

California State University, San Bernardino

CSUSB ScholarWorks

Theses Digitization Project

John M. Pfau Library

2011

My Clinic web application

Rohan Thomas DMello

Follow this and additional works at: <https://scholarworks.lib.csusb.edu/etd-project>



Part of the [Databases and Information Systems Commons](#)

Recommended Citation

DMello, Rohan Thomas, "My Clinic web application" (2011). *Theses Digitization Project*. 3958.
<https://scholarworks.lib.csusb.edu/etd-project/3958>

This Project is brought to you for free and open access by the John M. Pfau Library at CSUSB ScholarWorks. It has been accepted for inclusion in Theses Digitization Project by an authorized administrator of CSUSB ScholarWorks. For more information, please contact scholarworks@csusb.edu.

MY CLINIC WEB APPLICATION

A Project
Presented to the
Faculty of
California State University,
San Bernardino

In Partial Fulfillment
of the Requirements for the Degree
Master of Science
in
Computer Science

by
Rohan Thomas DMello
December 2011

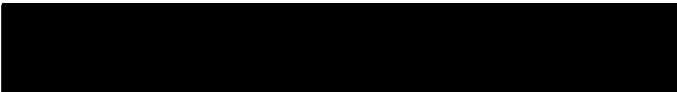
MY CLINIC WEB APPLICATION

A Project
Presented to the
Faculty of
California State University,
San Bernardino

by
Rohan Thomas DMello


December 2011

Approved by:


Dr. Richard Botting, Advisor, School
of Computer Science and Engineering

11/14/2011
Date


Dr. Ernesto Gomez


Dr. Kerstin Voigt

ABSTRACT

Web security and ease of use are very important factors when developing a web base application. I have developed such an application called 'My Clinic'. The purpose of this project is to facilitate doctors and patients in many ways. In this application doctors, patients and staff members have different functionalities that will be decided at the time of log in. It provides security to confidential information by showing only certain information to the logged in person based on the access permission. This Project is written using Java web technologies such as JSP and Servlets that adhered to J2EE specifications.

ACKNOWLEDGEMENTS

I would like to thank following people who helped me during this project. First and foremost, I would like to thank my project advisor Dr. Richard Botting, for his constant help, advice and tremendous support. He has always been very quick in his feedback about the implementation and design. I would also like to thank Dr. Voigt and Dr. Gomez who were in my committee, for their support. Dr. Mendoza, my graduate advisor, for her advice throughout the entire course. Special thanks to Ken Han for his help in setting up server machine. I would also like to thank Monica for being helpful and patient with me all the time for any requests made. I also would like to thank all the faculty members of Computer Science and Engineering Department for giving me this opportunity to pursue my Master's Degree. Last but not least, I would like to thank my family for their encouragement.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS	iv
LIST OF TABLES	viii
LIST OF FIGURES	ix
CHAPTER ONE: INTRODUCTION	
Background	1
Purpose	2
Scope	2
Significance of the Project	3
Products	3
Definitions, Acronyms, and Abbreviations	4
CHAPTER TWO: ARCHITECTURE	
Project Design	7
Model View Controller Architecture	7
System Security	9
Interface Design	10
System Interface	10
User Interface	12
Software Interfaces	12
Communication Interface	12
User Characteristic	13
System Flowchart	17
CHAPTER THREE: DATABASE DESIGN	

Data Analysis	18
Database Specification	18
Database Schema Logical Model - Relational Schema	20
Database Design	23
CHAPTER FOUR: PROJECT IMPLEMENTATION	
User Interface Design	32
Static Pages	33
Login Page	34
Search Pages	35
Edit Patient Page	38
Doctor Interface Pages	39
Staff Interface Pages	42
Patient Interface Pages	47
Pharmacy Interface Pages	50
CHAPTER FIVE: SYSTEM VALIDATION	
Unit Testing	52
System Integration Testing	57
System Testing	60
CHAPTER SIX: MAINTENANCE MANUAL	
Downloading and Installing CentOS	61
Downloading and Installing Java Development Kit	61
Downloading and Installing Tomcat	62

Downloading and Installing MySQL Database	62
Database Backup	63
System Backup	63
CHAPTER SEVEN: CONCLUSION AND FUTURE DIRECTIONS	
Conclusion	64
Future Directions	65
REFERENCES	66

LIST OF TABLES

Table 1. Database Schema Logical Model	20
Table 2. Structure of Table 'appointment'	23
Table 3. Structure of Table 'check_payments'	24
Table 4. Structure of Table 'doctor'	24
Table 5. Structure of Table 'employee'	25
Table 6. Structure of Table 'insurance'	26
Table 7. Structure of Table 'login'	27
Table 8. Structure of Table 'patient'	28
Table 9. Structure of Table 'payment_details'	29
Table 10. Structure of Table 'pharmacy'	29
Table 11. Structure of Table 'prescription'	30
Table 12. Structure of Table 'prescription_details'	31
Table 13. Unit Testing Results Table	52
Table 14. System Integration Test Results Table	57
Table 15. System Test Results Table	60

LIST OF FIGURES

Figure 1. Model View Controller Architecture Using Request Response Model	8
Figure 2. Component Diagram	11
Figure 3. Use Case Diagram - Doctor	13
Figure 4. Use Case Diagram - Staff	14
Figure 5. Use Case Diagram - Patient	15
Figure 6. Use Case Diagram - Pharmacy	16
Figure 7. System Flowchart	17
Figure 8. Class Diagram	19
Figure 9. Home Page	33
Figure 10. Login Page	34
Figure 11. Search by Name Page	35
Figure 12. Search by Email Page	36
Figure 13. List of Patients Page	37
Figure 14. Edit Patient Page	38
Figure 15. Doctor Home Page	39
Figure 16. Write Prescription Page	40
Figure 17. Doctor's Result Page	41
Figure 18. Staff Result Page	42
Figure 19. Make Payments Page	43
Figure 20. Make Appointment Page	44
Figure 21. Add Pharmacy Page	45

Figure 22. Edit Patient Info Page	46
Figure 23. Search Prescription Page	47
Figure 24. Prescription List Page	48
Figure 25. View Prescription Page	49
Figure 26. Pharmacy Home Page	50
Figure 27. Pharmacy Action Page	51

CHAPTER ONE

INTRODUCTION

This chapter gives a brief description of the background, purpose and scope of the project.

Background

'My Clinic' is a web based application, which enables doctor to log in to the application and get some of the web services that are usually provided by Hospitals or Healthcare Organizations. This application has four different modules, namely Doctor, Patient, Staff and Pharmacy. All these modules have different access permissions and depending on their access permissions the information will be provided to that particular module. This web application will also serve as a general website, where visitors can surf around and look for information about doctors, services they provide and pharmacies on their board.

Today, Internet has become an integral part of every person's everyday life. Internet has transformed the world into a small virtual place, being such a huge network of interconnected networks. Computers, Smart Phones, Tablets, E-readers all this kind of devices are used to access internet. Growth of internet based application that is also

called as web based applications are at the risk of being attacked by hackers, ill minded people. So it becomes very important to build a highly secured application from being attacked by these people, who tries to get personal information of the users.

Purpose

The purpose of this web application is to provide doctor and patient to have one place where their data is stored and they can access thorough secured environment. It helps doctors to write prescription online and be viewed by patient at their ease. The staff of the clinic can do more work with this application like giving an appointment to a patient, checking payment details of the patient, editing patient information and so on. Patients also have similar functionality to make appointment. The Pharmacy module will be to reduce patient's time a lot, as a prescription can be viewed by a pharmacist online.

Scope

The scope of this project is to develop the web application that is scalable over the geographic region and secure at the same time. It allows doctors and staff to add patient's information and access them later. Patients can also see all the medical records from past years. Patients

also have the option from which pharmacy to choose online and can use in built email feature to send an email to a pharmacy or clinic. The clinic that will be using this application will save a lot of time and energy as it is web based and less manual work and storage. Since application is compliant with ADA rules it can be accessible to the patients with disabilities.

Significance of the Project

This project is going to provide a small clinic a range of features that are usually used in Hospitals and other Health care organizations. Among the products on the market that integrates Doctor, Patient and Pharmacy for a small clinic, this product is one of the most efficient products.

Products

- My Clinic: A website/web application that is represented to the users using JSP, controlled using JAVA and served using MySQL database.
- Software Manual: A systematic documentation on how the software was implemented and should be maintained in the future.

Definitions, Acronyms, and Abbreviations

The following terms and definitions are used the project.

SDK - A Software Development Kit is typically a set of development tools that allows for the creation of applications for certain software package, software framework, hardware platform, gaming console or similar platform. [1]

Java SDK - is a SDK by Sun Microsystems, owned by Oracle Corporation, and is aimed at Java Developers. Java is a object oriented programming language. [2]

Eclipse - Eclipse is a multi-language software development environment comprising an integrated development environment (IDE) and an extensible plug-in system. Eclipse is free and open source. [3]

JSP - Java Server Pages is a Java technology that helps software developers serve dynamically generated web pages based on HTML, XML, or other document types. [4]

Java Servlet - A Servlet is a Java class in Java EE that conforms to the Java Servlet API, a protocol by which a Java class may respond to requests. They are not tied to a specific client-server protocol, but are most often used with the HTTP protocol. [5]

API - An application programming interface (API) is a particular set of rules ('code') and specifications that software programs can follow to communicate with each other. It serves as an interface between different software programs and facilitates their interaction, similar to the way the user interface facilitates interaction between humans and computers. [6]

HTTP - The Hypertext Transfer Protocol (HTTP) is a networking protocol for distributed, collaborative, hypermedia information systems. HTTP is the foundation of data communication for the World Wide Web. [7]

UML - Unified Modeling Language (UML) is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created, by the Object Management Group. [8]

XML - Extensible Markup Language (XML) is a set of rules for encoding documents in machine-readable form. It is defined in the XML 1.0 Specification produced by the W3C, and several other related specifications, all gratis open standards. [9]

Dia - is free and open source general-purpose diagramming software, developed originally by Alexander Larsson. [10]

HTML - Hypertext Markup Language (HTML) is the predominant markup language for web pages. HTML elements are the basic building-blocks of web pages. [11]

CSS - Cascading Style Sheets (CSS) is a style sheet language used to describe the presentation semantics (the look and formatting) of a document written in a markup language. [12]

JavaScript - JavaScript is a prototype-based scripting language that is dynamic, weakly typed and has first-class functions. It is a multi-paradigm language, supporting object-oriented, imperative, and functional programming styles. [13]

CHAPTER TWO

ARCHITECTURE

Project Design

'My Clinic' implements a web based client-server system. The front end is implemented using JSP. At the back end, Java is used for server side needs and MySQL for the database needs. J2EE Framework is used to maintain the lifecycle of the pages being served to the web browser. All this integration of serving pages and connecting to the database is done using Apache Tomcat web server.

Model View Controller Architecture

Model-View-Controller (MVC) is a software architecture, currently considered an architectural pattern used in software engineering. [23]

The MVC architecture can be can be divided into 3 different categories:

- Model - handles the data and business logic a java class can be considered as model
- Controller - handles the User Interface and application logic. A servlet is an example of controller.

- View - handles the presentation logic and provides the data to the graphical user interface objects.

The following figure depicts the flow of MVC architecture.

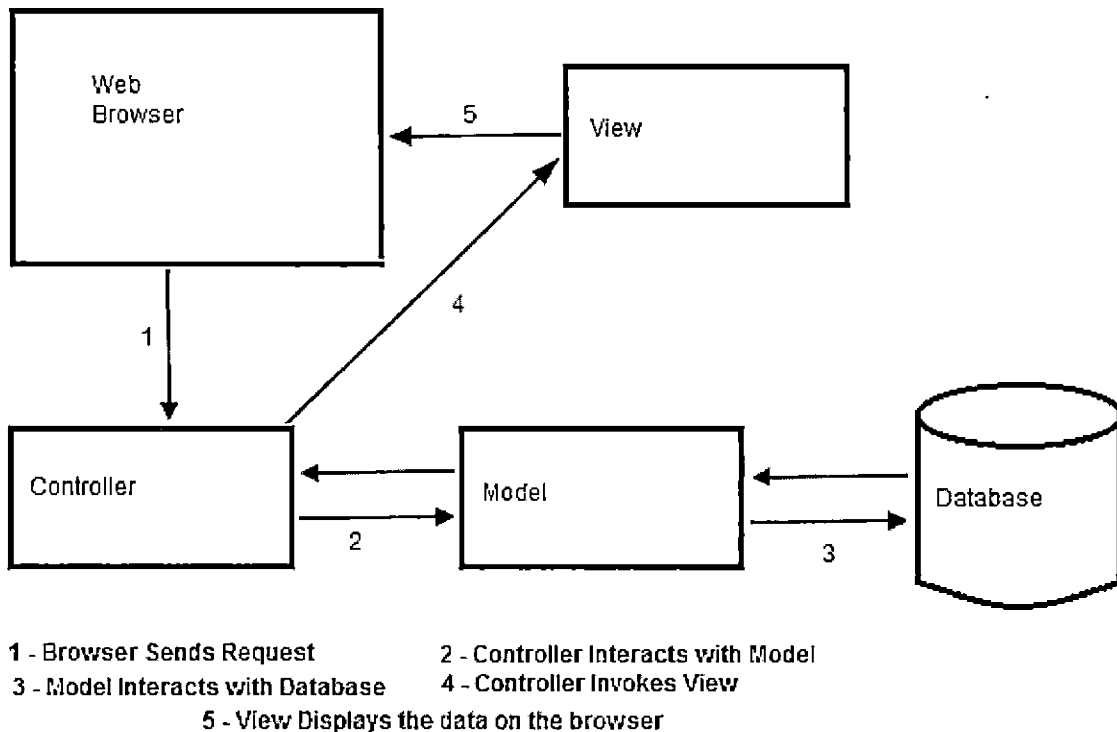


Figure 1. Model View Controller Architecture Using Request Response Model

This categorization of the architecture for the user request can be broken up as follows:

1. The web browser sends a request to the controller for a page.
2. The controller interacts with the model.

3. The model then interacts with database, if any call to database, and sends the data back to controller.
4. Controller invokes the view with data and information received from the model.
5. View then renders the page on the web browser.

System Security

In a system that is web based, security is very important quality. Since most of the web pages in the application takes the input data from the user. People with bad intentions try to attack the database by sending data that are interpreted as SQL commands. This is called 'SQL injection'. Java has an answer for this, which is called 'PreparedStatement' and parameterized SQL. This is used in combination with java beans. So that it ensures that no null values are passed to the sql query. A simple example of the SQL Injection is

```
SELECT * FROM users WHERE name = '' OR '1'='1';
```

This holds the principal of the Object Oriented Programming language feature of Encapsulation. Java bean classes are used to encapsulate many objects into one bean object and passed around as a single bean object.

The security aspect is very important when it comes to password. There are so many hackers out there who ready to

sneak into the communication line and read your password. So in this application the password is stored in the database using 64 bit encryption. It not just uses the encryption for the password but it uses an extra field that is used along with the password to uniquely identify the similar passwords. When user is created for the first time the password's hash value is calculated and stored in database.

Interface Design

This section elaborates about the different interfaces used in the application.

System Interface

As 'My Clinic' is MVC based architecture that is database driven, there are following three components:

1. A client machine with web browser.
2. A web server such as Apache Tomcat used in this project.
3. A database server that is MySQL Database in this project.

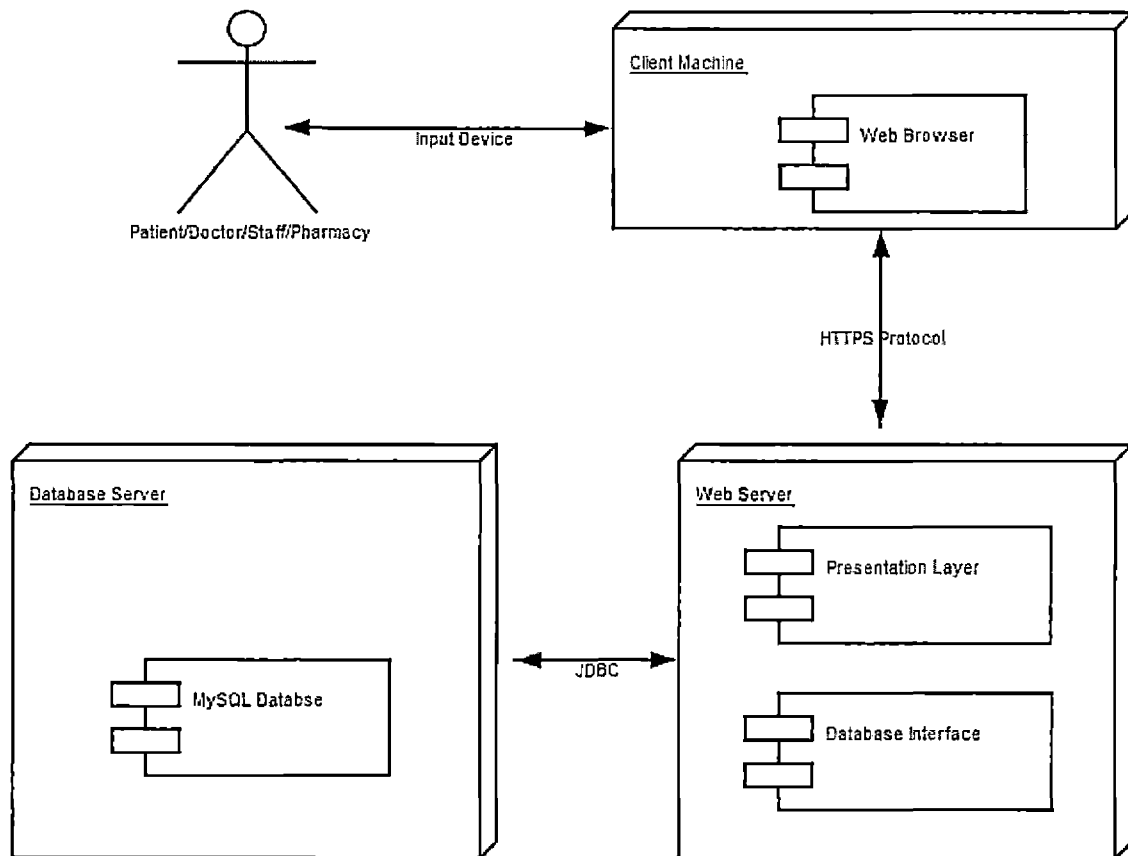


Figure 2. Component Diagram

When a user accesses the system, the machine sends a request to the web server. Machine is essentially a web browser, sends a request using HTTPS protocol to the web server. Web server then establishes connection with database server through database interface. The interface is JDBC interface. All steps are performed for most of the communication between client and server.

User Interface

'My Clinic' user interface is categorized into four different modules as follows:

1. Doctor Module Interface
2. Staff Module Interface
3. Patient Module Interface
4. Pharmacy Module Interface

All interfaces include login interface page. After login page the user is validated. Depending on the role and access permission the information displayed on particular interface.

Software Interfaces

'My Clinic' software interface is viewed over the internet or intranet through web browser. It can run on any web browser for Windows, Mac or Linux. The language used to implement this project is JAVA, JSP, HTML, CSS and JavaScript. The web server is running on CentOS 5.

Communication Interface

'My Clinic' uses J2EE framework to communicate between web browser and web server, according to the J2EE specification. The JDBC handles the communication between Java and MySQL.

User Characteristic

This section explains all the modules with use case for each module. 'My Clinic' system allows doctors a certain number of features. The features are shown in Figure 2.3. A doctor can create a new patient, delete a patient record, edit its information and most importantly write a prescription for the patient. The use case diagram is shown below:

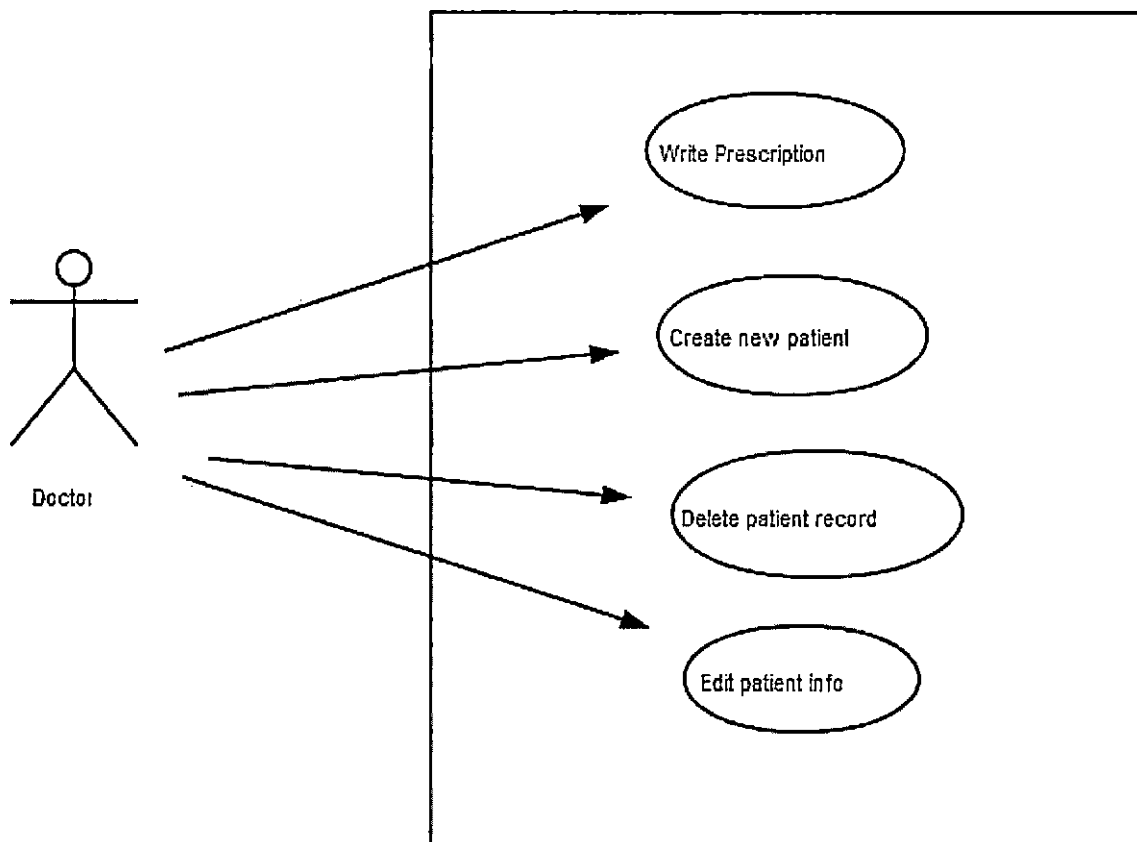


Figure 3. Use Case Diagram - Doctor

Staff members in 'My Clinic' system can perform more responsibilities than any other user in the system. Apart from adding patient info and all the actions that are performed by doctor except writing prescription, staff members can make payment, make appointment, send email and add pharmacy. The use case diagram for Staff is shown here.

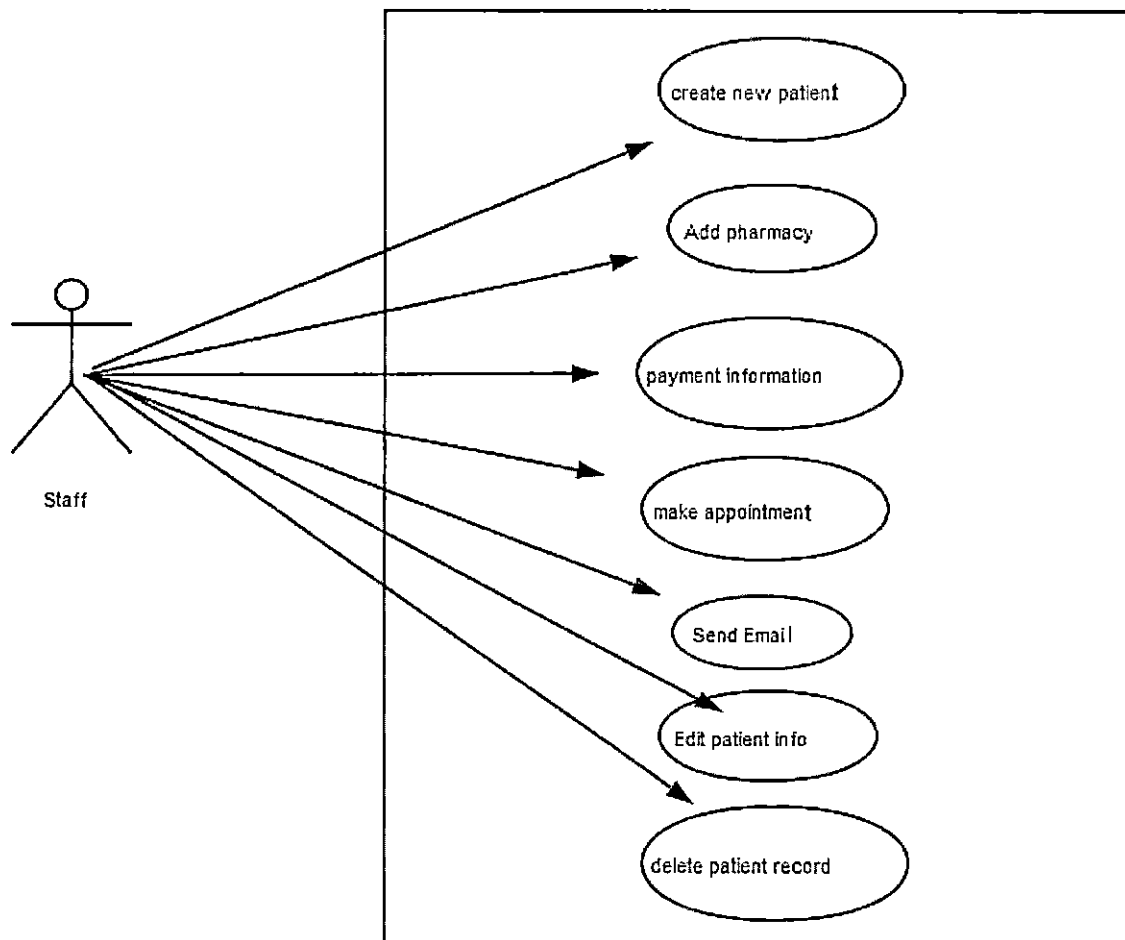


Figure 4. Use Case Diagram - Staff

Another important user of this system is Patient. A patient can make appointment from this application. Patient can also edit information that will be updated in the database. Patient can see the prescription at their ease when they want to refer to it later in the time. Patient can also send email and search for the pharmacies that are on the clinic's panel.

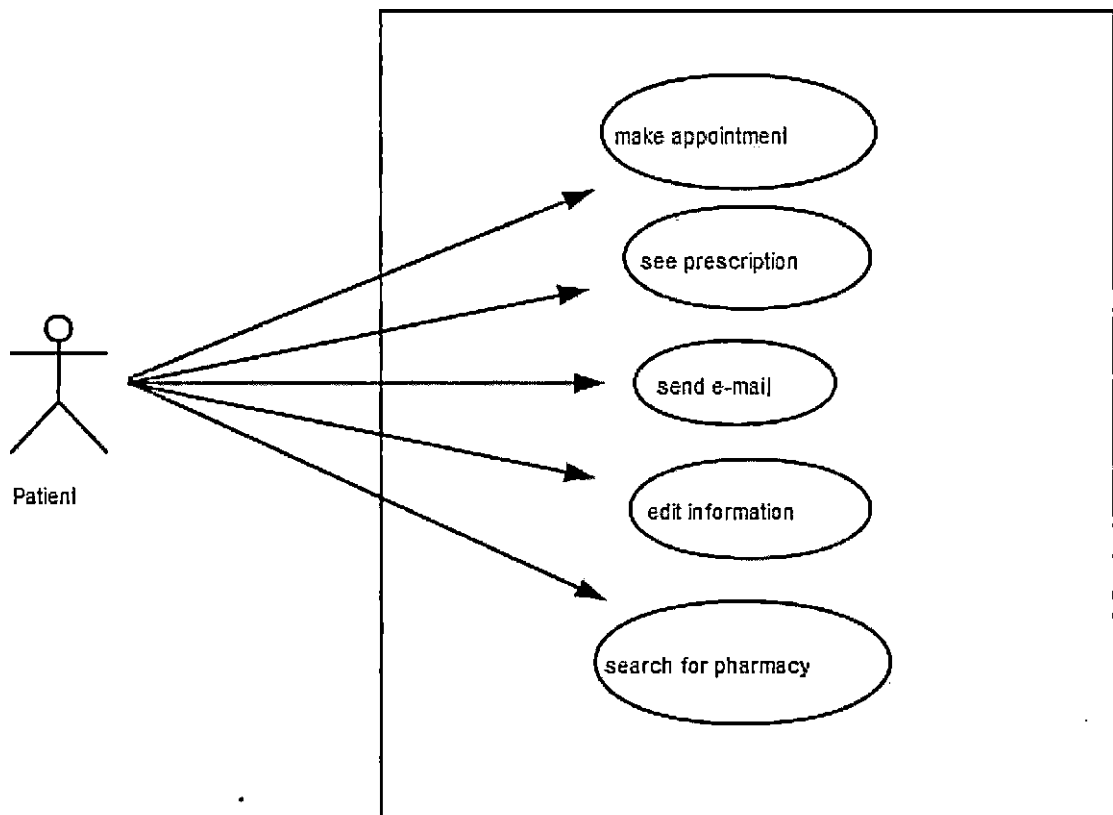


Figure 5. Use Case Diagram - Patient

One more user of this system is Pharmacy. This user is added to make things easier on patient end. Pharmacy can login and see the prescription of the patient, and send an email accordingly to the patient about the status of the prescription. It updates the database for that particular patient if a prescription assigned has a refill option.

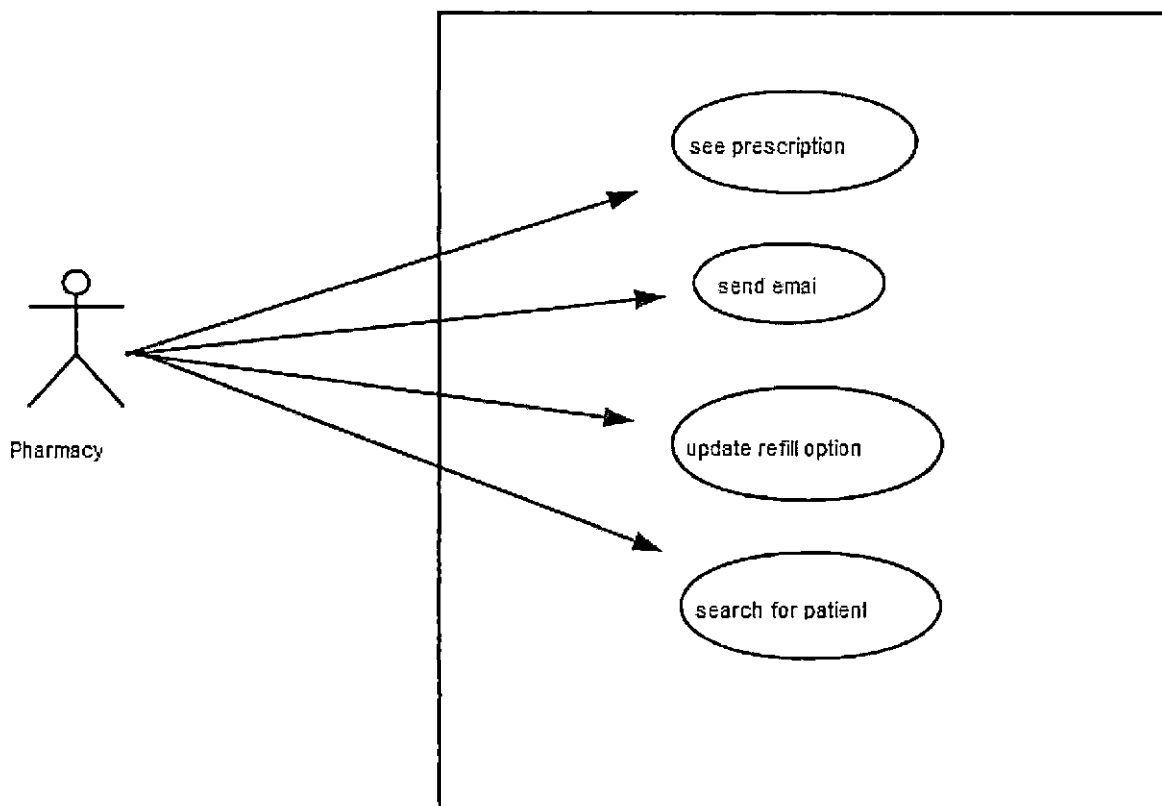


Figure 6. Use Case Diagram - Pharmacy

System Flowchart

A systematic flowchart of a system is shown in the following diagram. It depicts the flow of the system after login, then how the access is passed on and information is displayed.

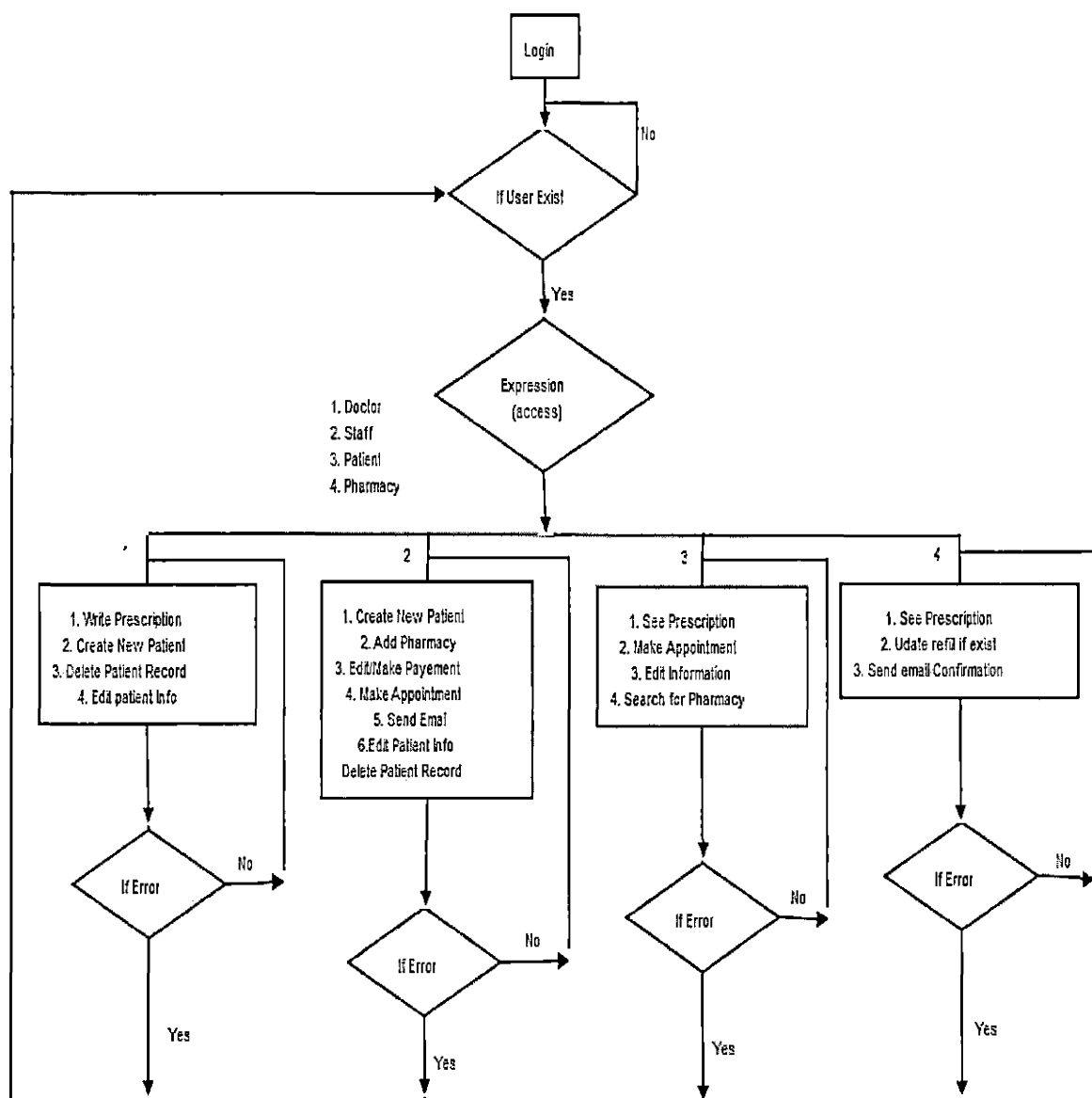


Figure 7. System Flowchart

CHAPTER THREE

DATABASE DESIGN

Data Analysis

'My Clinic' is based on the J2EE specification that implements the database solution using JDBC Driver. JDBC acts as a bridge between MySQL server and JAVA application, and JSP pages. This enhances the ability of java application to connect to the database. When the application interacts with the database the connection remains open until the request is served. At the time of updating the database it remains open and once the request is closed the connection is closed. This is useful to conserve the resources and for security of the data.

Database Specification

In designing schema for the database, two distinct parts have been identified for 'My Clinic' application. First one is the conceptual model, showing the relation between the entities. Second is the logical model that includes structure and the constraints of the data.

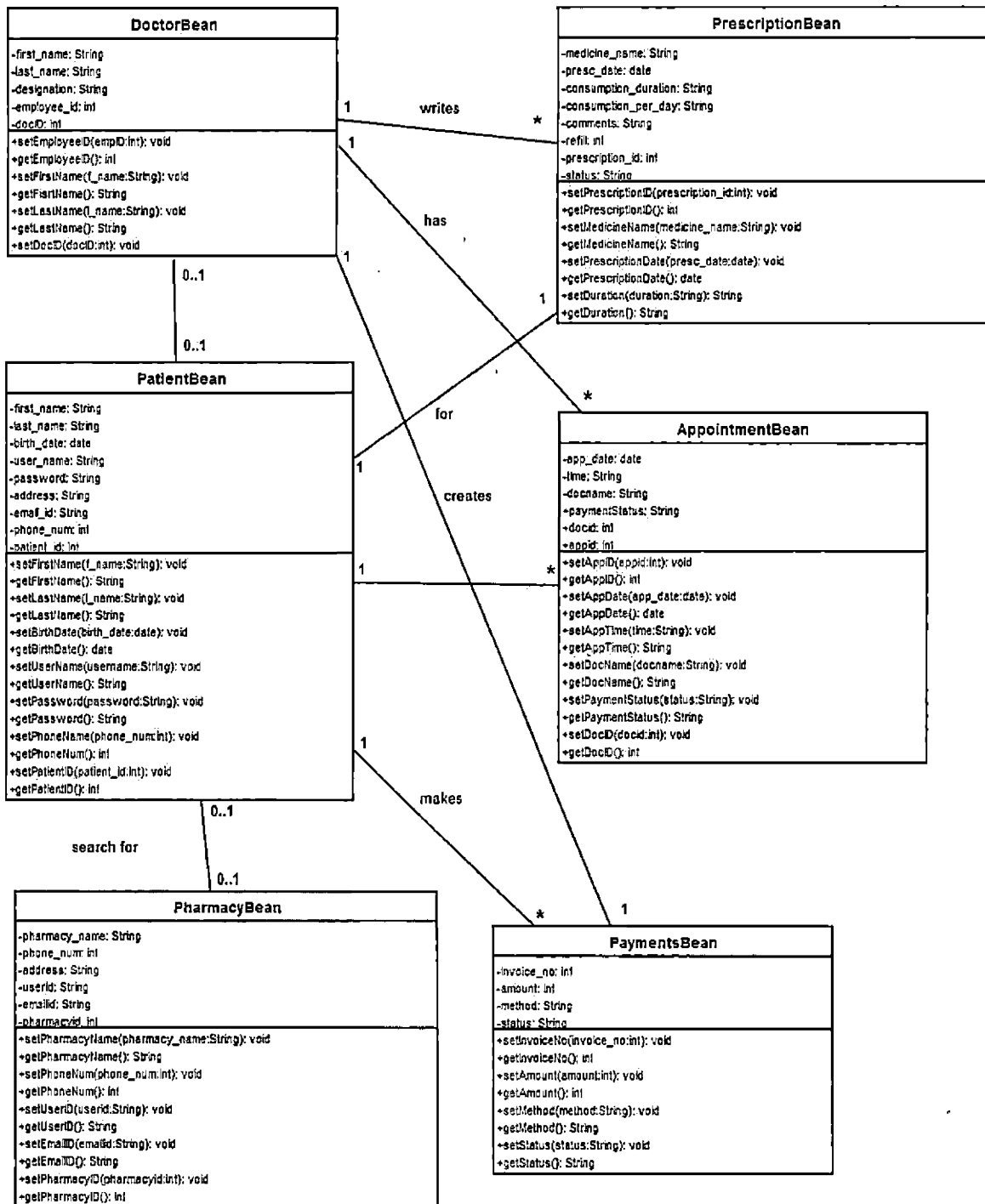


Figure 8. Class Diagram

Database Schema Logical Model - Relational Schema

The database relational schema is shown in the following table. The primary key is underlined in each table.

Table 1. Database Schema Logical Model

Appointment

<u>appointment id</u>	date	time	pay_status	fk_patientID
fk_docID				

Check Payment

<u>Check num</u>	bank_name	fk_invoiceno		
------------------	-----------	--------------	--	--

Doctor

<u>Doc id</u>	field	fk_userID		
---------------	-------	-----------	--	--

Employee

<u>Employee id</u>	first_name	last_name	address	designation
fk_userid				

Insurance

<u>Insurance id</u>	company_name	policy_number	co_pay	fk_invoice
				no

Login

<u>User ID</u>	password	extra	access_code	
----------------	----------	-------	-------------	--

Patient

<u>Patient id</u>	first_name	last_name	birth_date	address
phone_num	email_id	fk_userid		

Payment Details

<u>Invoice no</u>	amount	method	status	fk_appintmentid
-------------------	--------	--------	--------	-----------------

Pharmacy

<u>Pharmacy id</u>	pharmacy_name	address	phone_num	email_id
fk_userid				

Prescription

<u>Prescription id</u>	fk_patientid	fk_docid		
------------------------	--------------	----------	--	--

Prescription Details

<u>Prescription</u> <u>date</u>	medicine_ name	consumption_ _duration	consumption_ per_day	comments
refill	fk_prescr iptionid			

Database Design

'My Clinic' application stores the information about the patient. The Logical model establishes following detailed design in MySQL database server. The following tables describe field, data type, length, primary key, null or non-null keys and description of the field.

Table 2. Structure of Table 'appointment'

Field	Type	Key	Null	Description
Appointment_id	Int	Primary	No	Unique ID for appointment
Date	Date		No	Appointment date
Time	Time		No	Appointment time
Pay_status	Varchar		No	Whether or not patient has paid for this appointment or not
Fk_patientid	Int	Foreign	No	Patient id for which the appointment is made
Fk_docid	Int	Foreign	No	Doctor with whom the appointment is scheduled

Table 3. Structure of Table 'check_payments'

Field	Type	Key	Null	Description
Check_number	Int	Primary	No	Check number
Bank_name	Varchar		No	Name of the Bank
Fk_invoiceno	Int	Foreign	No	Invoice number to which this check is associated with

Table 4. Structure of Table 'doctor'

Field	Type	Key	Null	Description
doc_id	Int	Primary	No	Doctor id to get doctor's information
Field	Varchar		No	Doctor's specialization in the particular field
Fk_userid	Varchar	Foreign	No	User id to identify the doctor.

Table 5. Structure of Table 'employee'

Field	Type	Key	Null	Description
employee_id	Int	Primary	No	Unique ID for a particular employee i.e. staff or doctor
First_name	Varchar		No	First Name of the employee
Last_name	Varchar		No	Last name of the employee
Address	Varchar		No	Address of the employee
Designation	Varchar		No	Whether an employee is doctor, staff or receptionist.
Fk_userid	Varchar	Foreign	No	User id to identify the user

Table 6. Structure of Table 'insurance'

Field	Type	Key	Null	Description
insurance_id	Int	Primary	No	Unique ID for insurance
Company_name	Varchar		No	Name of the provider
Policy_number	Varchar		No	Policy number
Co_pay	Float		No	Co pay that patient has to pay
Fk_invoiceno	Int	Foreign	No	Invoice number to identify which patient and what appointment

Table 7. Structure of Table 'login'

Field	Type	Key	Null	Description
user_id	Varchar	Primary	No	Unique ID for a user of the system
Password	Varchar		No	A password for the user
Extra	Varchar		No	An extra field to uniquely identify the password, since passwords can be same
Access_code	Int		No	Access code to identify which user is logged in

Table 8. Structure of Table 'patient'

Field	Type	Key	Null	Description
patient_id	Int	Primary	No	Unique ID for a patient
First_name	Varchar		No	First Name of the patient
Last_name	Varchar		No	Last name of the patient
Address	Varchar		No	Address of the patient
Birth_date	Date		No	Birth date
Phone_num	Int		No	Phone number
Email_id	Varchar		No	Email id
Fk_userid	Varchar	Foreign	No	User id to identify a patient

Table 9. Structure of Table 'payment_details'

Field	Type	Key	Null	Description
patient_id	Int	Primary	No	Unique ID for a patient
First_name	Varchar		No	First Name of the patient
Last_name	Varchar		No	Last name of the patient
Address	Varchar		No	Address of the patient
Birth_date	Date		No	Birth date
Phone_num	Int		No	Phone number
Email_id	Varchar		No	Email id
Fk_userid	Varchar	Foreign	No	User id of a patient

Table 10. Structure of Table 'pharmacy'

Field	Type	Key	Null	Description
pharmacy_id	Int	Primary	No	Pharmacy ID
Pharmacy_name	Varchar		No	Name of the pharmacy
Address	Varchar		No	Address of the patient
Phone_num	Int		No	Phone number
Email_id	Varchar		No	Email id
Fk_userid	Varchar	Foreign	No	User id to identify a pharmacy

Table 11. Structure of Table 'prescription'

Field	Type	Key	Null	Description
Prescription_id	Int	Primary	No	Prescription id
Fk_docid	Int	Foreign	No	Doctor id who wrote the prescription
Fk_patientid	Int	Foreign	No	Patient id to get the prescription for this patient

Table 12. Structure of Table 'prescription_details'

Field	Type	Key	Null	Description
Prescription_date	Date		No	Prescription date
Medicine_name	Varchar		No	Medicine name
Consumption_duration	Varchar		No	Consumption duration
Consumption_per_day	Varchar		No	Consumption per day
Comments	Varchar		No	Any additional comments
Refill	Int		No	Refill for this prescription
Fk_prescriptionid	Int	Foreign	No	Prescription id to get the information

CHAPTER FOUR

PROJECT IMPLEMENTATION

User Interface Design

'My Clinic' is a purely web based application. Therefore the user interface is designed in such a way that it supports all browsers. Such as Microsoft Internet Explorer, Mozilla Firefox, Opera Browser. All the interfaces are generated dynamically at the server side by JSP based on the access level of the user.

Static Pages

These pages make the web site for the general audience, whom are not the part of this application.

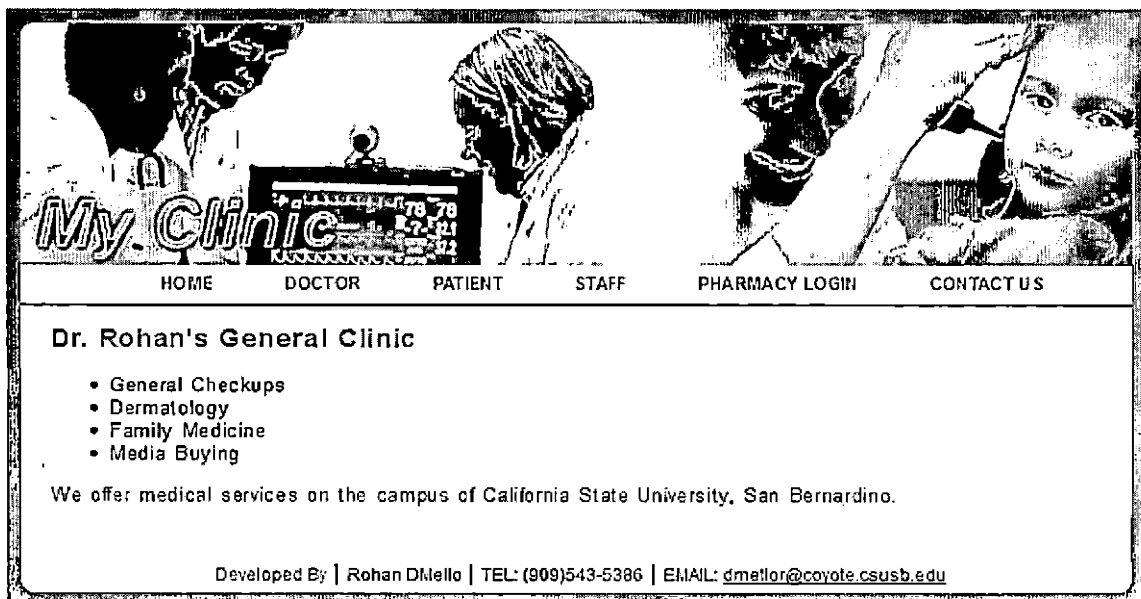
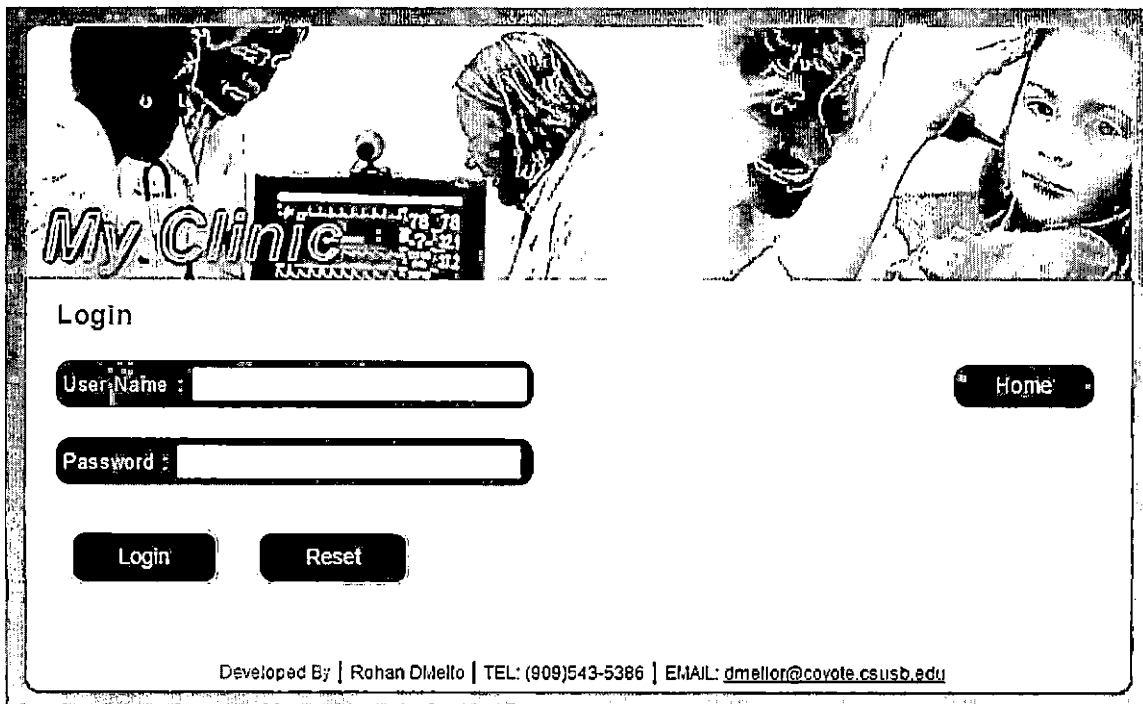


Figure 9. Home Page

These users and go around the website and see what services are being offered by clinic, whom to contact and so on. The above figure shows the common user interface that will be used in all other subsequent page.

Login Page

The login page is main page that will be used by all the users of the system.



My Clinic 78 78 2 324 312

Login

User Name :

Home

Password :

Login Reset

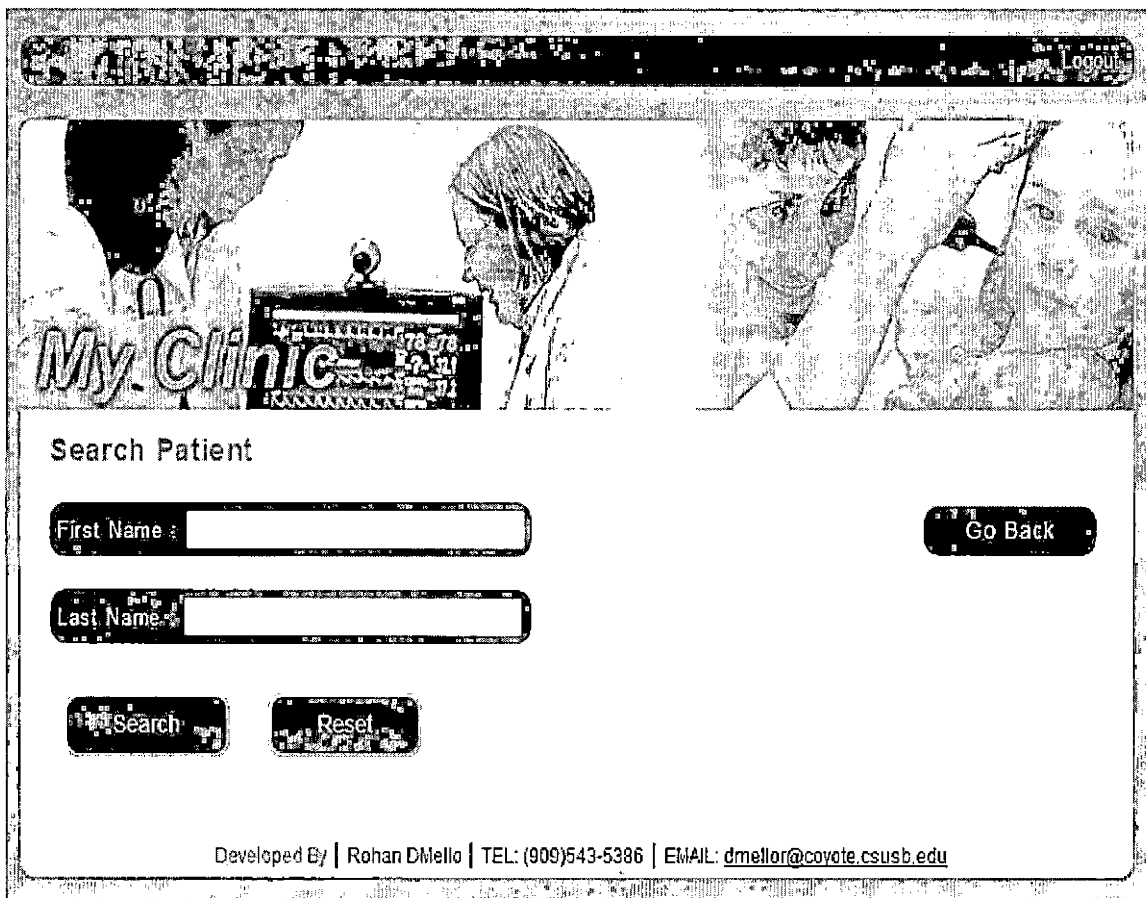
Developed By | Rohan DMello | TEL: (909)543-5386 | EMAIL: dmello@covote.csusb.edu

Figure 10. Login Page

This page is same for all users and can be accessed from any users namely doctor, patient, staff and pharmacy. The figure above shows the interface for login page.

Search Pages

Search by Name Page: The search page by name is used to search patient using their first name and last name or combination of both. The figure below shows the search page.



The screenshot shows a web browser window with a header banner for 'My Clinic' featuring three medical professionals. Below the banner is a 'Search Patient' section with two text input fields labeled 'First Name' and 'Last Name'. To the right of the 'First Name' field is a 'Go Back' button. Below the input fields are 'Search' and 'Reset' buttons. At the bottom of the page, a footer contains the text: 'Developed By | Rohan DMello | TEL: (909)543-5386 | EMAIL: dmello@covate.csusb.edu'.

Figure 11. Search by Name Page

Search by Email Page: This page is used to search the patient by their email id. Select the email-id form the drop down list. After selecting it will show the information of the patient and the task that can be performed on will be displayed.

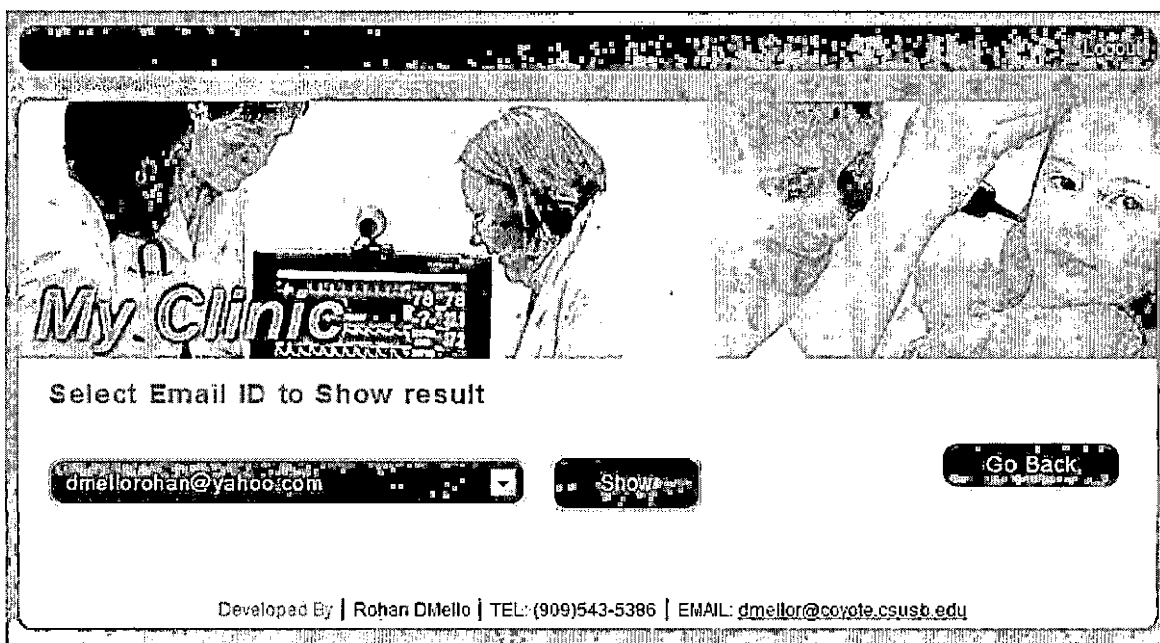


Figure 12. Search by Email Page

List of Patients Page: This page shows the entire list of patients and after selecting any of the patients, it will take you to the result page of the particular module.

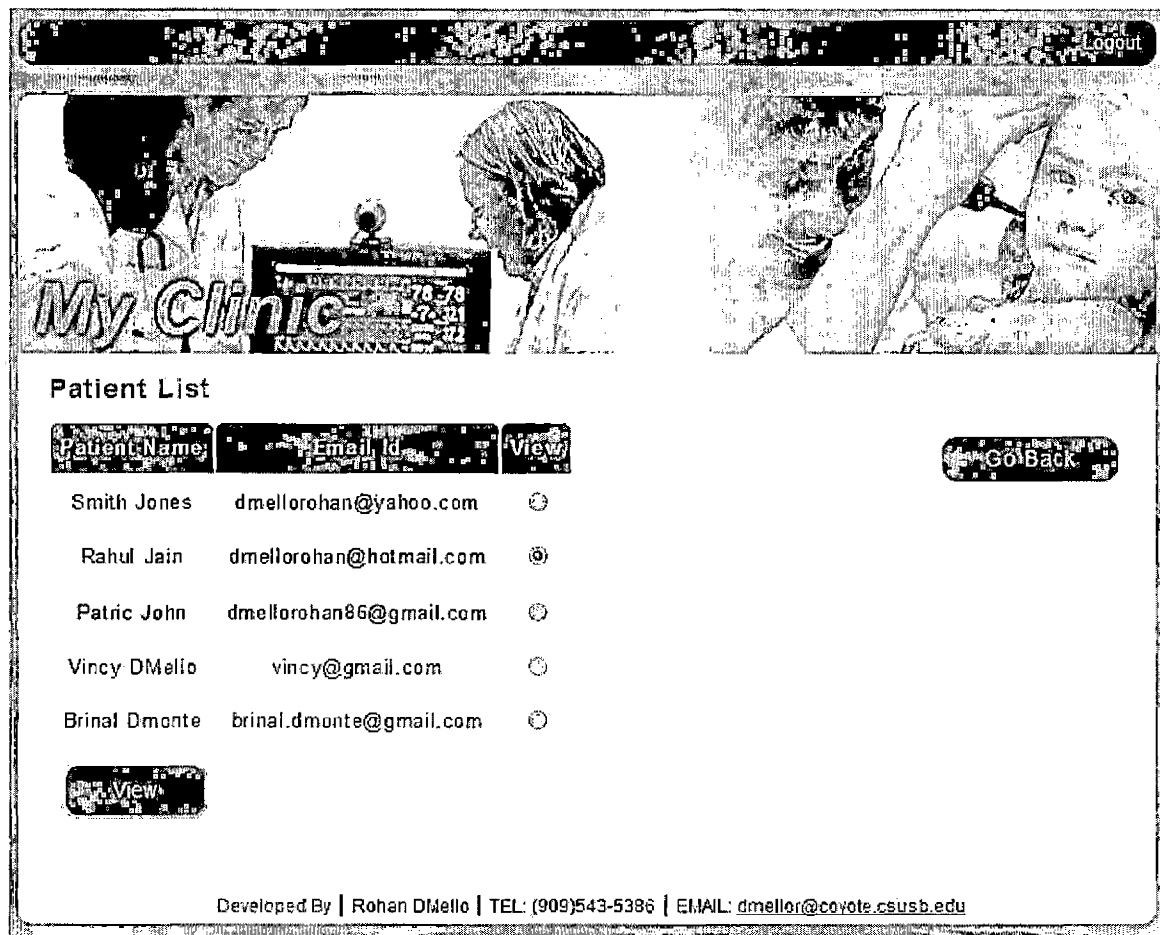
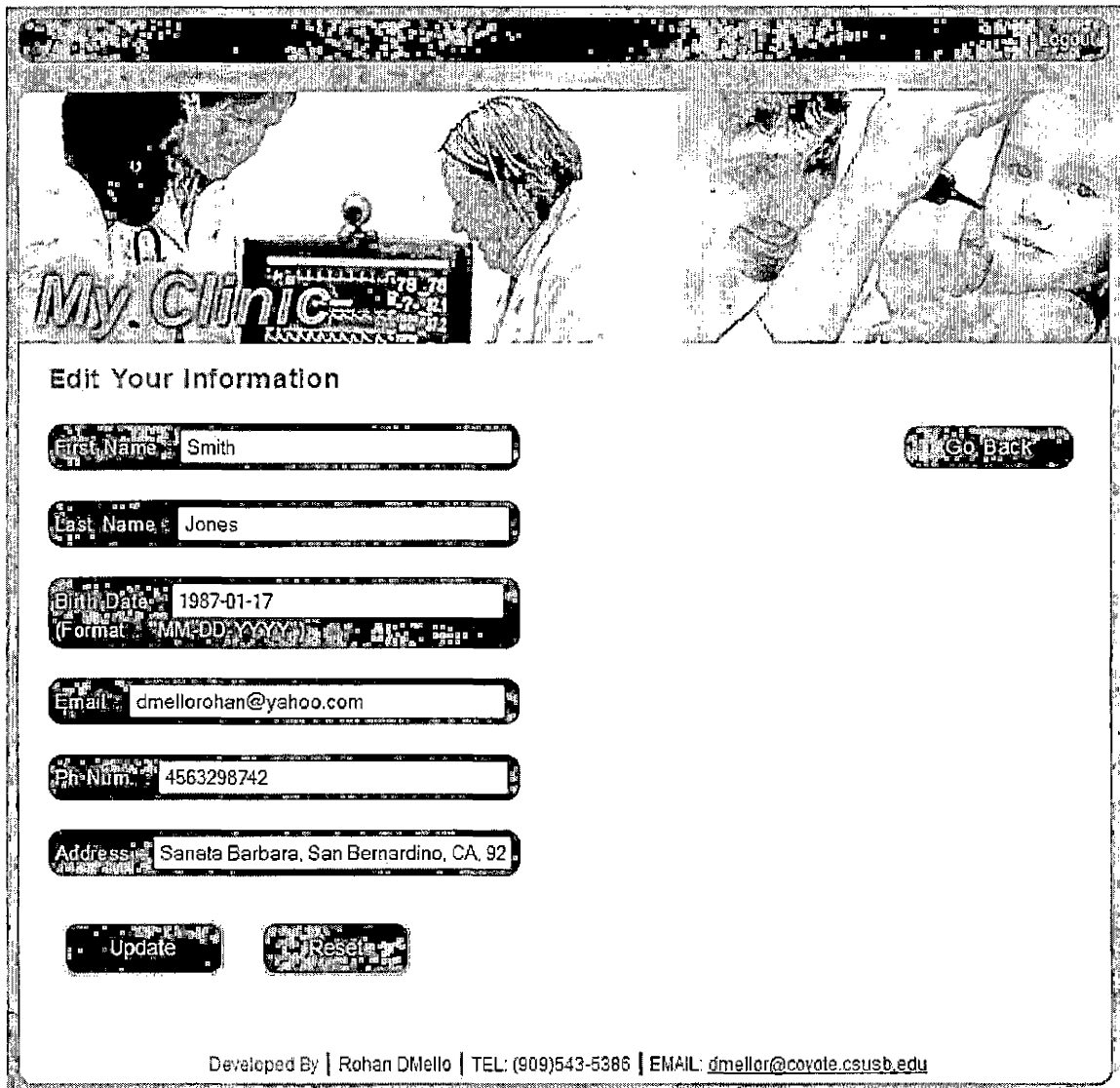


Figure 13. List of Patients Page

Edit Patient Page

This page edits the patient information. This page is similar to all the users namely doctor, patient and staff.



The screenshot shows a web interface for editing patient information. At the top, there is a banner with the text "My Clinic" and a "Logout" link. Below the banner, the title "Edit Your Information" is displayed. The form contains several input fields with the following data: First Name: Smith, Last Name: Jones, Birth Date: 1987-01-17 (Format: MM-DD-YYYY), Email: dmellorohan@yahoo.com, Ph-Num: 4563298742, and Address: Santa Barbara, San Bernardino, CA, 92. There are "Go Back", "Update", and "Reset" buttons. At the bottom, a footer contains the text: "Developed By | Rohan DMello | TEL: (909)543-5386 | EMAIL: dmello@covale.csusb.edu".

My Clinic

Logout

Edit Your Information

First Name:

Last Name:

Birth Date:
(Format: MM-DD-YYYY)

Email:

Ph-Num:

Address:

Developed By | Rohan DMello | TEL: (909)543-5386 | EMAIL: dmello@covale.csusb.edu

Figure 14. Edit Patient Page

Doctor Interface Pages

Doctor Home Page: This page is the first one that you get when you login as doctor. It has all the options that a doctor uses. Main functionality is to write prescription for a particular patient. The following snapshot shows the home page for doctor.

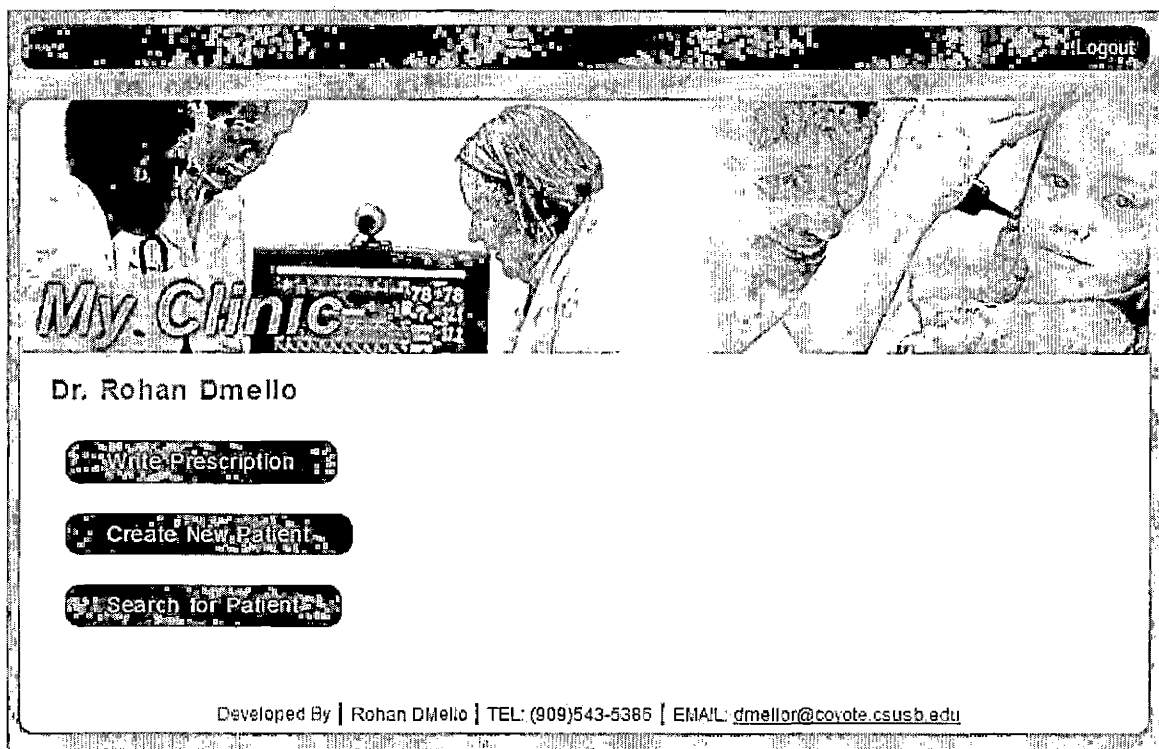


Figure 15. Doctor Home Page

Write Prescription Page: This particular page is used to write the prescription for a patient. After saving the prescription into the database it will be visible to both patient and pharmacy as well.

Logout

My Clinic

Prescription

First Name

Last Name

Email ID

Medicine Name

Consumption Duration

Consumption Per Day

Comments

Refills

Go Back

Save Reset

Developed By | Rohan DMello | TEL: (909)543-5386 | EMAIL: dmello@coyote.csusb.edu

Figure 16. Write Prescription Page

Result Page: The result page shows the patients Information and gives doctors ability to delete the patient or edit their information. The following figure shows the results page.

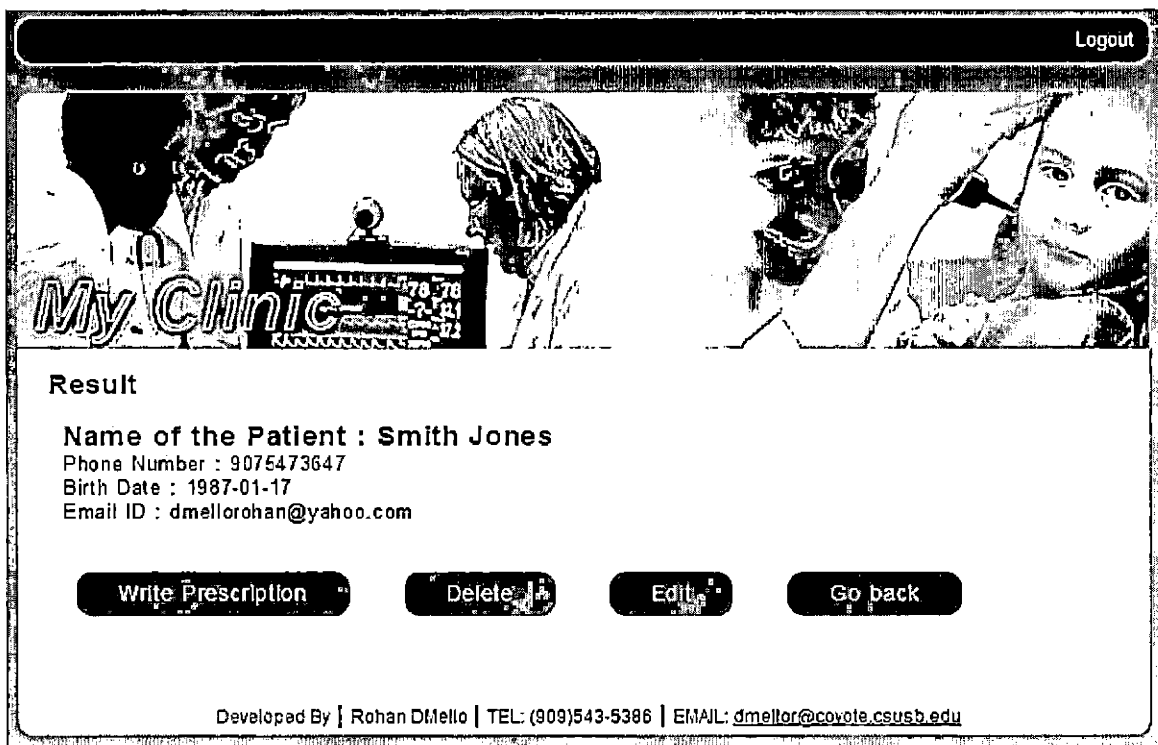
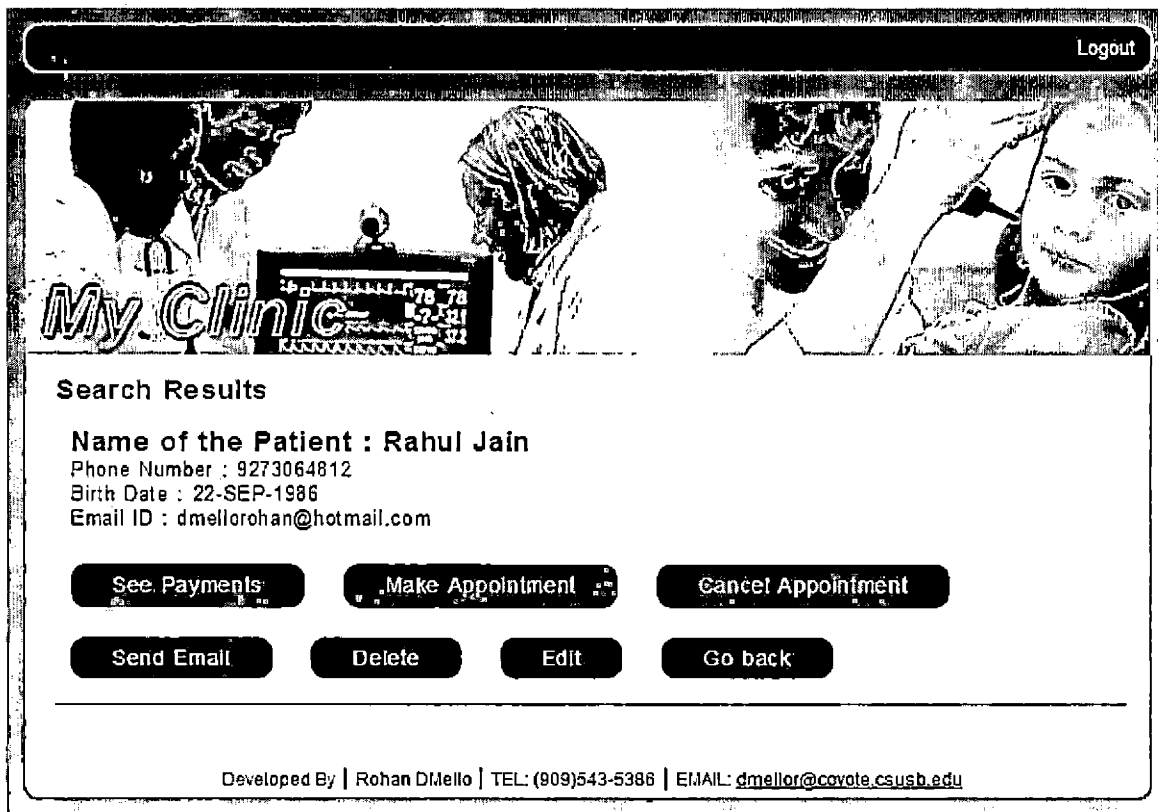


Figure 17. Doctor's Result Page

Staff Interface Pages

Create New Patient Page: This page has the same interface as the 'create new patient' page in doctor module. Staff usually creates the new patient and saves it into the database.



The screenshot shows a web application interface for a clinic. At the top right, there is a 'Logout' button. Below this is a banner image featuring the text 'My Clinic' and a collage of people. The main content area is titled 'Search Results' and displays patient information for 'Rahul Jain'. Below the patient details, there are six buttons arranged in two rows: 'See Payments', 'Make Appointment', 'Cancel Appointment', 'Send Email', 'Delete', 'Edit', and 'Go back'. At the bottom, a footer line provides contact information for the developer, Rohan DiMello.

Logout

My Clinic

Search Results

Name of the Patient : Rahul Jain
Phone Number : 9273064812
Birth Date : 22-SEP-1986
Email ID : dmellorohan@hotmail.com

See Payments Make Appointment Cancel Appointment

Send Email Delete Edit Go back

Developed By | Rohan DiMello | TEL: (909)543-5386 | EMAIL: dmellor@covote.csusb.edu

Figure 18. Staff Result Page

Payments Page: The make payment page is used to make the payments of the patients. It has three options as follows

1. Cash
2. Check
3. Insurance

Logout

My Clinic

Enter Payment Details

Amount :

Method :

Bank Name :

Check Number :

Company Name :

Policy Number :

Co-pay :

Status :

[Go Back](#)

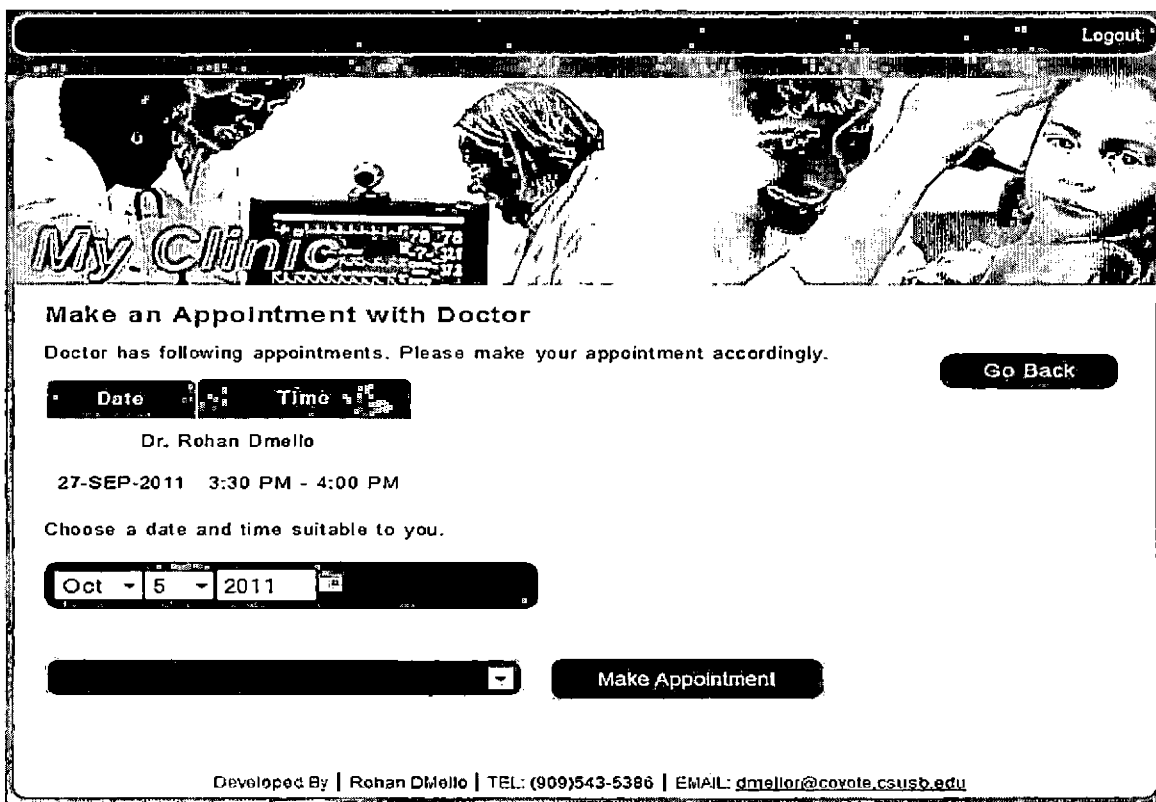
[Update](#) [Reset](#)

Developed By | Rohan DMello | TEL: (909)543-5386 | EMAIL: dmello@covate.csusb.edu

Figure 19. Make Payments Page

The method of payment is a select option. If we select the cash or insurance method the text box belonging to each of the method gets enabled. The following figure shows the interface for the payments page.

Make Appointment Page: An appointment is made by selecting a doctor and then picking up an empty slot. Then on the next page it makes the appointment. The following figure shows the interface.



The screenshot shows a web application titled "My Clinic" with a header image of medical professionals. The main heading is "Make an Appointment with Doctor". Below this, it states "Doctor has following appointments. Please make your appointment accordingly." and provides a "Go Back" button. The appointment details for "Dr. Rohan Dmello" are listed as "27-SEP-2011 3:30 PM - 4:00 PM". A prompt "Choose a date and time suitable to you." is followed by a date selection interface showing "Oct 5 2011". There is a dropdown menu for time selection and a "Make Appointment" button. The footer contains contact information: "Developed By | Rohan DMello | TEL: (909)543-5386 | EMAIL: dmello@covote.csusb.edu".

Logout

My Clinic

Make an Appointment with Doctor

Doctor has following appointments. Please make your appointment accordingly.

Date **Time**

Dr. Rohan Dmello

27-SEP-2011 3:30 PM - 4:00 PM

Choose a date and time suitable to you.

Oct 5 2011

Make Appointment

Developed By | Rohan DMello | TEL: (909)543-5386 | EMAIL: dmello@covote.csusb.edu

Figure 20. Make Appointment Page

Add Pharmacy Page: This particular page is to add the pharmacy on their clinic's panel. The following figure shows the interface of the page.

Logout

My Clinic

Add Pharmacy Details

Pharmacy Name :

User Name :

Password :

Confirm Password :

Email :

Ph Num. :

Address :

Developed By | Rohan DMello | TEL: (909)543-5386 | EMAIL: dmello@coyote.csusb.edu

Figure 21. Add Pharmacy Page

Edit Page: This page interface is same for the Doctor, patient and staff to edit information of the patient. The figure below shows the interface of the page.

Logout

My Clinic

Edit Your Information

First Name :

Last Name :

Birth Date :
(Format : "MM-DD-YYYY")

Email :

Ph Num. :

Address :

Developed By | Rohan DMello | TEL: (909)543-5386 | EMAIL: dmellor@coyote.csusb.edu

Figure 22. Edit Patient Info Page

Patient Interface Pages

Edit Information Page: As mentioned in the previous sections the edit page interface is same for all the best

Search Prescription page: The search prescription page gives patient ability to search page by prescription date.

The following figure shows the interface.

Logout

My Clinic

Search Prescription Details

Date of Prescription :
(Format : "DD-MON-YYYY") eg. 22-SEP-2011

Go Back

Search Reset

Developed By | Rohan DMello | TEL: (909)543-5386 | EMAIL: dmellor@covote.csusb.edu

Figure 23. Search Prescription Page

Prescription List Page: Once the prescription is searched by date or searches all the prescription. Then all the matching results will be listed by the date given the prescription and will have a view option to view it. The page looks like as shown in the following figure.

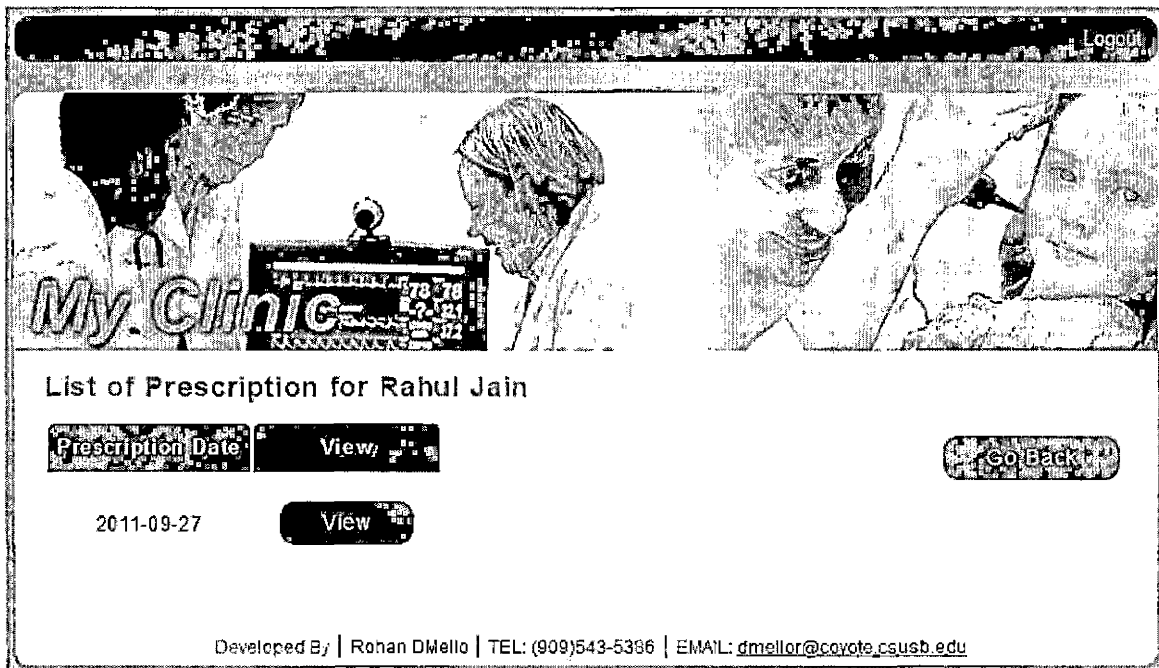


Figure 24. Prescription List Page

View Prescription Page: This page is being navigated from Prescription List page where is displays all the description of the prescription on particular day. The following figure shows how the prescription details are displayed.

Logout

My Clinic

Prescription for Rahul Jain on 2011-09-27

[Go Back](#)

Doctor's Name	Medicine Name	Consumption Duration	Consumption Per Day	Comments	Refill
Dr. Rohan Dmello	Paracetamin	3 days	2 times	Take after meal	0

Developed By | Rohan Dmello | TEL: (909)543-5386 | EMAIL: dmello@coyote.csusb.edu

Figure 25. View Prescription Page

Pharmacy Interface Pages

Pharmacy Home Page: This page interface has all the option that are provided for a pharmacy. The following figure shows the home page for the Pharmacy Home page.

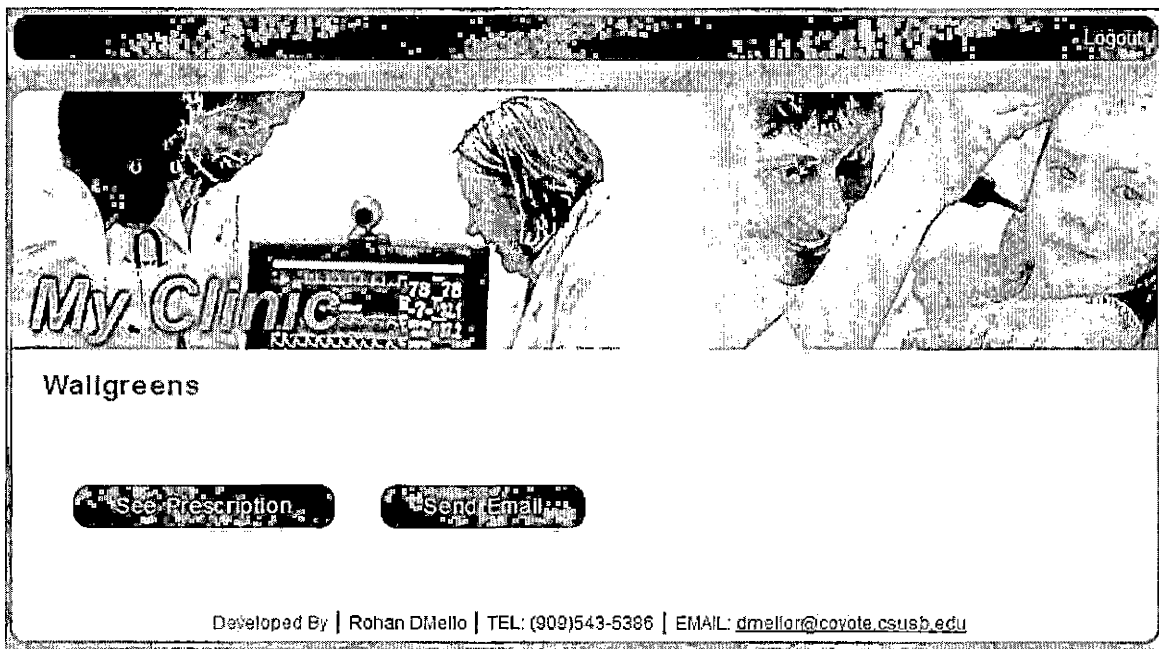


Figure 26. Pharmacy Home Page

See Prescription Page: This page first takes you to the search the patient and then after searching for the particular patient and then it gives the list of prescription for the Patient. Then after displaying the particular prescription it shows the prescription as shown in the figure below.

Logout

My Clinic

Prescription

[Go Back](#)

Doctor's Name	Medicine Name	Consumption Duration	Consumption Per Day	Comments	Refill	Status
Dr. Rohan Dmello	Solix	3 days	2 times	take after meal	2	No Action

[Ready](#)

Developed By | Rohan Dmello | TEL: (909)543-5386 | EMAIL: dmellon@covale.csusb.edu

Figure 27. Pharmacy Action Page

CHAPTER FIVE

SYSTEM VALIDATION

This section documents the performance of the 'My Clinic' System. The purpose of the system validation is to evaluate the attributes or capabilities of 'My Clinic' system and determine if it meets the required results. The following sub sections depict the results in different levels of testing.

Unit Testing

Unit testing is the first testing done in the testing phase of Software Development Lifecycle. In this phase the individual units are tested to ensure that they work as they are intended to work. The results are shown in the following table.

Table 13. Unit Testing Results Table

Page	Test Performed	Results
Static Pages	1. Check if all the style sheets are loaded when the site is loaded. 2. Disable all the all style sheets and check if all the descriptive images have alt attribute. 3. Check for no link duplication on	Pass

	the same page.	
Login page	<ol style="list-style-type: none"> 1. Check for null input in the text field for username and password field. 2. Once logged in, check if the page redirects to the intended user page. 3. If the username and/or password entered is wrong, it should show error message on login page. 	Pass
Search page	<ol style="list-style-type: none"> 1. Check if the page shows you the expected result. 2. Check if there is some problem it should show an error message and redirect to index page. 3. Check if there are no results it should say so. 	Pass
Doctor Home page	<ol style="list-style-type: none"> 1. Check if all the options are displayed on the doctor home page as intended. 2. Verify all the links are live. 3. Check for duplicate links. 	Pass
Write	<ol style="list-style-type: none"> 1. Check all the text boxes appear on 	

Prescript-ion	<p>the page.</p> <p>2. Check if all drop-down boxes have their values displayed in the box.</p> <p>3. After update it should redirect to doctor home page with success message.</p>	Pass
Create Patient	<p>1. Check if the page displayed with all the text boxes.</p> <p>2. After submitting the information it should be entered into the database.</p> <p>3. After database entry it should be redirected the page it was called from.</p>	Pass
Search Page	<p>1. Enter the name of the patient and check if shows any results or shows "No patient with such name" message.</p> <p>2. Check if the back button takes you back to the same page you came back from.</p>	Pass
Edit Page	<p>1. Verify all the text boxes are filled with information extracted from database.</p>	Pass

	<p>2. Once submitted, check the database if it is updated.</p> <p>3. Check if it is redirected to the proper page.</p>	
Delete patient	<p>1. Use the delete button and check if the page is redirected to the intended page.</p> <p>2. Check the database if all the data associated with a particular patient is deleted.</p>	Pass
Payments Page	<p>1. If the payment status is pending then the make/edit payment link should be activated.</p> <p>2. The make/edit payment link should take to the appropriate page.</p> <p>3. If the status is other than pending or paid, the make/edit link should take you to the page having only status update page.</p>	Pass
Make / Edit Payment page	<p>1. Check if all the options are on the page displayed.</p> <p>2. Check if in the dropdown menu method, cash or insurance is selected the appropriate text</p>	Pass

	boxes gets activated.	
Make Appointment page	<ol style="list-style-type: none"> 1. Check if the link takes you to show page, that will show the list of doctors to choose from. 2. After selecting a doctor it should take you to the page to make appointment. 3. Check if after making appointment it should take you to the page with update message. 	Pass
Send Email	<ol style="list-style-type: none"> 1. Check if the page is loaded the 'from' text box has the email id of the sender already loaded. 2. Check if email is sent, once send email buttons is clicked. 	Pass
Add Pharmacy Page	<ol style="list-style-type: none"> 1. Check if the page gets redirected to the intended page. 2. Check if the pharmacy is added in to the database and staff page should have success message. 	Pass
See Prescript-ion	<ol style="list-style-type: none"> 1. Check if the page shows the particular prescription of the patient. 	Pass

Search Pharmacy	1. Verify the page contains results with pharmacy address and a link to the map.	Pass
List Prescript-ion page	1. Verify the list is shown with view button. 2. Check if the message 'No Prescriptions' is displayed if there are no prescriptions.	Pass
Show Prescript-ion page	1. Check if the page is displayed with 'Ready' button. 2. Check if the page is displayed with 'Done' Button.	Pass

System Integration Testing

System testing is performed once all the units are tested properly. 'My Clinic' is divided into four modules, Doctor, Staff, Patient and Pharmacy. The testing is done on complete integrated system to evaluate the system's compliance with all the requirements as well as compliance with ADA. Table 5.2 shows the system integration testing.

Table 14. System Integration Test Results Table

System / Module	Tests Performed	Results

Doctor	<ol style="list-style-type: none"> 1. Verify only authenticated personnel can enter into the doctor module. 2. Verify all the pages functions properly that are listed in doctor module. 3. Check if the doctor can create patient, write prescription. 	Pass
Staff	<ol style="list-style-type: none"> 1. Verify only authenticated staff personnel can enter into staff module. 2. Verify all the links on the staff module works properly and make database calls when needed. 3. Check if the staff member can create patient, make appointment, cancel appointment, send email to patient delete patient and edit patient's information. 	Pass
Patient	<ol style="list-style-type: none"> 1. Verify only authenticated patients can only enter the system. 2. Verify all the links on the patient are working. 3. Check if the patient can make appointment, see prescription, send 	Pass

	email to clinic, and search pharmacy	
Pharmacy	<ol style="list-style-type: none"> 1. Verify only pharmacies that are on panel of the clinic can login. 2. Check if the pharmacy can see prescription and send an email to patient once prescription is ready and update database. 	Pass

System Testing

Table 15. System Test Results Table

System	Tests	Results
MySQL	Install MySQL and startup and stop the service and verify for it working.	Pass
Tomcat	Install Tomcat web server, start the server, stop server and verify its working with no errors	Pass
Browser	Check the system on all the browser	Pass
Data test	Test the system with all the real data and see if system works without errors.	Pass

CHAPTER SIX

MAINTENANCE MANUAL

Maintenance manual is an important aspect of any software application. So without stable and scalable web server and database server 'My Clinic' will not be up all the time. The following subsections are dedicated for installation and configuration of the system.

Downloading and Installing CentOS

CentOS is stable, free and fully functional distribution OS it was chosen to be operating system for the server side.

To install CentOS, the easiest method is to download it from CentOS official website [14] and burn it on to a disk. The installation steps are given in CentOS Server Base install website [15]. The steps should be followed as instructed, so that there are no issues at the time of tomcat installation or MySQL installation.

Downloading and Installing Java Development Kit

Download the latest version of the JDK (Java Development Kit) from Java Official Website [16]. Download the following version:

Linux x86 - RPM Installer

Change the privileges on the file and make it executable.

```
$ chmod +x jdk-<version>-linux-i586-rpm.bin
```

```
$ ./jdk-<version>-linux-i586-rpm.bin
```

Setup java environment variables as follows:

```
export JAVA_HOME=/user/java/latest
```

```
export PATH=$PATH:$JAVA_HOME/bin
```

Downloading and Installing Tomcat

Apache tomcat is a web contained developed by Apache Software Foundation. Tomcat implements JSP and servlet specifications. Download the latest tomcat distributions from Tomcat Official website [17].

Install the tomcat from following website [18] and follow all the instruction, so that it is configured to the requirements of our application.

Downloading and Installing MySQL Database

'My Clinic' application is a web application having to save data into database. MySQL is the best choice for the application. Download the MySQL from it official website [19]. Install it on CentOS using the instruction give in the following website [20].

Create a new database and name it 'myclinic' using following instruction:

```
create database myclinic;  
  
use myclinic;
```

Database Backup

The data stored into the database must be taken onto an external storage device. The backup is important in case of loss of data. The following instruction shows how to take a backup of database:

```
$ mysqldump -u root -p <DB_pas> <NAME_OF_DATABASE>.sql
```

System Backup

It is also important to take a backup of the system. Such as JSP files, Java class file, configuration files etc. and save them on an external device.

CHAPTER SEVEN

CONCLUSION AND FUTURE DIRECTIONS

Conclusion

'My Clinic' is an excellent application for small clinics, since it is usually very costly for small clinics to use enterprise level applications like used in Hospitals or other HealthCare Organizations. It has many features that are useful to the clinics for doing many tasks with less time and more efficiently. The interesting feature of the application is that it can send a quick email using the in build send email feature. This application also has pharmacy module that helps patients and doctors a lot since most of the communication can be done online saving lot of time. Since the system is web based it give doctor ability to manage more than one clinic. It gives patients ability to view their prescription online, view their appointments. The system eliminates a lot of paperwork and will be more efficient in terms of correctness and time consumption.

Future Directions

The development of this project is directed towards small clinics, but it still has a lot of scope to achieve the features provided in the enterprise application. Some of the work that should be done in the future is list as follows:

- Implement file upload/download for patient's reports.
- Adding payment method by Credit card/Debit card.
- Adding Google map API to search near-by pharmacy based on patient's location.
- Build a smartphone application for patient to check information on fly.

REFERENCES

- [1] SDK [Accessed: 30 September 2011]
http://en.wikipedia.org/wiki/Software_development_kit
- [2] Java Programming Language [Accessed: 30 September 2011]
[http://en.wikipedia.org/wiki/Java_\(programming_language\)](http://en.wikipedia.org/wiki/Java_(programming_language))
- [3] Eclipse IDE [Accessed: 30 September 2011]
[http://en.wikipedia.org/wiki/Eclipse_\(software\)](http://en.wikipedia.org/wiki/Eclipse_(software))
- [4] JSP - Java Server Pages [Accessed: 30 September 2011]
http://en.wikipedia.org/wiki/JavaServer_Pages
- [5] Java Servlets [Accessed: 30 September 2011]
http://en.wikipedia.org/wiki/Java_Servlet
- [6] API [Accessed: 30 September 2011]
<http://en.wikipedia.org/wiki/API>
- [7] HTTP [Accessed: 30 September 2011]
<http://en.wikipedia.org/wiki/HTTP>
- [8] UML [Accessed: 30 September 2011]
http://en.wikipedia.org/wiki/Unified_Modeling_Language
- [9] XML [Accessed: 30 September 2011]
<http://en.wikipedia.org/wiki/XML>
- [10] Dia UML tool [Accessed: 30 September 2011]
[http://en.wikipedia.org/wiki/Dia_\(software\)](http://en.wikipedia.org/wiki/Dia_(software))

- [11] HTML [Accessed: 30 September 2011]
<http://en.wikipedia.org/wiki/Html>
- [12] CSS [Accessed: 30 September 2011]
http://en.wikipedia.org/wiki/Cascading_Style_Sheets
- [13] JavaScript [Accessed: 30 September 2011]
<http://en.wikipedia.org/wiki/JavaScript>
- [14] Centos Official Website
<http://www.centos.org/modules/tinycontent/index.php?id=30>
- [15] Centos Base Server Install
http://cse.csusb.edu/turner/wiki/CentOS_Server_Base_Install
- [16] Java Official Website
<http://www.oracle.com/technetwork/java/javase/downloads/jdk-6u25-download-346242.html>
- [17] Tomcat Official Website
<http://tomcat.apache.org/download-70.cgi>
- [18] Tomcat Installation Guide
http://cse.csusb.edu/turner/wiki/CSE_405_Tomcat_Setup_Instructions_for_Lab_Accounts
- [19] MySQL Official Website
<http://dev.mysql.com/downloads/mysql/>
- [20] MySQL Installation Guide

http://cse.csusb.edu/turner/wiki/CSE_405_MySQL_Setup_Instructions_for_Lab_Accounts

[21] MySQL 5.5 reference manual.

<http://dev.mysql.com/doc/refman/5.5/en/>

[22] R. S. Pressman, *Software Engineering: A Practitioner's Approach*, Sixth Edition, New York, NY: McGraw-Hill Companies, Inc., 2005.

[23] MVC Architecture [Accessed: 14 September 2011]

<http://en.wikipedia.org/wiki/Model-view-controller>