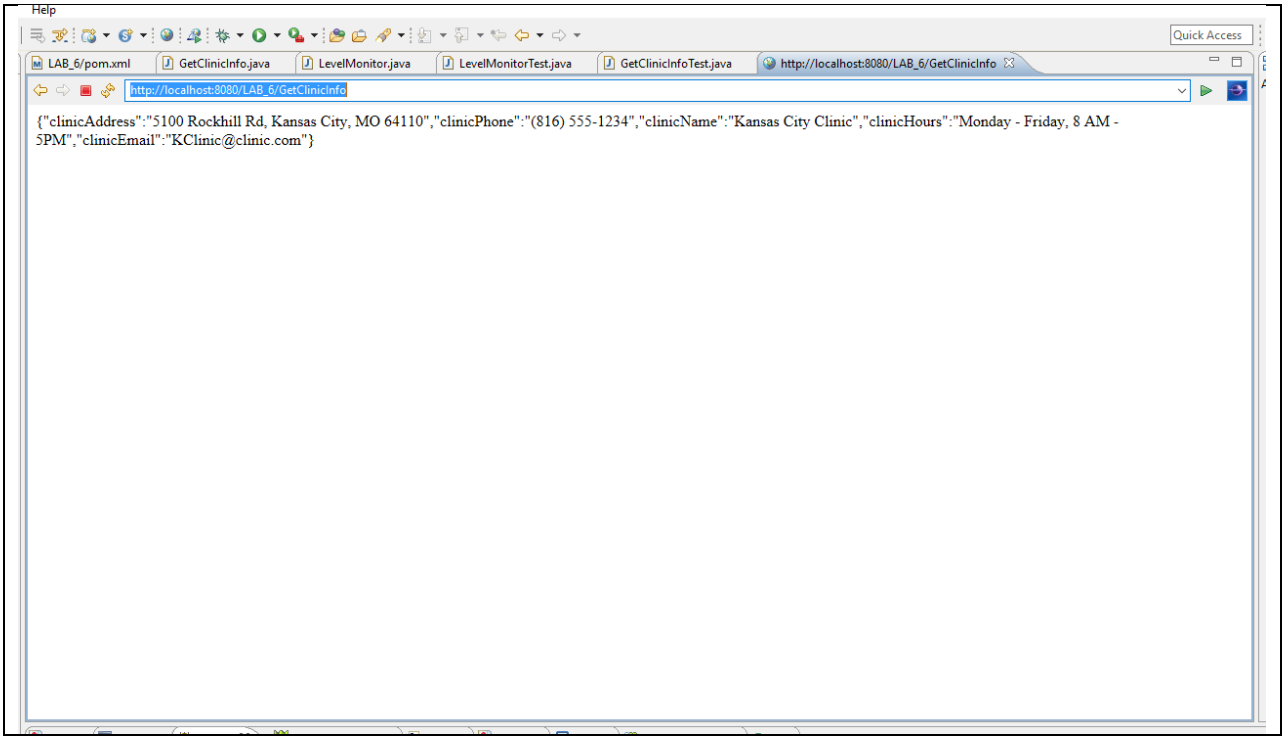
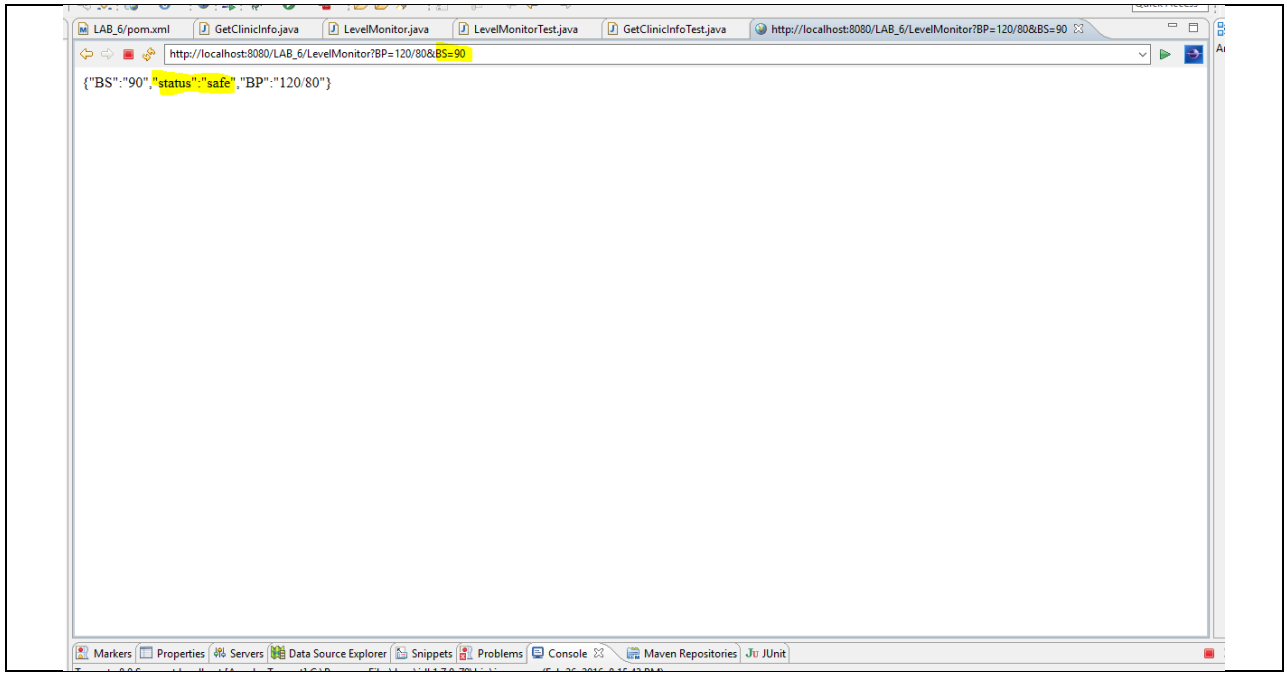
Luke McDuff

Lab 6 – Servlet, Rest Service, Tomcat/Eclipse

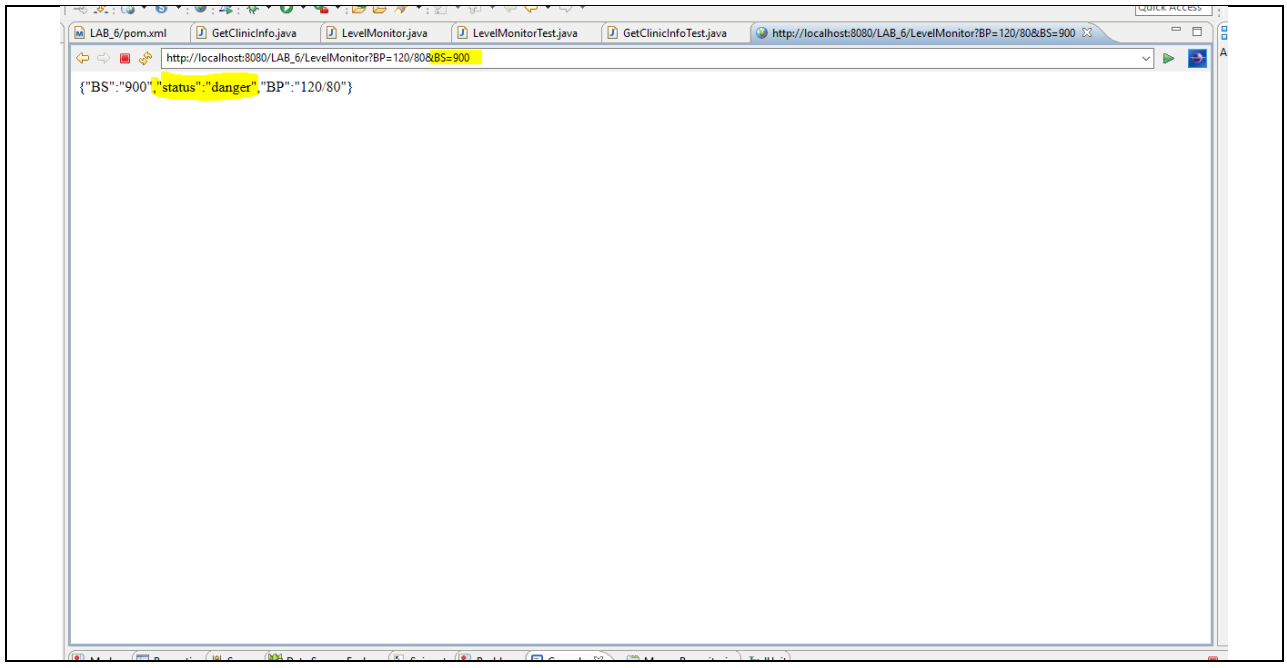
For Lab 6 I created two REST services that will be used for my team's project. The first service returns a JSON string of information about the clinic. This will be request by the mobile clients. This will allow for easy changes to the clinic information by the administrators.



For the second REST service I created a service that receives a POST that contains blood pressure and blood sugar levels. Part of the functionality of the app will contain a log that at-risk patients can track their vitals and save it to the server. When the server saves the info it will check it first. The service will always send a response JSON string, however if any of the levels are very high or very low, a warning flag will be set in the JSON string. This will trigger an alert on the phone for the patient to call the doctor immediately. The first image shows success with good levels, the second image show the warning flag set because of high blood sugar.



Below: with warning set



Unit Tests for LevelMonitor() class servlet, tests the methods that check if Blood Sugar and Blood Pressure is in the correct ranges, determines whether to send a warning back to user

```
public class LevelMonitorTest {

	@Test
	public void isBloodPressureSafeTrueTest() {
		String BP = "120/80";
		LevelMonitor lm = new LevelMonitor();
		assertEquals(true, lm.isBloodPressureSafe(BP));
	}


	@Test
	public void isBloodPressureSafeFalseHighTest() {
		String BP = "145/95";
		LevelMonitor lm = new LevelMonitor();
		assertEquals(false, lm.isBloodPressureSafe(BP));
	}

	@Test
	public void isBloodPressureSafeFalseLowTest() {
		String BP = "85/55";
		LevelMonitor lm = new LevelMonitor();
		assertEquals(false, lm.isBloodPressureSafe(BP));
	}

	@Test
	public void isBloddSugarSafeTrueTest(){
		String BS = "150";
		LevelMonitor lm = new LevelMonitor();
		assertEquals(true, lm.isBloodSugarSafe(BS));
	}

	@Test
	public void isBloddSugarSafeFalseHighTest(){
		String BS = "301";
		LevelMonitor lm = new LevelMonitor();
		assertEquals(false, lm.isBloodSugarSafe(BS));
	}

	@Test
	public void isBloddSugarSafeFalseLowTest(){
		String BS = "69";
		LevelMonitor lm = new LevelMonitor();
		assertEquals(false, lm.isBloodSugarSafe(BS));
	}
}
```

Unit Test for GetClinicInfo() class servlet, tests the method that builds the JSON return object

```
public class GetClinicInfoTest {

	@Test
	public void buildClinicInfoTest() throws JSONException {
	    GetClinicInfo gci = new GetClinicInfo();
	    JSONObject info = gci.buildClinicInfo("loc1","umkc","555-5555","a@a.com","7-8");

	    assertEquals(info.get("clinicName"), "loc1");
	    assertEquals(info.get("clinicAddress"), "umkc");
	    assertEquals(info.get("clinicPhone"), "555-5555");
	    assertEquals(info.get("clinicEmail"), "a@a.com");
	    assertEquals(info.get("clinicHours"), "7-8");
	}

}
```