Clinic Management System

(Project Report)

**Group B**

Team Members

1. Ipsit Patra
2. Shravan Varma Epuri
3. Anusha Reddy Guntakandla
4. Asim Mohammed
5. Andrew Sennebogen

Table of Contents

Goals and Objectives………………………………………………………………………………. 3

Subsystem Decomposition……………………………………………………………………… 4

State Machine and Control Flow……………………………………………………………… 5

Technologies and Libraries used in the project………………………………………… 6

Functional Description……………………………………………………………………………. 7

JMeter Result Evaluation…………………………………………………………………………10

Test Case…………………………………………………………………………………………………12

Traceability Matrix…………………………………………………………………………………..14

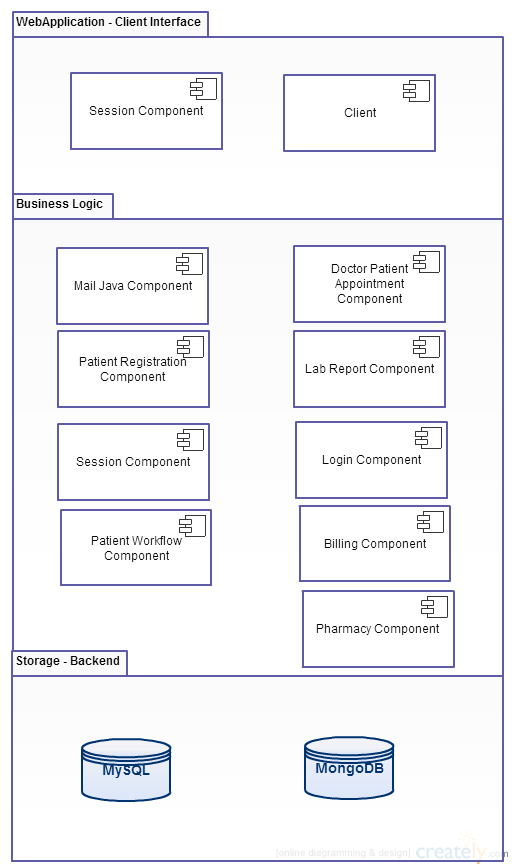
**Goals and Objectives**

Clinic Management System is intended to ease tracking of the Patients information required by the hospital in an organized and quickly accessible manner. This System is developed for different users to access role based information. The doctors can view appointments with the patients and a particular patient’s medical history. It helps Patients to check and book appointments with doctors. This system would improve communication across the various departments in the hospital without having individuals to submit patient’s records in person. It also provides centralized billing for the charges incurred by the patient at the various departments in the hospital. These bills can be claimed based on the insurance policy the patient has.

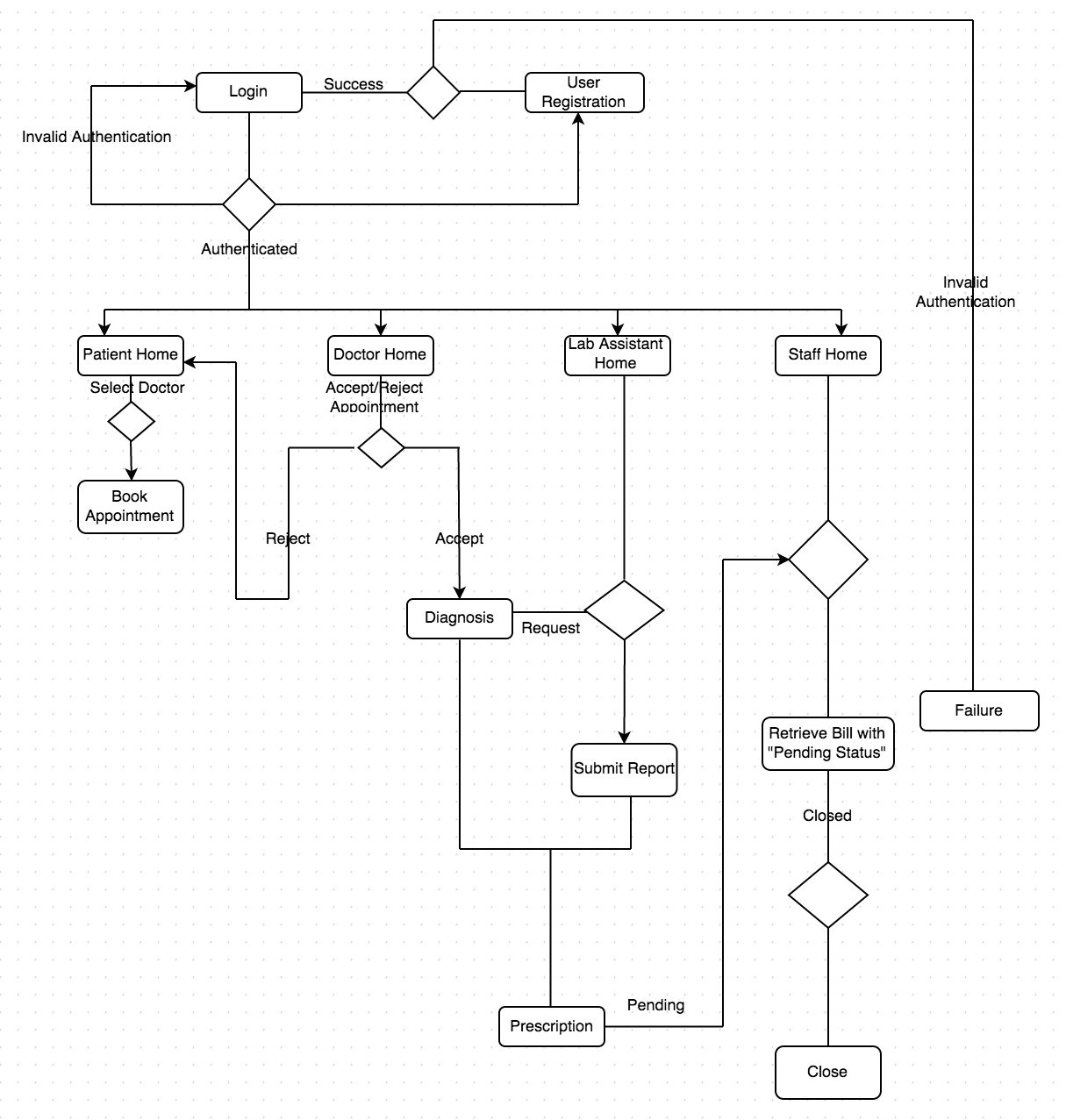
There are four major functionalities that the software does:

* **Register Patient**- This is to add new Patients within the system. This would include details of the patient like the name, address and phone number.
* **Schedule an Appointment**- Once the patient have been registered, he can book the appointment based on Doctor availability. This doctor can accept or reject the appointment based on the priority of other patients under his list.
* **Patient Diagnosis Workflow** - Here, the patient can be referred to Lab for diagnosis and then Patient can reschedule the appointment with the doctor once the lab reports are available for pickup. If the patient is not referred for Diagnosis then the prescription is sent to pharmacy.
* **Billing** –The patient’s bill is based on the doctor, diagnosis and prescription fee.

**Subsystem Decomposition**



**State Machine and Control Flow**



**Technologies and Libraries used in the project**

* Java EE
* Eclipse
* WebServices
* Enterpise Java Beans
* Json
* JUnit
* JMeter
* Nosql(Mongo DB)
* Relational DB(MySQL)

**Functional Description**

**Login Module**

There are four different users who can login to the clinic management system website and perform their respective duties.

1. Doctors
2. Patients
3. Lab Assistants
4. Hospital staff

User is presented with login form. Once username and password is entered, Login Module determines if the user entered right credentials. If the credentials are correct, user is allowed to access rest of the content. If credentials were not correct, user is prompted for credentials again. The user is navigated to the home page on successful authentication. The username and password fields are mandatory fields. The action cannot be completed if either one is missing.

**New User Registration**

Clinic Management System allows new users to create new account. The login account is a must to access their respective accounts.

The following details are required for creating a new user account

* Name
* Email
* Date of Birth
* Phone Number
* Address

All the fields are mandatory.

On submitting the values of all the required fields a new user is created and the details are inserted into the respective user details table in the database. The user would receive an email with their respective password, which can be changed to their desired user name later on by themselves. This verifies the email id provided by the user.

Once the required values are stored in the database the user is navigated to the login page. The user will be able to login using the newly created username and password.

**Patient Module**

The patient is routed to the home page after successful login. The patient will be presented two options before him/her.

1. **View all appointments**: This part of the home page presents all the appointments that the patient has in the coming future. These appointments are the ones, which the patient requested for, and are accepted by the doctor.
2. **Book an appointment**: This part of the home page allows the user to request an appointment with the doctor based on the problem he is suffering from. The following details are required for the patient to book an appointment.

* Specialization of the doctor
* Name of the doctor
* Appointment date
* Appointment Time
* Severity of the problem

All the fields are mandatory.

The patient is let known if the requested doctor is available and if available, the details are stored in the database.

**Doctor Module**

The doctor is routed to the home page after successful login. The patient will be presented two options before him/her.

1. **View all the pending appointments:** This part of the home page presents all the appointments that the doctor has in the coming future which are not accepted. These appointments are the ones, which the patient requested for, and can be accepted by or rejected by the doctor. The corresponding status is updated in the database after the doctor accepts or rejects.
2. **Routing the requests to Lab or Staff:** This part of the home page contains the appointments which are accepted and in process. The doctor would have two options that he can choose from to continue with the patient appointment.

* **Diagnosis**: The doctor can send the patient Id to a lab assistant for further diagnosis, which includes the test that needs to be done on the patient. This page requires the doctor to fill in the following details to submit the patient details for diagnosis.
  + 1. Lab Assistant name
    2. Type of Test
* **Prescription:** Once the doctor receives the report of the diagnosis, he can prescribe the patient with the required medication and send to the Hospital staff. This page requires the doctor to fill in the following details.
  + - * + Medicine Name and Quantity
        + Short note

**Lab Assistant Module**

The Lab Assistant is routed to the home page after successful login. She/he would have all the details of the patient sent by the particular doctor and the details of the test that need to be done on the patient.

Once he performs the required test, the lab assistant will be given an option to select the patient which routes him to the next page where in he/she can upload the report and submit the same.

The format of the forms that the lab assistant can upload is presented on the page.

**Staff Module**

The staff user is routed to the home page after successful login. The user would be presented with all the details of the patient, types of test undergone and the prescription that the patient is prescribed. This allows the staff user to calculate the bill and generate the same. Once the staff generated the bill, he closes the workflow.

**JMeter Result Evaluation**

=>J-Meter was used to- evaluate the various times taken to evaluate the queries to the database. Since the queries to the databases are the main bottleneck, the various database queries were simulated for varying thread loads (Number of Clients).

=> Environment Variables Used:

No of Threads (Users): 100  
Loop count: 10  
Hence we have 100\*10 samples: 1000 for each query.

Each Query had an Error % of 0%

Email module had an error of 13.50% because of the huge

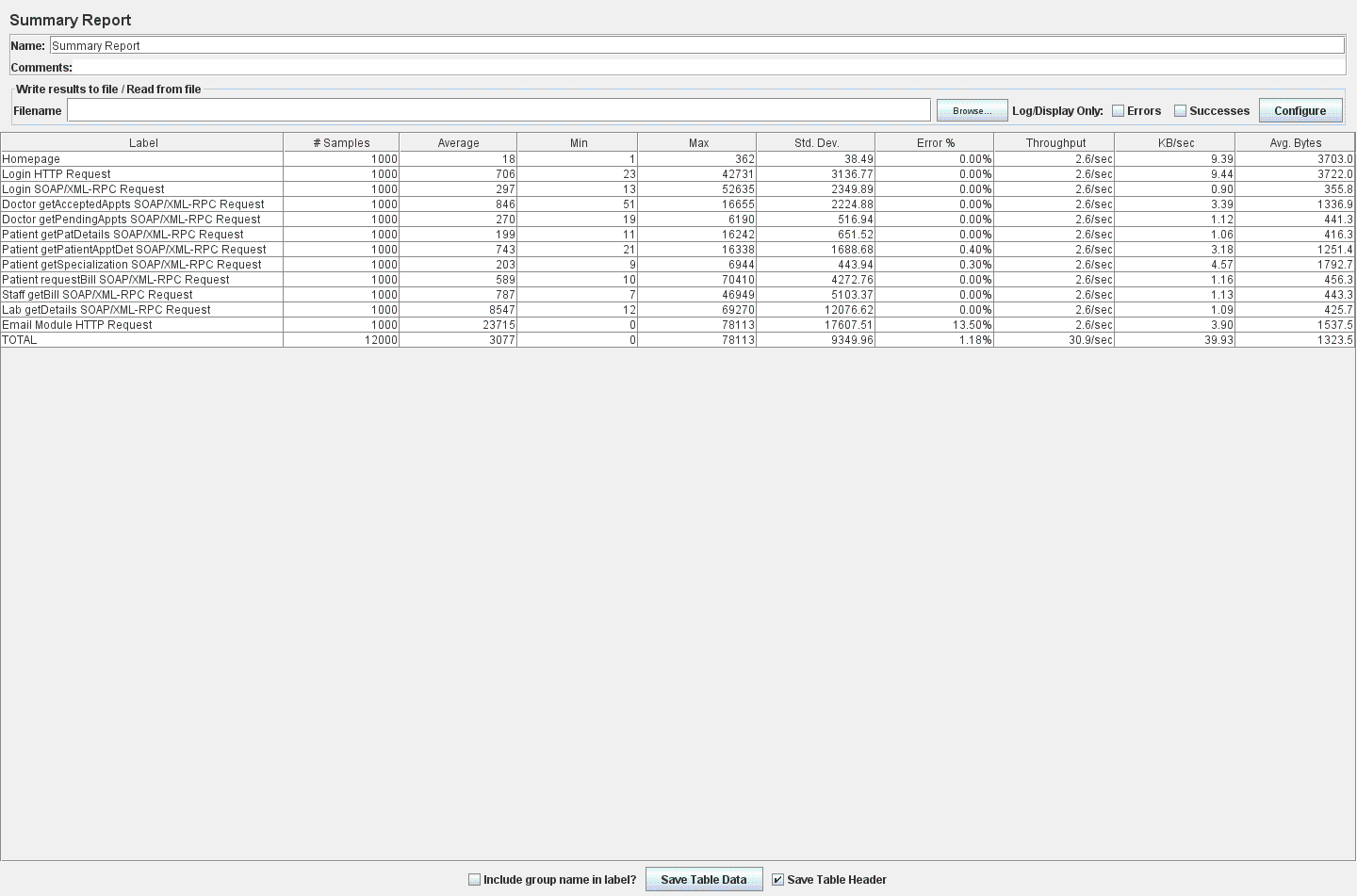
login attempts from the Gmail server

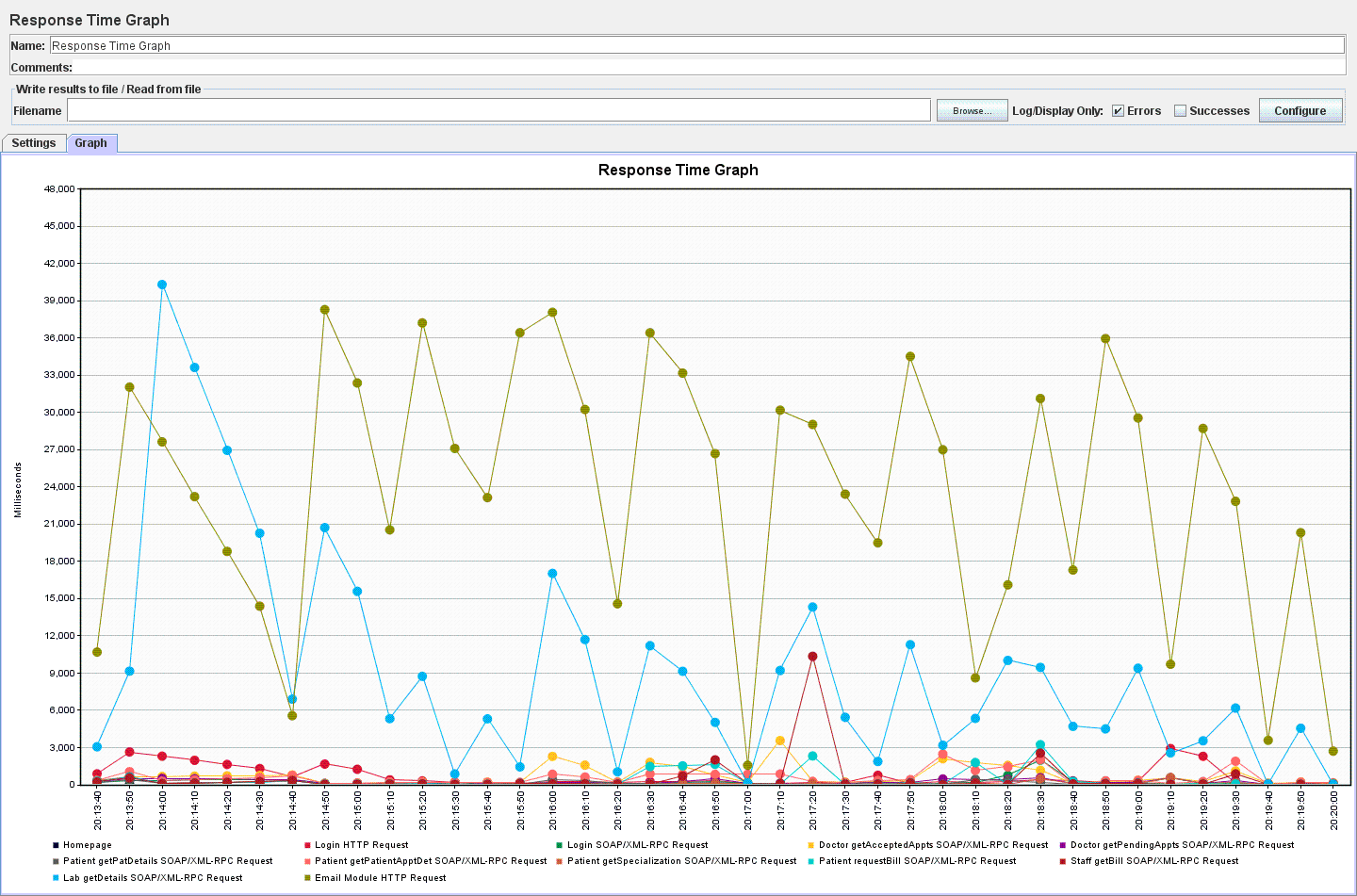
(**Log**: javax.mail.AuthenticationFailedException: 454 4.7.0 Too

many login attempts, please try again later.

l2sm16680786ioe.34 – gsmtp)

The average Standard Deviation was: 9349.96 ms. The average Throughput was: 30.9 Kbits/s  
The Response Time Graph is shown below:

****

****

**Test Case**

Testing Patient

PatientTest.java

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test case | Method tested | Input | Expected Output | Result |
| testGetPatApptDet | getPatApptDet | 2000 | 6 | PASS |
| testGetSpec | Getspecialization | N/A | 50 | PASS |
| testGetDocNames | getdocNames | “Allergist” | 203 | PASS |
| testGetTimings | Gettimings | 1001 | 48 | PASS |
| testGetPatDet | getPatDetails | 2001 | “Hop” | PASS |
| testDocAvail | docAvailability | 1000, "12/25/2015", "01:00-01:30" | false | PASS |

Testing Doctor

DoctorTest.java

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test case | Method tested | Input | Expected Output | Result |
| testGetAcceptAppoints | getAcceptedAppoints | 1001 | 4 | PASS |
| testGetlabAssistDet | getLabAssistDet | N/A | 100 | PASS |
| testGetpendAppts | getPendingAppoints | 1001 | 1 | PASS |
| testUpdDocDiagnosis | updDocDiagnosis | 100001, 3000, 2000 | True | PASS |
| testUpdDocPrescrip | updDocPrescrip | 100001, "Crocin", "Normal Fever" | True | PASS |
| testUpdDocPatAppt | updDocPatAppt | “Reject”, 100010 | True | PASS |

Testing Login

LoginTest.java

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test case | Method tested | Input | Expected Output | Result |
| testLogin | validateCredentials | cms@gmail.com | Password@220 | PASS |
| testAttempts | getNoOfAttempts | cms@gmail.com | 2 | PASS |
| testStatus | getStatus | cms@gmail.com | Accept | PASS |

Testing Staff

StaffTest.java

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test case | Method tested | Input | Expected Output | Result |
| testStaff | getStaffCred | 5001 | Present | PASS |
| getStaffDet | getStaffDet | N/A | 5 | PASS |
| sendBillStatus | billStatus | 5001 | Billed | PASS |

Testing Lab

LabTest.java

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test case | Method tested | Input | Expected Output | Result |
| testGetDet | getLabDet | 2000 | 1 | PASS |
| testLabRec | getLabRec | 5001 | Present | PASS |
| getLabDet | getDet | N/A | 5 | PASS |
| testLabReportUpdate | labReportUpdate | 100001, 2000, 1000, 5000, "9009", "negative" | False | PASS |

**Traceability Matrix**

