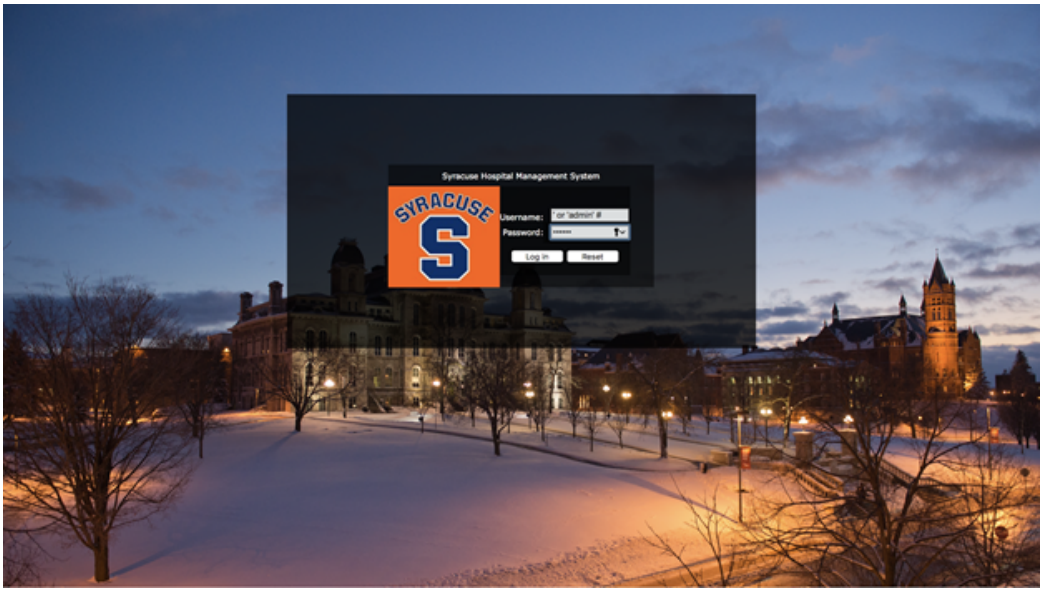


Figure 15: Login with malicious input

Click *login* and see what will happen.

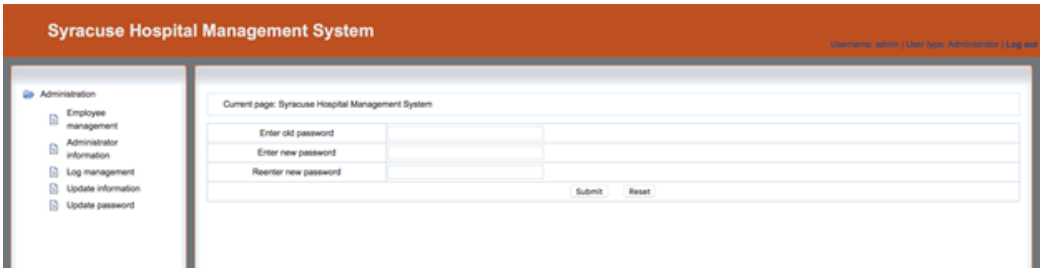


Figure 16: Login success

6.1.3 Countermeasure

A most common countermeasure of SQL injection is 'prepared statement'. It means to create a query format with question signs. When a query request is sent to database, we replace the question signs in the format with actual data. Whatever we write will be treated as data, rather than a part of the

query conditions. Luckily, there is such method provided in JAVA, we just have to use provided class to achieve the countermeasure. Here is the code.

```
String select_str1 = "select * from sysuser where binary uname=?";
PreparedStatement select_statement1 = dao.getConn().prepareStatement(select_str1);
select_statement1.setString(1,username);
ResultSet rs = select_statement1.executeQuery();
List<HashMap> list = resultSetToList(rs);
rs.close();
select_statement1.close();
if (list.size() == 1) {
    HashMap map = list.get(0);
    List<HashMap> ulist = dao
            .select("select * from sysuser where binary uname='"
                    + username + "' and upass='" + password
                    + "'");
    String select_str2 = "select * from sysuser where binary uname=? and upass = ?";
    PreparedStatement select_statement2 = dao.getConn().prepareStatement(select_str2);
    select_statement2.setString(1,username);
    select_statement2.setString(2,password);
    ResultSet rs2 = select_statement2.executeQuery();
    List<HashMap> ulist = resultSetToList(rs2);
    rs2.close();
    select_statement2.close();
    if (ulist.size() == 1
            && password.equals(map.get("upass").toString())) {

        request.getSession().setAttribute("admin", map);
        dao = new CommDAO();
        HashMap<String,Object> ext = new HashMap<String,Object>();
        ext.put("userid", map.get("id"));
        ext.put("tname", map.get("tname"));
        ext.put("oper", "Login");
```

Figure 17: Prepared Statement

After applying prepared statement method, let's try to repeat the attack. Here is the result.

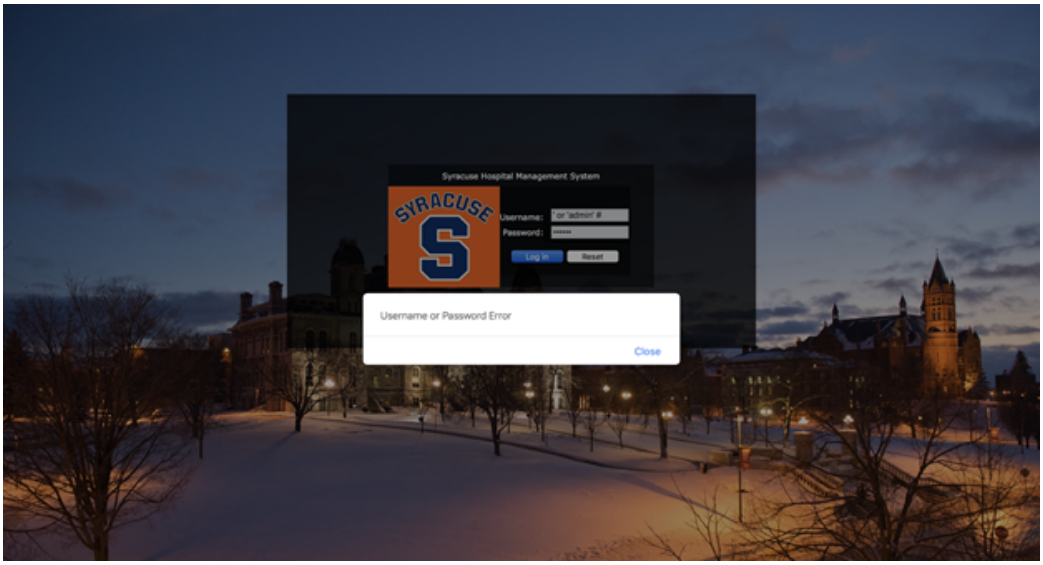


Figure 18: Can't login

As we can see, the attacker is unable to login as administrator now! SQL injection attack is prevented.

6.2 CSRF attack

6.2.1 Introduction

Cross-site request forgery, is a type of malicious exploit of a website where unauthorized commands are transmitted from a valid user that the target web application trusts. There are many ways in which a malicious website can transmit such commands. With a little help of social engineering (such as sending a link via email or chat), the attacker can lead the victim to access the malicious website, which may contains specially-crafted image tags, hidden forms, and JavaScript XMLHttpRequests, to send the unauthorized commands.

6.2.2 Attack Simulation

With CSRF attack, an attacker may trick the victim into executing actions of the attacker's choosing, without the victim approval. If the victim is a

normal user, a successful CSRF attack can force the user to perform state changing requests like transferring funds, changing their email address, and so forth. If the victim is an administrative account, CSRF can compromise the entire target web application.

In our case, we write a malicious website to mock the behaviour of the attacker. After the victim login to the system, the victim access the malicious page and click submit.

Figure 19: malicious website mock

As a result, the victim's password has been modified.

id	uname	upass	utype	tname	sex	tel	email	addrs	filename
1	admin	123456	Administrator	admin	Male	13733422488	131313@qq.com	1312312312	no.jpg
8	K001	123456	Warehous...	Wang Lao	Female	02911111111	123@123.com	Building 3	no.jpg
9	D001	123456	Doctor	Zhang Lao	Female	02911111111	132@123.com	Shandong Jinan Friendship	no.jpg
10	C001	123456	Reception	Li Lao	Female	02911111111	1d23@qq.com	Shenzhen	no.jpg
12	test	123456	Doctor	Suan Da	Female	111000001010	54543534	3131231231	no.jpg
20	Test	123456	Administrator	Name	Male	13132131	3123123@qq.com	Shenzhen	no.jpg
21	doctor	123456	Doctor	Yifei Liu	Female	88881234	lyf@qq.com	Shanghai	no.jpg
22	C002	123456	Reception	Toll Collector	Male	68123123	c003@qq.com	Our hospital	
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Figure 20: Before CSRF attack

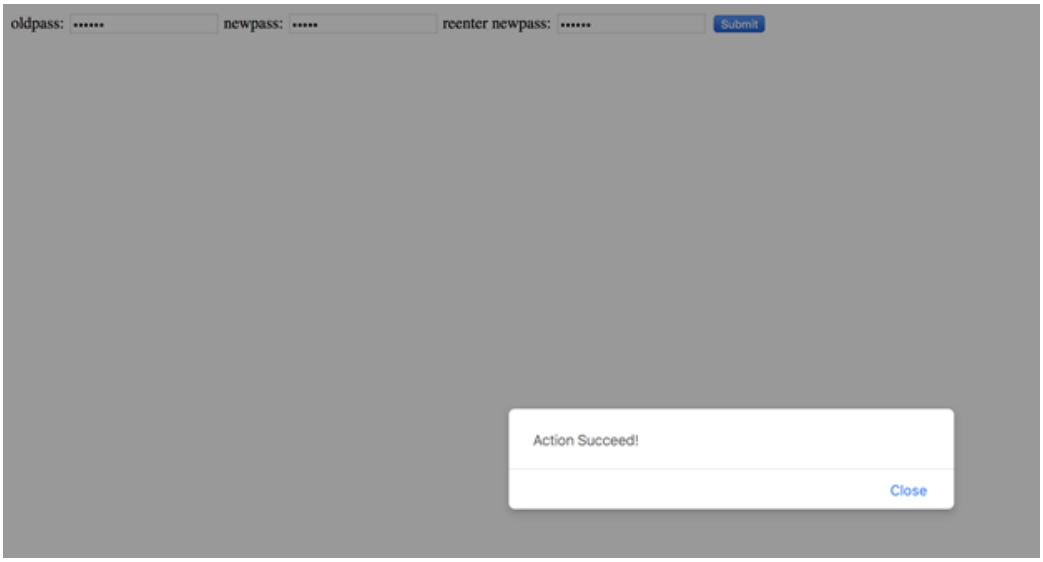


Figure 21: modify password success

id	uname	upass	utype	tname	sex	tel	email	addrs	filename	si
1	admin	123456	Administrator	admin	Male	13733422488	131313@qq.com	1312312312	no.jpg	20
8	K001	abcde	Warehous...	Wang Lao	Female	02911111111	123@123.com	Building 3	no.jpg	20
9	D001	123456	Doctor	Zhang Lao	Female	02911111111	132@123.com	Shandong Jinan Friendship	no.jpg	20
10	C001	123456	Reception	Li Lao	Female	02911111111	1d23@qq.com	Shenzhen	no.jpg	20
12	test	123456	Doctor	Suan Da	Female	11000000 1010	54543534	3131231231	no.jpg	20
21	doctor	123456	Doctor	Yifei Liu	Female	88881234	lyf@qq.com	Shanghai	no.jpg	20
22	C002	123456	Reception	Toll Collector	Male	68123123	c003@qq.com	Our hospital		20
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NU

Figure 22: After CSRF attack, the password has beem modified

6.2.3 Countermeasure

A practical countermeasure against CSRF is to ensure the request has the same origin with the web application domain. The referrer header, which can enable the new web page to see where the request originated, is a ideal target to verify whether the origin of request matches the target origin or not. Since the Referrer header can be only set by the browser, checking the Referrer header becomes a commonly used method of preventing CSRF.

In our case, we implement the Referrer header checking like the following Figure 23:

```
else if("uppass".equals(ac)){

    String clientadd = request.getRemoteAddr();
    String serveradd = InetAddress.getLocalHost().getHostAddress();
    if(clientadd.equals(serveradd)) {updatePass(request, response);}
    else{request.setAttribute("error", "");
         go("/admin/uppass.jsp", request, response);}
}
```

Figure 23: CSRF countermeasure

After enabling the countermeasure, we can observe the forged request from the malicious website has been blocked.

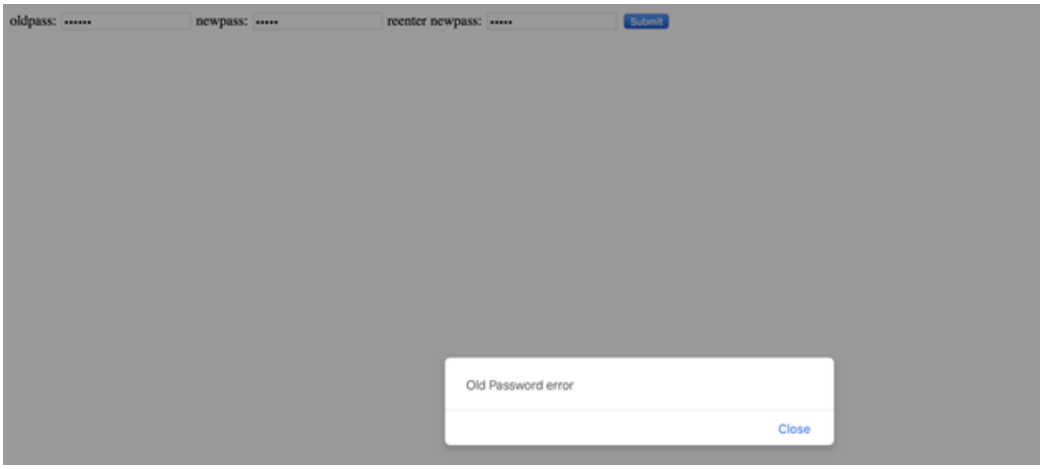


Figure 24: Unauthorized request has been blocked

6.3 XSS attack

6.3.1 Introduction

Cross-Site Scripting (XSS) attacks are a type of injection, in which malicious scripts are injected into otherwise benign and trusted web sites. XSS attacks

occur when an attacker uses a web application to send malicious code, generally in the form of a browser side script, to a different end user. Flaws that allow these attacks to succeed are quite widespread and occur anywhere a web application uses input from a user within the output it generates without validating or encoding it.

6.3.2 Attack Simulation

An attacker can use XSS to send a malicious script to an unsuspecting user. The end user's browser has no way to know that the script should not be trusted and will execute the script. Because it thinks the script came from a trusted source, the malicious script can access any cookies, session tokens, or other sensitive information retained by the browser and used with that site. These scripts can even rewrite the content of the HTML page.

Here is an example to show how a malicious script is inserted to our website. We modify the full name of an employee to be a script.

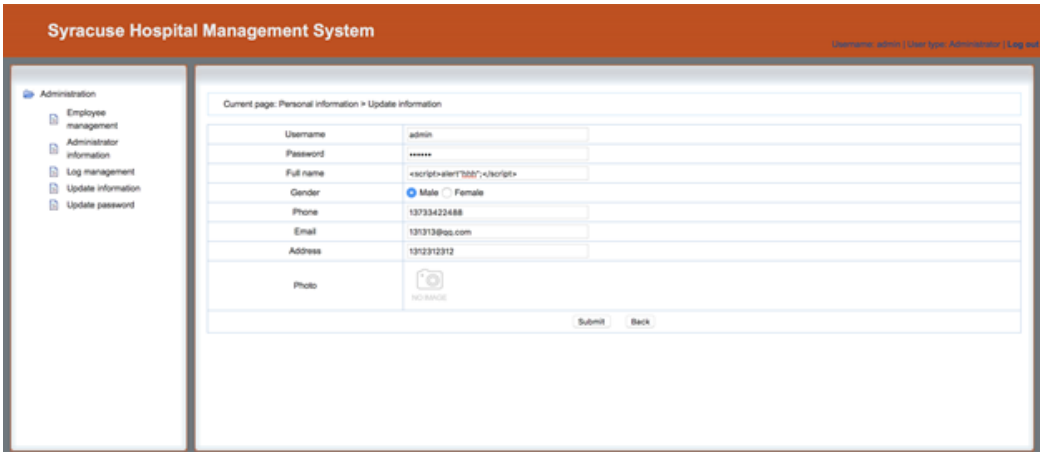
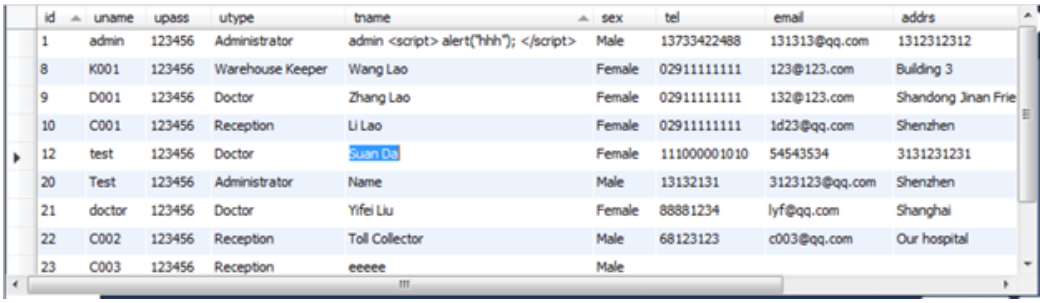


Figure 25: Input with script

Then we let the administrator login, and visit employee information page in to see what will happen.



id	uname	upass	utype	tname	sex	tel	email	addrs
1	admin	123456	Administrator	admin <script> alert('hhh'); </script>	Male	13733422488	131313@qq.com	1312312312
8	K001	123456	Warehouse Keeper	Wang Lao	Female	02911111111	123@123.com	Building 3
9	D001	123456	Doctor	Zhang Lao	Female	02911111111	132@123.com	Shandong Jinan Frie
10	C001	123456	Reception	Li Lao	Female	02911111111	1d23@qq.com	Shenzhen
12	test	123456	Doctor	Suan Da	Female	111000001010	54543534	3131231231
20	Test	123456	Administrator	Name	Male	13132131	3123123@qq.com	Shenzhen
21	doctor	123456	Doctor	Yifei Liu	Female	88881234	lyf@qq.com	Shanghai
22	C002	123456	Reception	Toll Collector	Male	68123123	c003@qq.com	Our hospital
23	C003	123456	Reception	eeeee	Male			

Figure 26: Improper input has been submitted

As we can see in Figure 27, there is an alert box shown in the page. It means that administrator's username and password information will be stolen by the attacker, if we change the content of the script. Therefore, our website is a potential victim of XSS attack

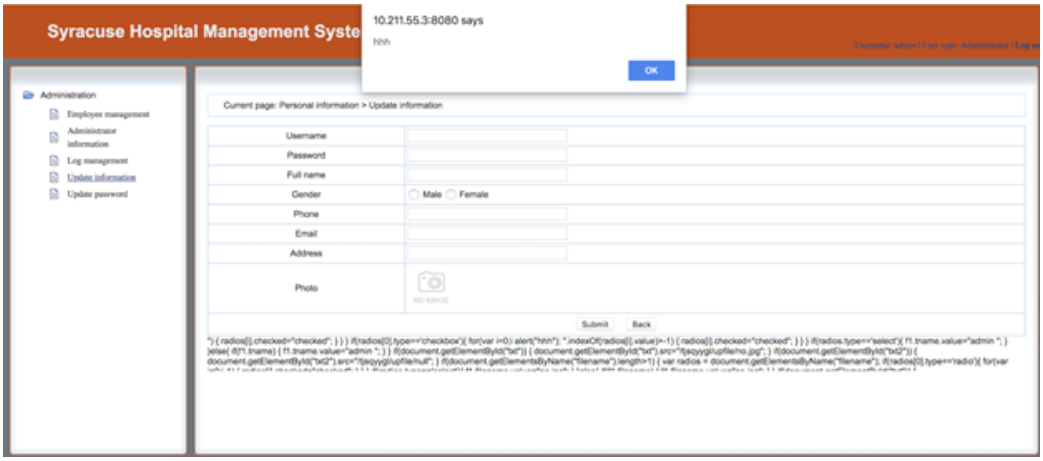


Figure 27: Alert box

6.3.3 Countermeasure

To prevent this attack what we have to do is to remove or replace these sensitive characters or words. For example, we can replace < and > with [and] . As a result, the original script won't be executed, because the string will never be treated as a script again.

Figure 28 is how we implement it.

```
private String protectXSS(String original) {
        return original.replace('<','[').replace('>',']');
       }
```

```
            value+=vstr+" ~ ";
        }
        if(value==null)value="";
        if(value.equals("null"))value="";
        if(value.length()>0)value=value.substring(0,value.length()-3);

        if(rsmd.getColumnTypeName(j).equals("int"))
        {
         sql+=rsmd.getColumnName(j)+"="+value+",";
        }else{
             sql+=rsmd.getColumnName(j)+"='"+protectXSS(value)+"',";
        }

    }else{

        if(extmap.get(rsmd.getColumnName(j))!=null)
        {
            if(rsmd.getColumnTypeName(j).equals("int"))
            {
             sql+=rsmd.getColumnName(j)+"="+extmap.get(rsmd.getColumnName(j))+",";
            }else{
                 sql+=rsmd.getColumnName(j)+"='"+protectXSS((String)extmap.get(rsmd.getColumnName(j)))+"',";
            }
        }
```

Figure 28: XSS countermeasure: Filtering

Now we repeat the attack and let the administrator visit employee information page again. As we can see, the sensitive characters are replaced, therefore XSS attack is prevented.

id	uname	upass	utype	tname	sex	tel	email	addrs	filename	si
1	admin	123456	Administrator	[script]alert'hhh';[/script]	Male	13733422488	131313@qq.com	1312312312	no.jpg	20
8	K001	123456	Warehous...	Xiaojun	Male	3159289672	xzhang@syr.edu	Syracuse	no.jpg	20
9	D001	123456	Doctor	Zhang Lao	Female	02911111111	132@123.com	Shandong Jinan Friendship	no.jpg	20
10	C001	123456	Reception	Li Lao	Female	02911111111	1d23@qq.com	Shenzhen	no.jpg	20
12	test	123456	Doctor	Suan Da	Female	11000001010	54543534	3131231231	no.jpg	20
21	doctor	123456	Doctor	Yifei Liu	Female	88881234	lyf@qq.com	Shanghai	no.jpg	20
22	C002	123456	Reception	Toll Collector	Male	68123123	c003@qq.com	Our hospital		20

Figure 29: Filtering works

6.4 Password Protection

The protect of password is a critical block of the whole system. To achieve this, we implement two measures in our system.

6.4.1 Hidden Password

To prevent user's password being pecked by others, we made it hidden when login or update password.

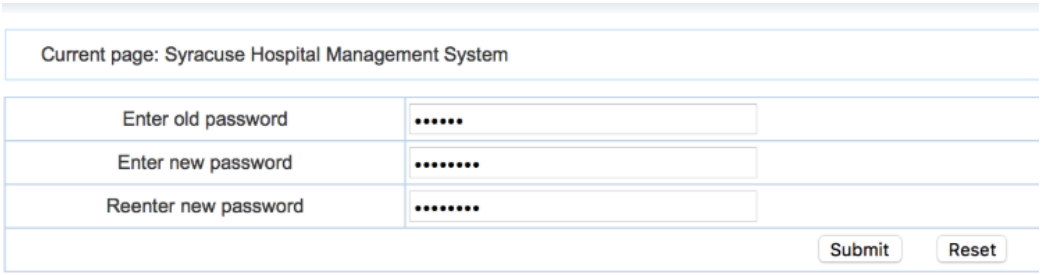


Figure 30: Hidden Password

6.4.2 Hashed Password

Instead of storing the plaintext of user password, the server only keep the hash value of the user password. With secure hash function, our system can provide the security promise that the attacker can not recover the user's password even if the sensitive data in the server leaks. Thus the password our user is secured.

We have completed the password hash processing part in Java, however implementing hashed password involves several parts of system modules, including adding new account, modifying password and information, and log in part. As a result, there are still some work to be finished in the future.

```
//This function is for converting the raw string password to hashed string via SHA-256 hash function
String passwordToHash(String passwd) {
      MessageDigest md;
      try {
       md = MessageDigest.getInstance("SHA-256");
       md.update(passwd.getBytes());
       byte[] hash = md.digest();
       return byte2Str(hash);
      } catch (NoSuchAlgorithmException e) {
       // TODO Auto-generated catch block
       e.printStackTrace();
      }
      return new String("");
     }

     String byte2Str(byte[] hash) {
      StringBuilder sb = new StringBuilder();
      for(byte bt : hash) {
       //System.out.println((int)bt + " " + ((bt&0xf0)>>4) + " " + (bt&0xf));
       sb.append((char)('a'+((bt&0xf0)>>4)));
       sb.append((char)('a'+(bt&0xf)));
      }
      return sb.toString();
     }
```

Figure 31: Hashed Password

7 System Evaluation and Analysis

7.1 Test Case

In this section, we will show how to use our website by showing the process of how a patient goes to see doctor in hospital, as an evaluation of the whole system. All roles except administrator will be involved in this process. Considering that we have already discussed administrator's functionality in Section 5.2.1. We won't talk about it again.

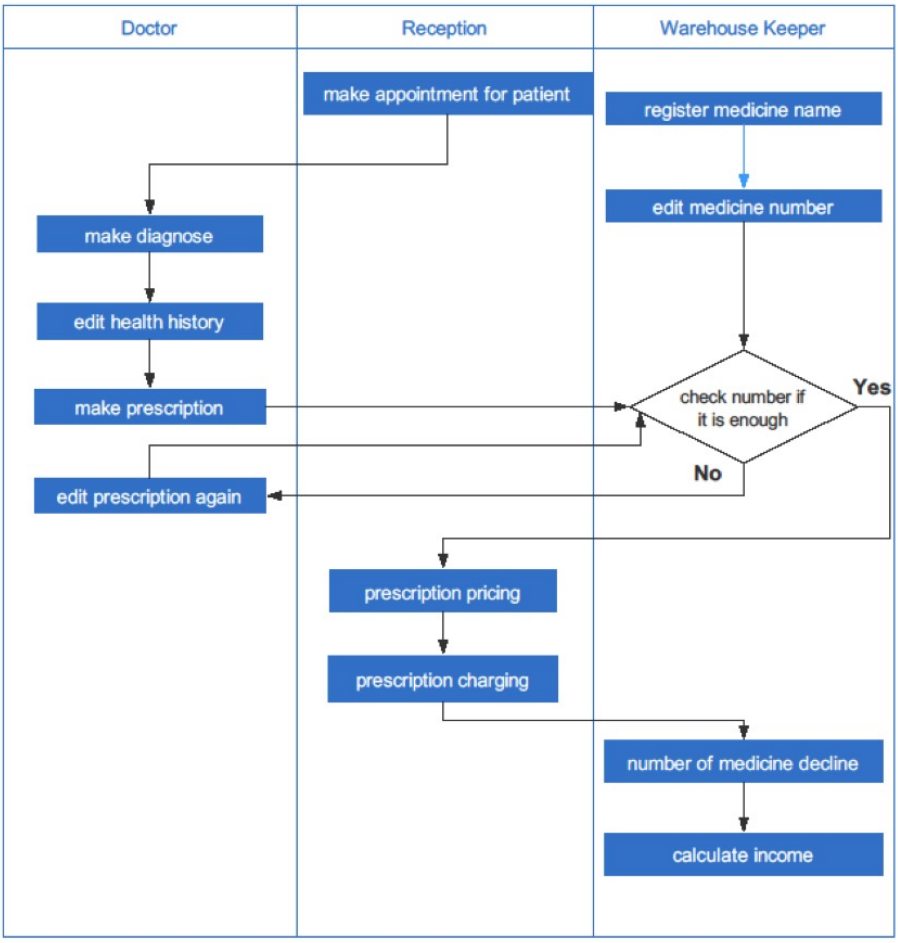


Figure 32: Process

1. When a patient enters the hospital, he first talks to a reception at front desk. And the reception will visit *patient making appointment* page and create a health history record.

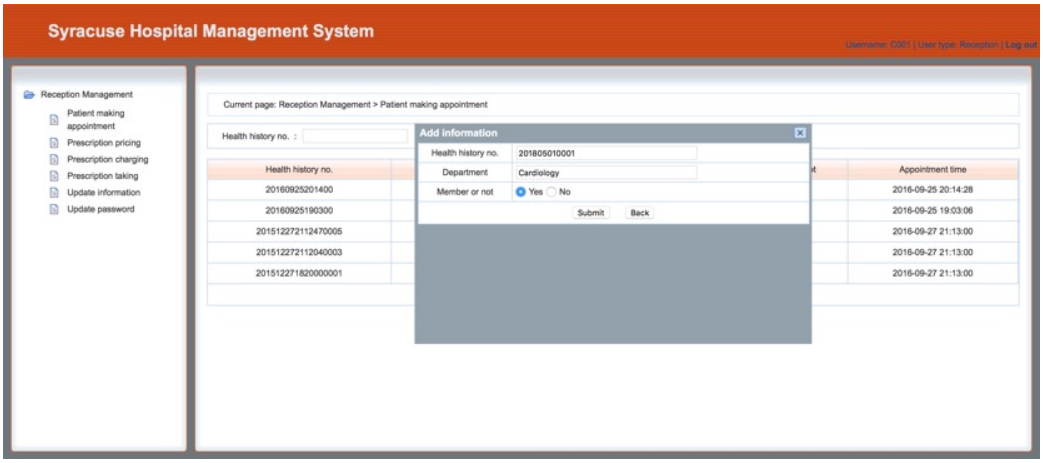


Figure 33: patient making appointment

2. Then the patient will visit the doctor. After the record is created, it will automatically appear in doctor's *Edit health history* Page, and doctor can edit it.

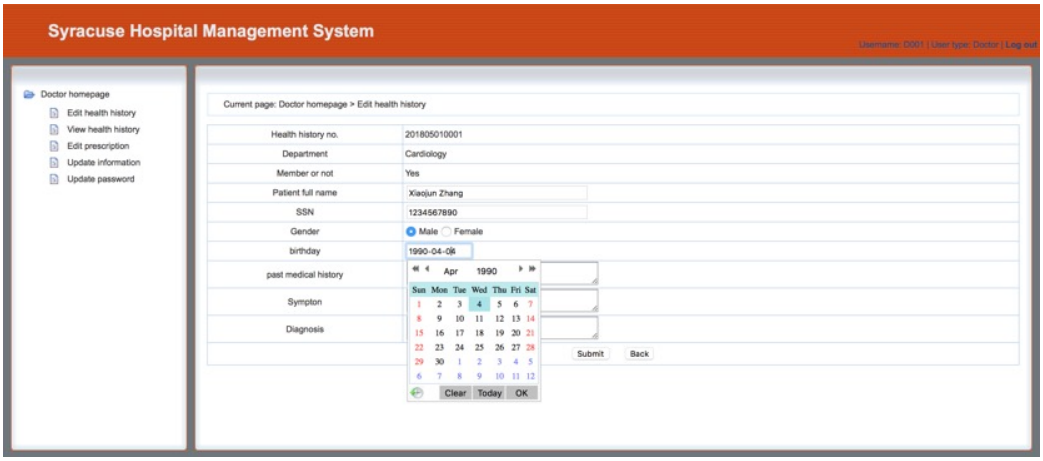


Figure 34: Edit health history

3. After editing the history, doctor can still review it in *View health history* page.

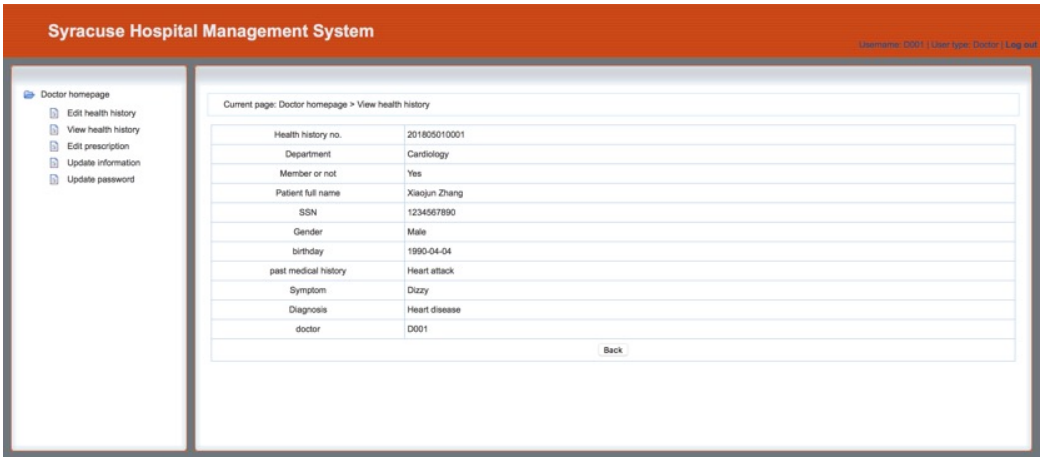


Figure 35: View health history

4. Then doctor will create a prescription for this health history. Mentioning that if doctor want to use multiple different kinds of medicines, he just have to add multiple times with certain quantities.

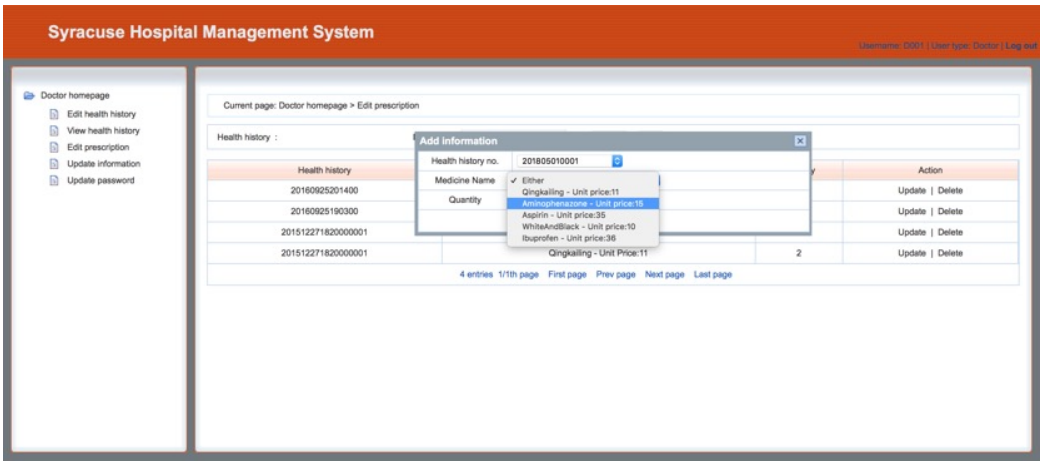


Figure 36: Edit prescription

5. Now the patient can go back to see the reception again. The reception will visit *Prescription pricing*, *Prescription charging* and *Prescription*

taking page one by one to help the patient to pay for and to fetch medicine.

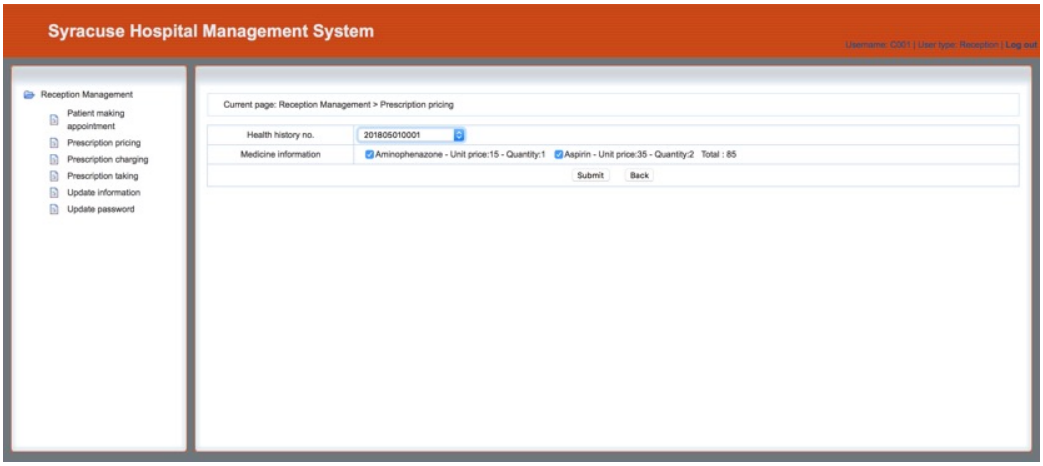


Figure 37: Prescription pricing

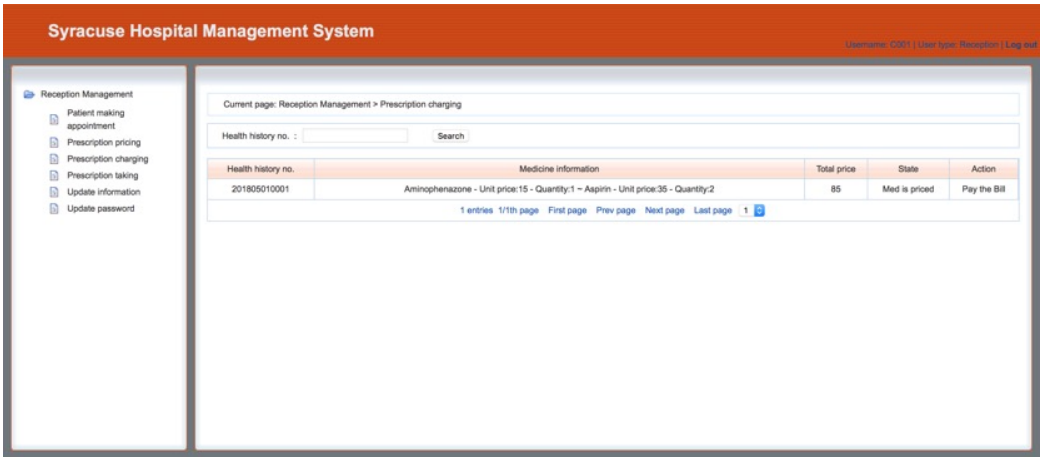


Figure 38: Prescription charging

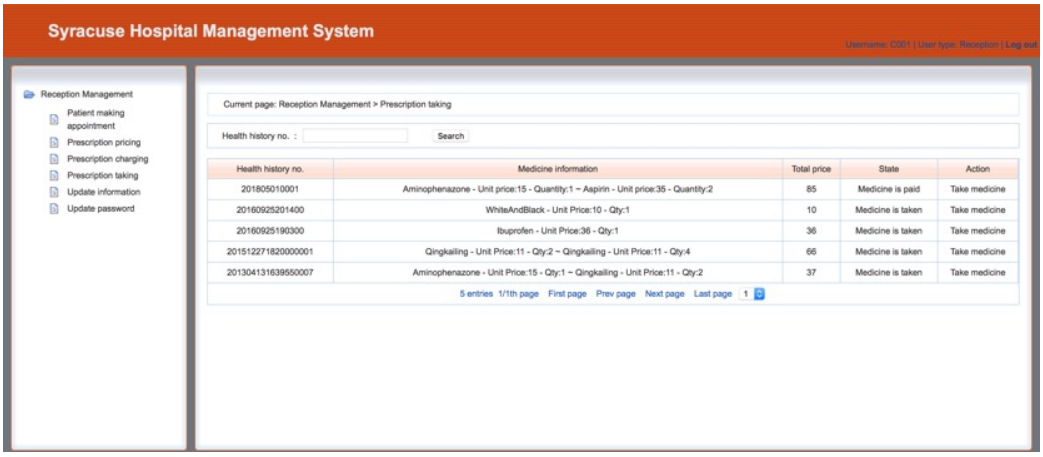


Figure 39: Prescription taking

6. We can check *Medicine summary* page before and after process 1 to 5. And we will see that the *current inventory* row is decreased after this process.

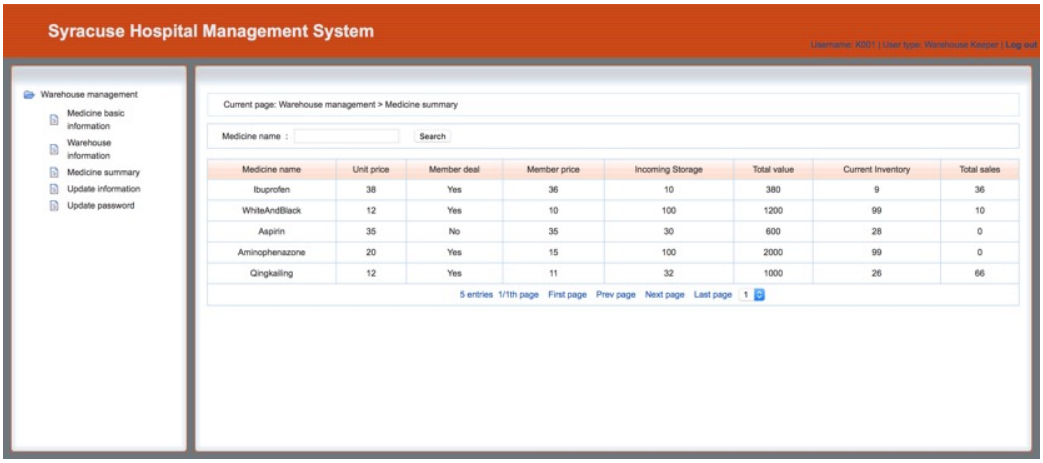


Figure 40: Medical Summary

7.2 Main functionality

7.2.1 Main functionality

This project focus on the information processing problems in real-time hospital scenarios, including the whole process from appointment to prescription. And our main goal is to achieve rapid response and high efficiency in hospital information management situations. The system mainly includes user management, medical record management, prescription management, warehouse management, log management, etc.

7.2.2 Details

- Based on the actual demand in hospital situation, we conclude the lightweight design of the hospital management system, which achieves low system requirements.
- we choose moderate open-source MySQL as our database management system and connect it with basic JDBC API, which provide with easy adoption and scalability support based on further demands.
- Roles defined in the system are clearly separated and the functionality of modules can meet the requirements of hospital scenarios. Our UI design is concise, clear and user-friendly, thus users can easily find the information and finish their operations without additional guidance.
- Thanks to the strong capability and cross-platform functionality of Java, our system can be implemented on various platforms, and support multiple devices with little effort.
- We implement several security countermeasures to eliminate the risk of various potential attacks, including SQL injection attack, CSRF and XSS attack. Therefore our system provide sufficient protection on sensitive information and user credentials in terms of confidentiality and integrity.

7.3 Features

7.3.1 Log Management

Event logs record events taking place in the execution of a system in order to provide an audit trail that can be used to understand the activity of the system, to diagnose problems and to recover from incorrect actions. In our hospital management system, we keep record of all employee actions, such as login, creating prescription and medicine into warehouse.

With this log table, the website is much more maintainable for administrator

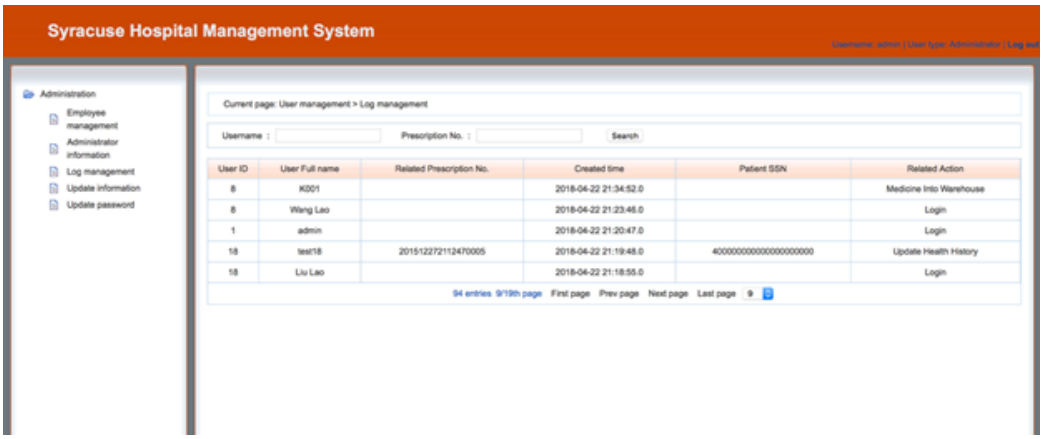


Figure 41: Log Management

7.3.2 Coherence and streamlined

It is our goal to design a highly efficient and practical health center management website, which means that different roles don't have to consider interaction with other roles and that our server will assist them to achieve interaction. Therefore, coherence is an import part of our design. There is coherence everywhere in our system.

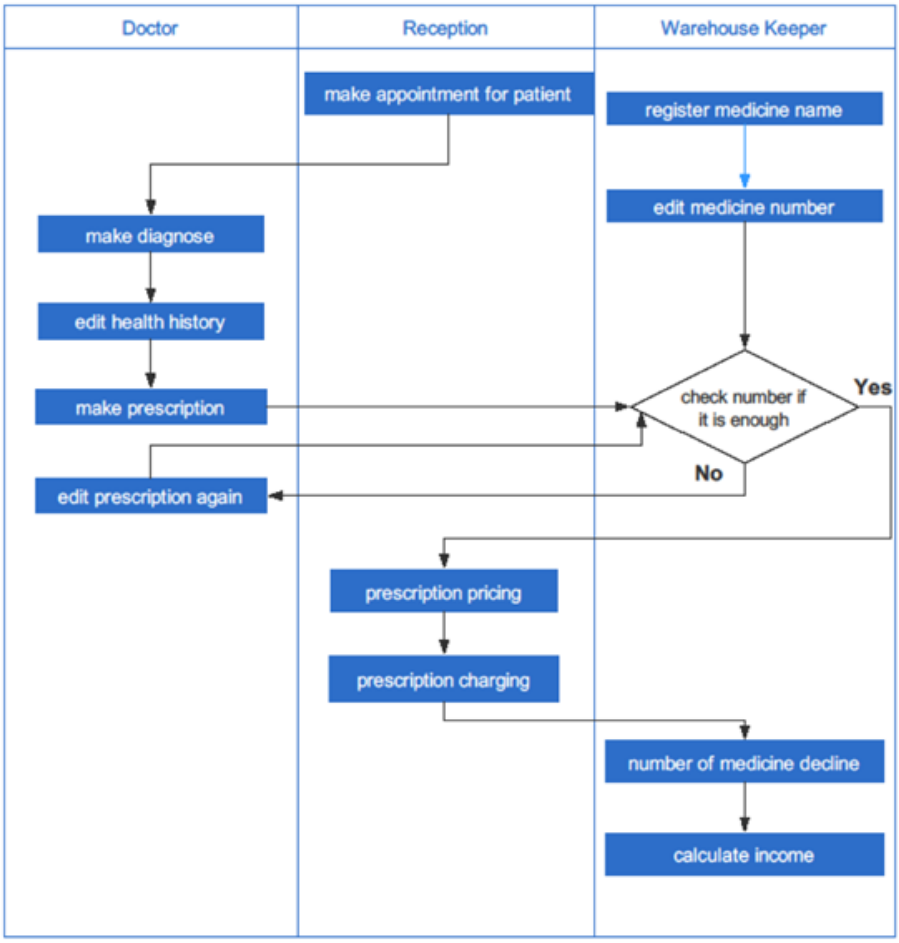


Figure 42: Coherence

1. Reception and Doctor

   After a reception makes an appointment for a patient, a fresh unedited health history record will automatically be created in doctors's *edit health history page*. After editing it, a doctor can create prescription for this history record. After that, a prescription will automatically appear in reception's *prescription pricing* page.

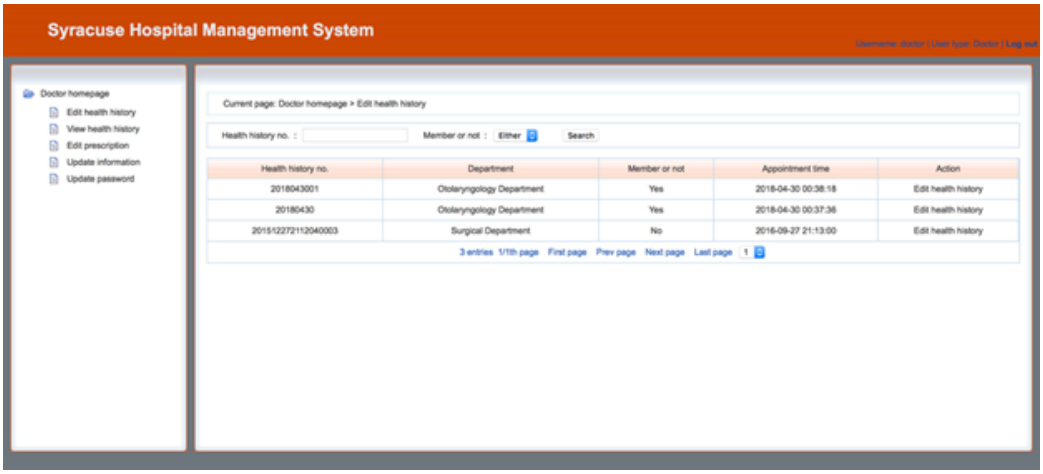


Figure 43: Edit health history record

2. Reception and Warehouse Keeper

   After a reception charges and takes medicine for a patient. This action is automatically record to our database, and the amount of available medicine in the warehouse is automatically decreased.

7.3.3 Detail oriented

To make our website as real and practical as possible, we focus on detail as best as we can.

- We make updating password logic and all other actions atomic and rigorous.
- We apply a lot of format constraints to simulate real personal information.

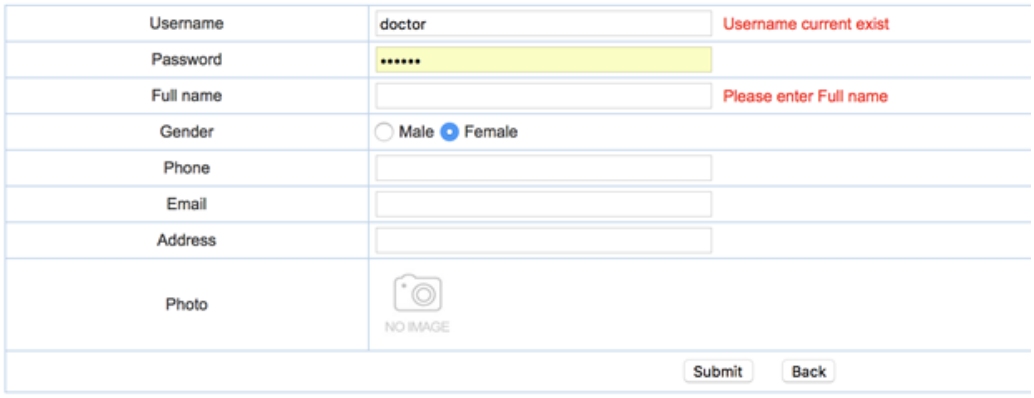


Figure 44: Constraint 1

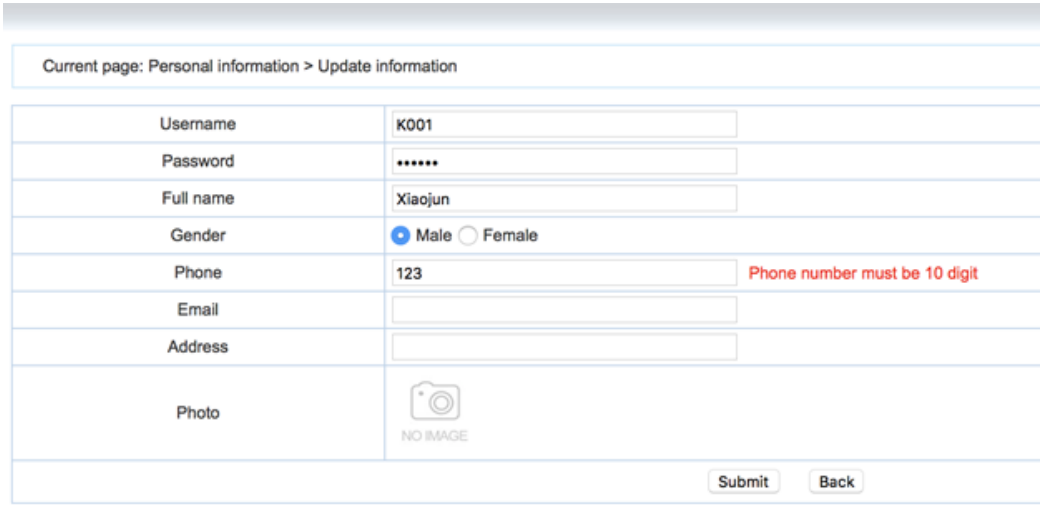


Figure 45: Constraint 2

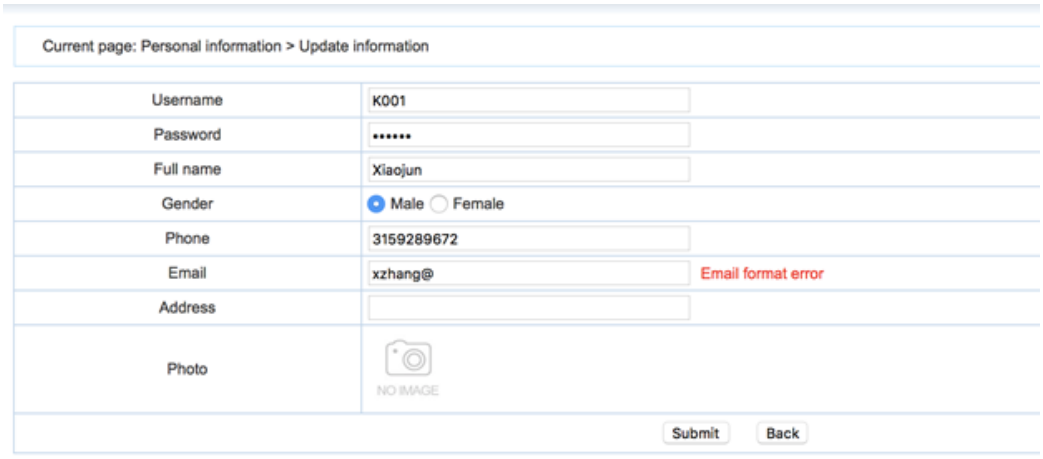


Figure 46: Constraint 3

8 Conclusion and Future Plan

8.1 Conclusion

After two months' effort of all our 8 members, we successfully build this health center, with both front-end and back-end side developed by ourselves. Our team mainly focus on how to make the website practical and secure.

To make the system practical, we put a lot of thoughts on how make a coherent and streamlined process and implementation details. For example, we did pattern validation, passed a fresh appointment to doctor's page, etc.

Considering that there are many sensitive information stored in hospital, we tried our best to make the website. We implemented a log table and managed to avoid many potential security problems. Consequently, the website is easy to maintain and difficult to attack.

Though we had thought of creating a role for patients, we decided not to include it after consideration. We designed this website for hospital internal use, and whatever patients want to do can be done with reception's assistance. Moreover, adding a patient role may greatly increase the attack surface of our website.

8.2 Future Plan

1. Complete password hashing.

   This is an important fail safe strategy. If only hash values of passwords of users are stored in our database, attackers will never know the actual passwords of users. We are about to finish it!

2. Improve health history management For now, all doctors can view all health history. We are planning to make doctors only able to view health histories related to him. The less privilege, the better! :)

Thanks to Dr. Yu and TAs!