

Team Members:

Pratyusha Dommata-11091

Shruti Rajan-10904

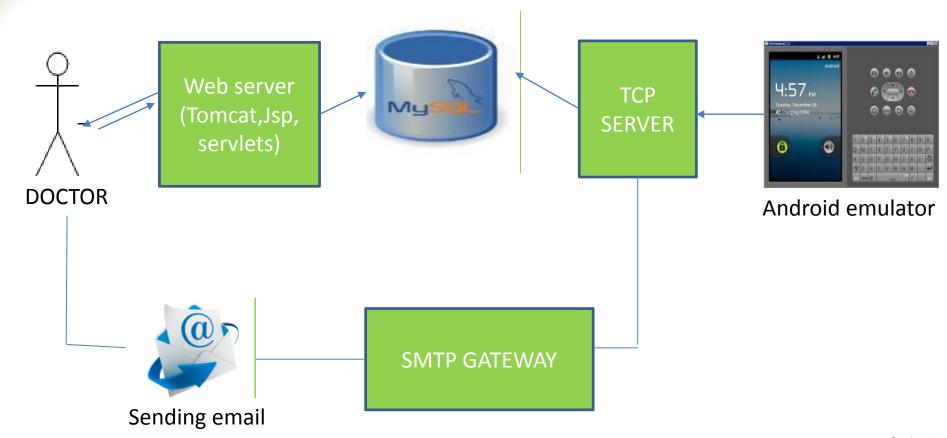
Sravani Anumula-12433

Uttam Polkampally-12211

Xianghui Ma-9957



Fall-Arm Architecture:





ANDROID

Android Application

- Patient has an Android Smartphone which is connected to Wifi or Cellular data
 Reads acceleration and gyroscope parameters and transmits the parameters to TCP Server
- Values for acceleration and gyroscope can be manipulated using a telnet connection.

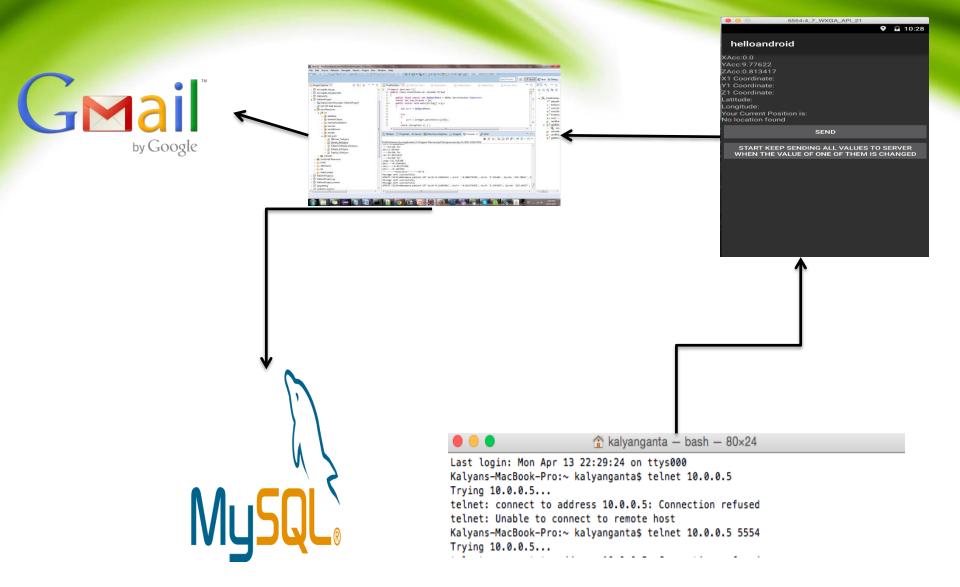
Accelerometer:

- Can detect movement and, more usefully, the rate of change of the speed of that movement.
- Measure the acceleration applied to the phone minus the force of gravity (x, y, z)

Gyroscope:

- Measure the current device orientation (<u>angular velocity</u>) on three axes in degrees
- Measure all type of rotations.
- Does not measure movement.
- Does not amplify hand jitter.

Demonstration:



```
239
                                                                         IP address which is
           try {
                                                                            required to let
240
          // socket = new Socket("172.19.0.126", 9999);
                                                                          another machine
241
               System.out.println("Creating socket at localhost 1");
                                                                            talk to android
242
          socket = new Socket("172.19.5.170", 9999)
                                                                         device, 9999 is the
243
          System.out.println("Creating socket at localhost");
                                                                        unique port number
244
            dataOutputStream = new DataOutputStream(socket.getOutputStream());
245
            dataInputStream = new DataInputStream(socket.getInputStream());
246
            String output_message= "pid:1234"+" "+xCoor.getText().toString()+" "+yCoor.getText().toString()+" "+zCoor.getText()
            dataOutputStream.writeUTF(dutput_message);
247
            540
                                                                   Patient ID
```

Two main elements of Location-Based Services (LBS) used to determine the devices current location

- ➤ Location Manager
- **≻**Location Providers

Use the **Location Controls** available from the <u>DDMS</u> perspective in Eclipse to push location changes directly into the emulators **GPS Location Provider**.

- ➤ Using the **Manual** tab you can specify particular latitude/ longitude pairs.
- ➤ Using the **KML** and **GPX** tabs let you load KML (Keyhole Markup Language) and GPX (GPS Exchange Format) files, respectively. Google Maps is one of the many applications bundled with the Android platform.

Demonstration on Android device(Smartphone):

- Make sure that the Android ADB can "talk to" your device.
 - Open a Windows Command Prompt (cmd.exe)/Mac Terminal.
 - Go to the folder (via cd) where ADB.exe is in, e.g, C:\Android\android-sdk\platform-tools.
 - Type adb devices
- 2. If your device is listed (serial number is displayed), go to the second check. Otherwise, this means ADB currently can't talk to your device. There're many reasons which can cause the problem, try:
 - In your device, turn on "USB Debugging Mode" in Settings -> Application -> Development.
 - Restart your Windows Mac.
 - Restart the Android device.
 - In Windows/Mac, reinstall device drivers.
- 3. Connect the device to your computer using the appropriate USB cable.
- 4. Inside your Eclipse installation, open the 'Run' menu and select 'Run Configurations.
- 5. Select the Android project you wish to run/debug and open the 'Target' tab.
- 6. Ensure 'Always prompt to pick device' is selected, click 'Apply' and 'Run.'
- 7. The 'Android Device Chooser' will open. If you've connected your device successfully, it will be listed under 'Choose a running Android device.' Select your device and click 'OK.'
- 8. Check your device your app will have made the leap from Eclipse onto your screen!

OUTPUT:

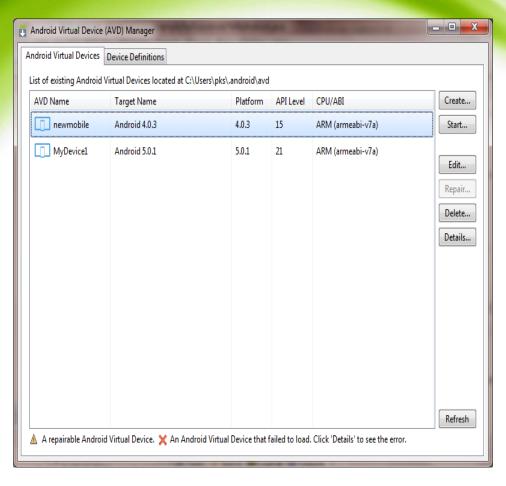


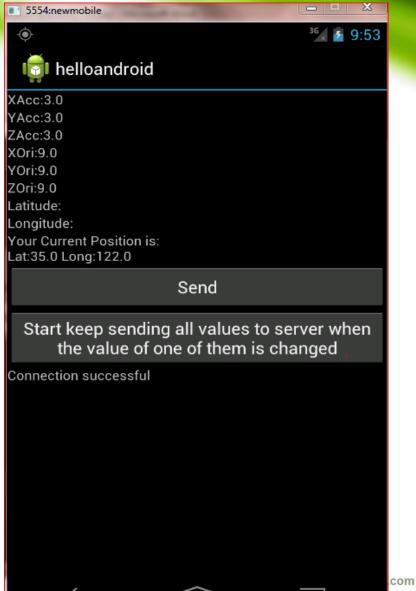
Android Client

Android Application:

- •Reads acceleration and gyroscope parameters and transmits the parameters to TCP Server
- •Values for acceleration and gyroscope can be manipulated using a telnet connection.

Android virtual Device





TELNET

```
C:\Windows\system32\cmd.exe

Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\pks>telnet 127.0.0.1 5554
```

```
Android Console: type 'help' for a list of commands
OK
sensor status
acceleration: enabled.

magnetic-field: enabled.

orientation: enabled.

temperature:

proximity: enabled.

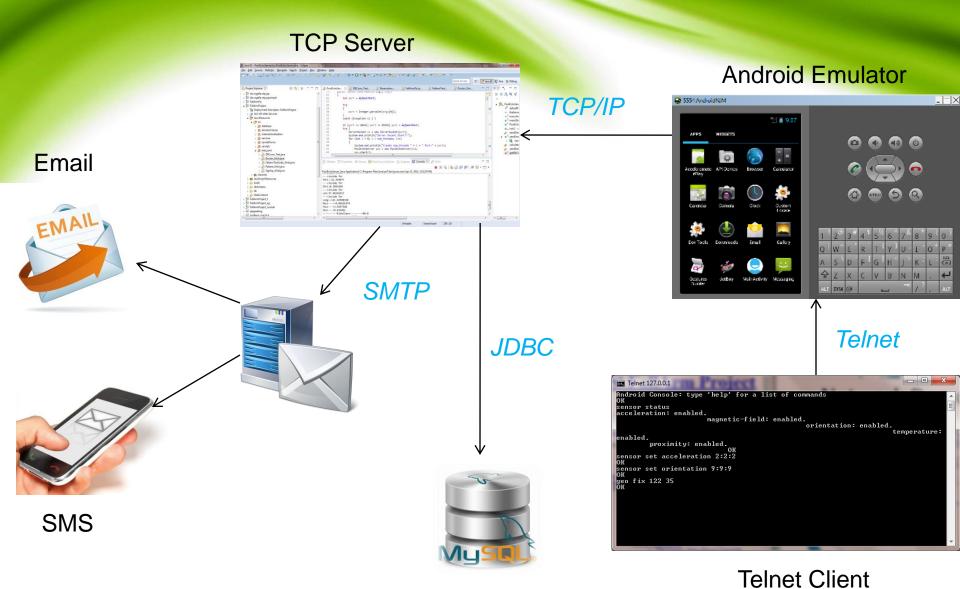
Sensor set acceleration 2:2:2
OK
sensor set orientation 9:9:9
OK
geo fix 122 35
OK

-
```



SOCKET PROGRAMMING

Network Demonstration



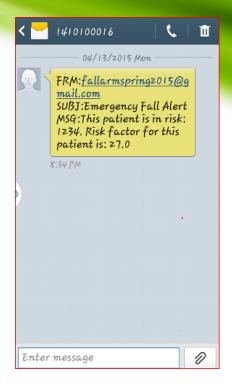
MySQL server

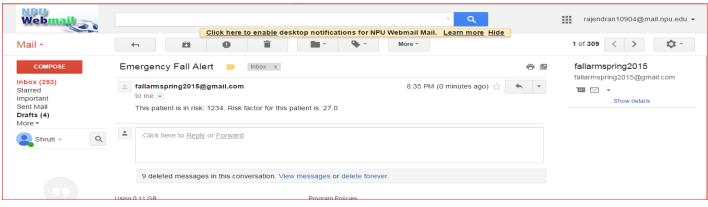
Socket Programming

- Using socket programming, server listens to device request on port 9999(Default Port, if not specified).
- Data sent from device :
 - Acceleration: 3 Float values X, Y, Z axis
 - Orientation: 3 Float values for X, Y, Z axis
 - Geo Location: Latitude and Longitude
 - Patient Id: Patient Id is Unique.
- The two key classes from the java.net package used in creation of server and client programs are:
 - -> ServerSocket
 - -> Socket

Email/SMS Notification

- >Implemented using JAVAX Mail
- ➤ Authentication with SMTP Mail
- Server sends Email/SMS using SSL connection to the mail server which is delivered to the staff.





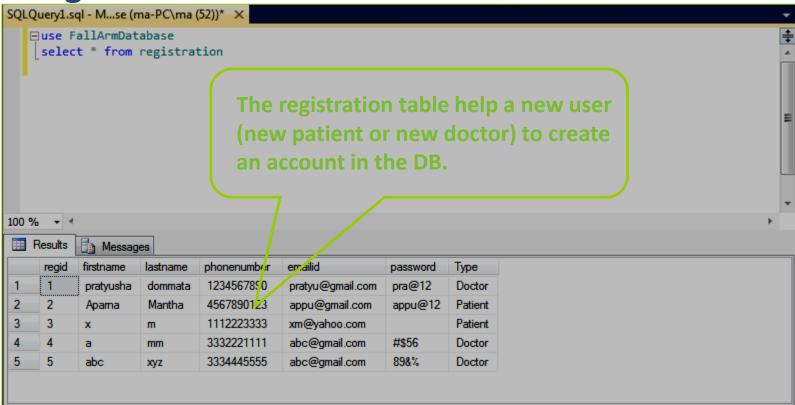


DATABASE & JDBC

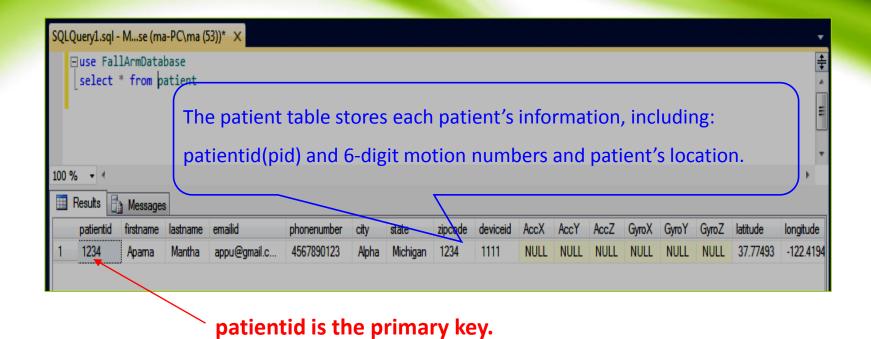
Fall-Arm Database

There are 3 tables in the FallArm Database:

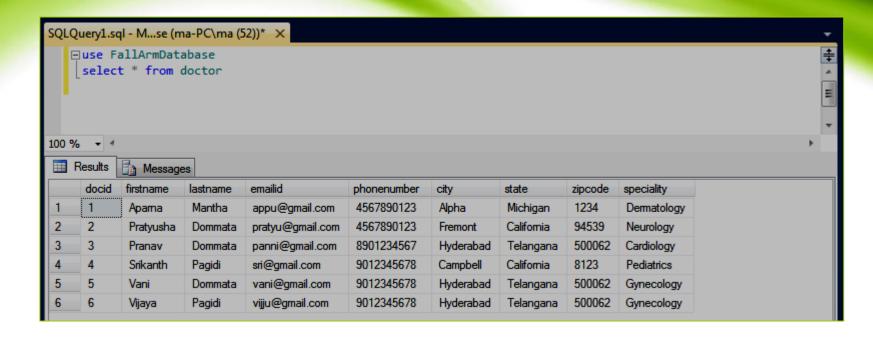
1. registration table



2. patient table



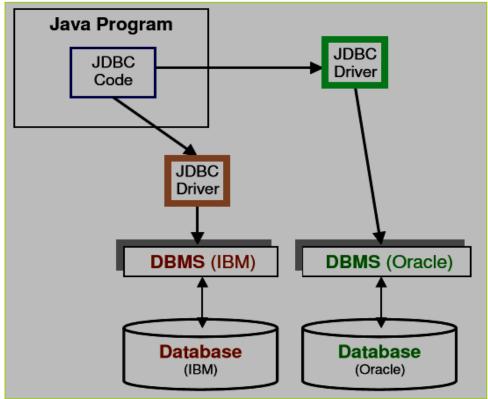
3. doctor table



The doctor table records each doctor's information.

JDBC

1. Database provider provides a JDBC driver that acts as an interface between the JDBC function calls and the DBMS system.



2. Programming with JDBC

- Load the JDBC Driver for your DBMS: Class.forName("com. mysql.jdbc.Driver");
- Obtain a connection to the driver:
 DriverManager.getConnection(<database-URL>) throws SQLException
- Create a statement: A statement object will send your SQL statement to the database and return the results to you.

Statement queryStmt = connection.createStatement();

- > Execute a statement
 - A DDL (Data Definition Language) statement can be executed with:
 executeUpdate (String sqlStmt);
 - A DML (Data Manipulation Language) statement can be executed with:
 ResultSet executeQuery (String sqlQuery);

Process the ResultSet:

You can use the next() method of the ResultSet to move you through each row of the result set table

To extract data in the table, use some of the following methods:
getString(int collndex) //Return String at the specified column number;
getString(String colName) //Return String at specified column name;
getInt(int collndex) //Use when column data is integer;
getInt(String colName) //Use when column data is integer;
getDouble(int collndex) //Use when column data is double;
getDouble(String colName) //Use when column data is double;
getDate(int collndex) //Use when column data is a date;

Close the ResultSet and the database connection (when done) queryRslt.close(); //close the ResultSet dbConn.close(); //close the database connection

3. Example code

This part of the code describes how to insert data into the registration table of the database FallArmDatabase.

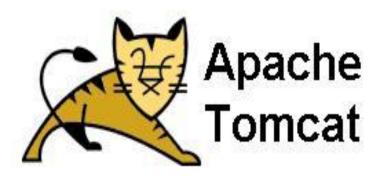
```
J Main.java
     String sql1 = "insert into registration (firstname, lastname, phonenumber, emailid, password, Type) values (?,?,?,?,?)";
     PreparedStatement stmt1 = null;
     try{
         Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriver"); //Load JDBC Driver for the DBMS
         conn = DriverManager.getConnection(url); //Obtain a connection to the driver
         stmt1 = conn.prepareStatement(sql1); //Create a statement
         stmt1.setString(1, firstName);
         stmt1.setString(2, lastName);
         stmt1.setLong(3, phoneNumber);
         stmt1.setString(4, email);
         stmt1.setString(5, passWord);
         stmt1.setString(6, type);
         stmt1.executeUpdate(); //Execute a statement
     }catch(SQLException se){
         se.printStackTrace();
     }catch(Exception e1){
         e1.printStackTrace();
     }finally{
         try{
             if(stmt1 != null)
                 stmt1.close();
         }catch(SQLException se2){
```



Internet Programming

- Our Website MEDICA
- Apache Tomcat Webserver
- Java Servlets
- Java Server Pages









Sample Form

```
<form name="successform"
action="${pageContext.request.contextPath}/Pa
tientDetailsServlet" method="post">
```

```
<fmt:message key="label.enterpid" />:
```

```
<input type="text" name="patientid" >
```

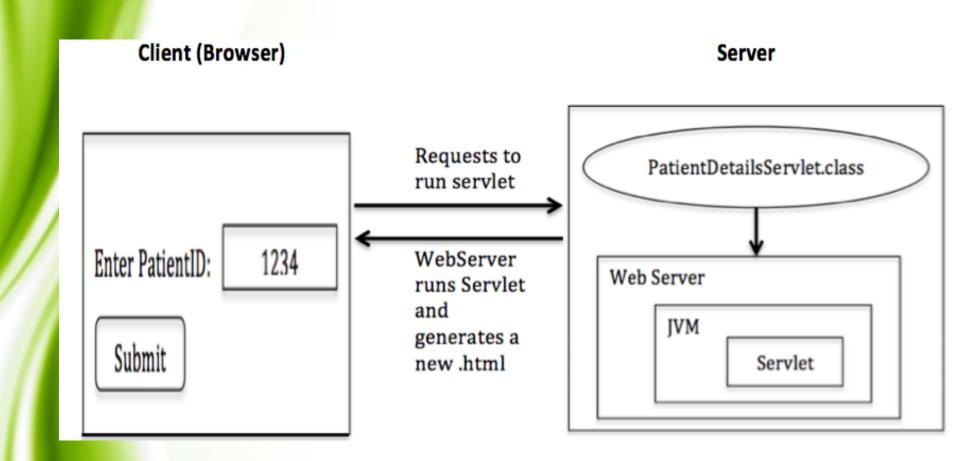
<input type="submit">

</form>

Sample Servlet

```
@WebServlet({ "/PatientDetailsServlet" })
public class PatientDetailsServlet extends HttpServlet{
    protected void doPost(HttpServletRequest request.
            HttpServletResponse response) throws ServletException, IOException [
        // TODO Auto-generated method stub
        HttpSession session = request.getSession(true);
        response.setContentType("text/html");
        try {
            int pid = Integer.parseInt(request.getParameter("patientid"));
            ArrayList<Patient> detailsList = new ArrayList<Patient>():
            detailsList = services.PatientDetails.patientDetails(pid);
            request.setAttribute("detailsList", detailsList):
            session.setAttribute("patient", detailsList.get(0));
            ServletContext context = getServletContext();
            RequestDispatcher dispatch = context
                    .getRequestDispatcher("/views/patientDetails.jsp");
            dispatch.forward(request, response);
            return;
        } catch (Exception e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
```

Server Side Programming



Internet Programming

GET

/FallArmProject/FindDoctorServlet?Speciality=Neurology

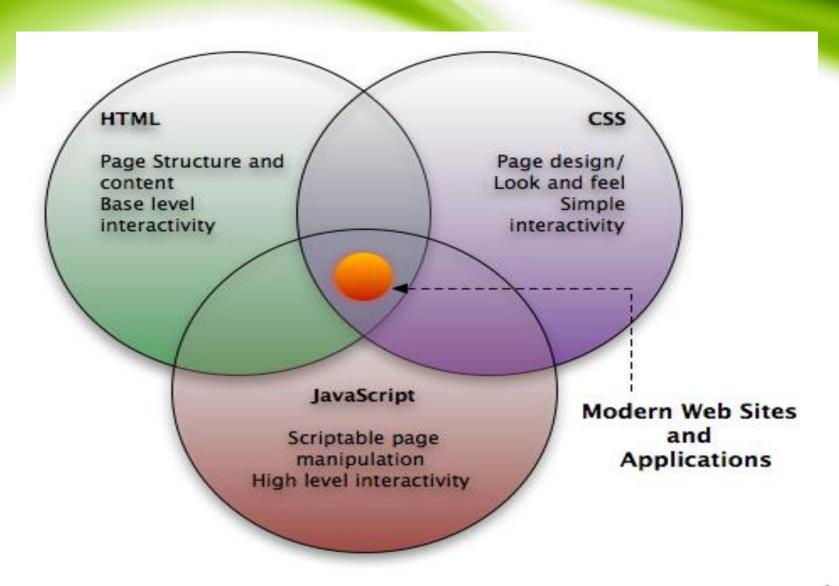
- POST
- Session Tracking

```
session.setAttribute("detailsList", detailsList);
session.getAttribute("detailsList");
```



WEB FRONT-END

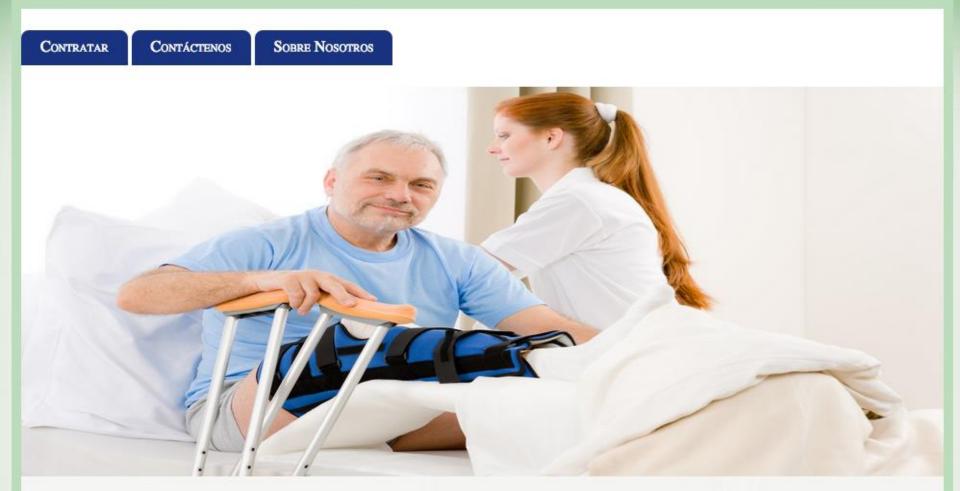
Web Front End



Internationalization – JSTL Tags

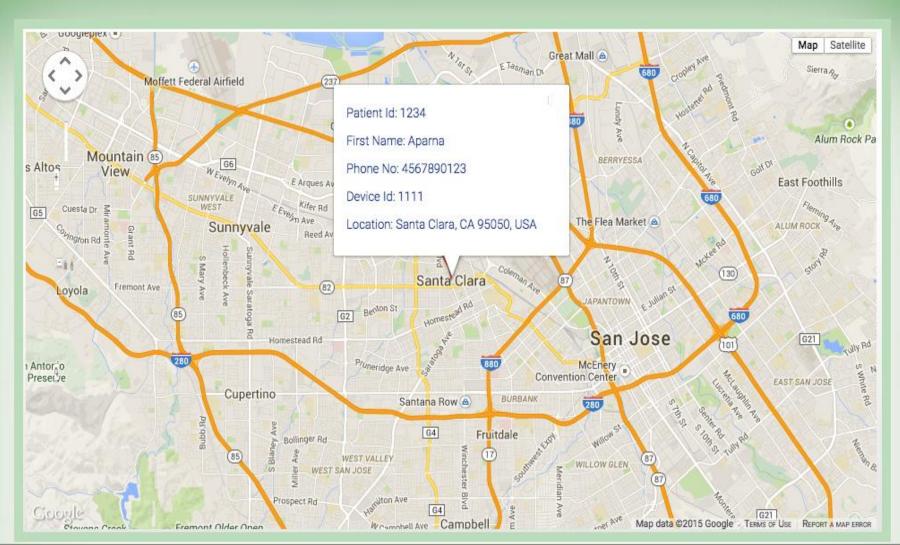


Español ‡



Google Maps







TESTING

QA Testing

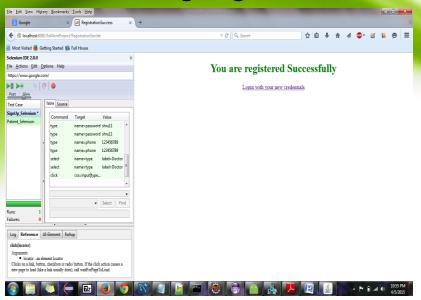
Interface on Web

- Web interface testing using Selenium RC
 - User Login
 - Doctor Details
 - Patient Registration
 - Patient Details
- Web Interface testing using JUnit4 / WebDriver

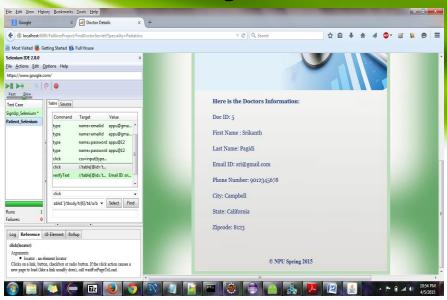
Internet/Database (JUNIT)

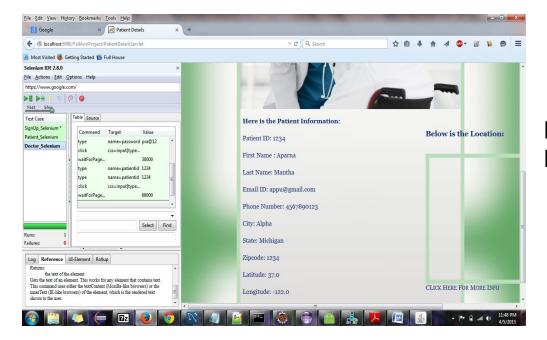
- Database components are tested using Junit 4.
 - Patient name
 - User type validation
 - DB Connection.

Testing Registration



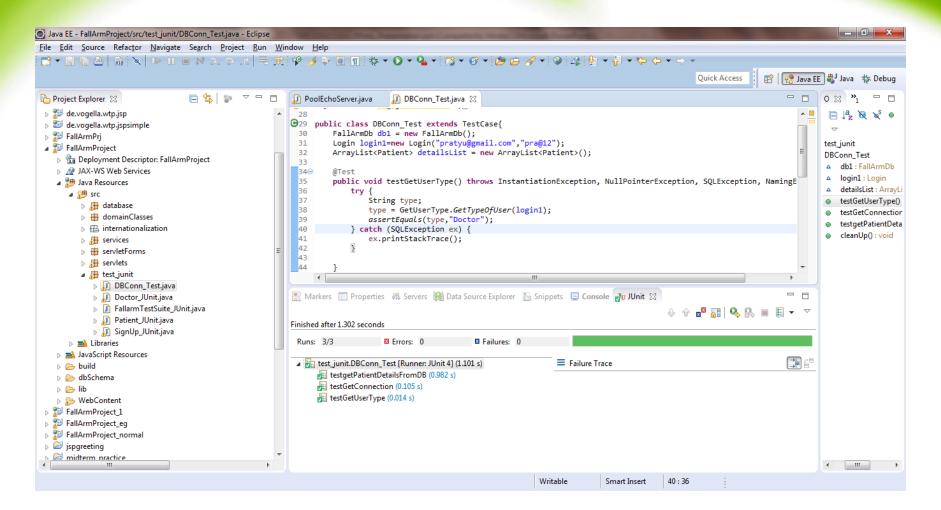
Patient viewing doctor info





Doctor viewing Patient info

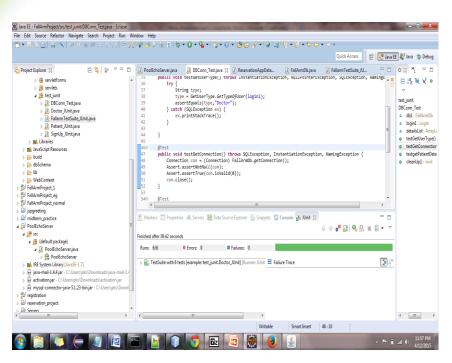
Testing public classes in JUnit

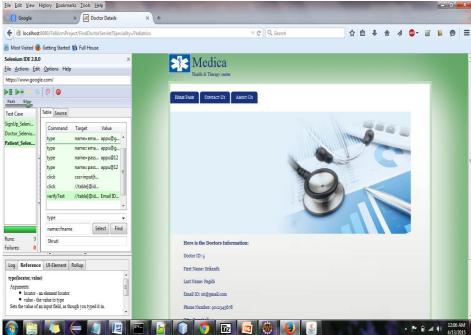


Running as a Test Suite

JUNIT

HTML





Additional Features

- Using Android Smartphone
- > Sending SMS to mobile
- > Google Maps showing patient info and fall location
- > Internationalization
- > Test Automation- Selenium

