Group 7: James Clark, Anthony Sommer, Saitejasree Ramala, and James Wehmueller
Dr. Yugyung Lee
CS 590BD: Big Data Analytics
June 24, 2014

Lab 3

1. Build an Android application that provides the following features
- Sensor Data Collection

This is a LogCat log. At the beginning, the sensor is detected. At the end, it is disconnected. In the disconnect event, an HTTP post is sent to Glassfish. Glassfish loads the data into HBase.

```
07-01 11:01:10.768: A/invoke method(7591): onHandleIntent
07-01 11:01:10.768: A/invoke method(7591): startScan
07-01 11:01:10.858: A/invoke method(7591): onLeScan
07-01 11:01:10.858: A/device found(7591): null @ -90
07-01 11:01:10.878: A/invoke method(7591): onLeScan
07-01 11:01:10.878: A/device found(7591): SensorTag @ -47
07-01 11:01:10.908: A/invoke method(7591): onLeScan
07-01 11:01:10.908: A/device found(7591): null @ -82
07-01 11:01:11.188: A/invoke method(7591): onConnectionStateChange
07-01 11:01:11.218: A/invoke method(7591): connectionState
07-01 11:01:11.218: A/connection state change(7591): 0 -> Connected
07-01 11:01:11.298: A/invoke method(7591): onServicesDiscovered
07-01 11:01:11.298: A/invoke method(7591): enableNextSensor
07-01 11:01:11.308: A/enable accelerometer(7591): ACCELEROMETER_CONFIG_CHAR
07-01 11:01:11.368: A/invoke method(7591): onCharacteristicWrite
07-01 11:01:11.368: A/invoke method(7591): readNextSensor
07-01 11:01:11.368: A/read accelerometer(7591): ACCELEROMETER_DATA_CHAR
07-01 11:01:11.409: A/invoke method(7591): onLeScan
07-01 11:01:11.409: A/device found(7591): Flex @ -94
07-01 11:01:11.469: A/invoke method(7591): onCharacteristicRead
07-01 11:01:11.469: A/update the ui(7591): update the ui
07-01 11:01:11.469: A/invoke method(7591): setNotifyNextSensor
07-01 11:01:11.469: A/notify accelerometer(7591): ACCELEROMETER_DATA_CHAR
07-01 11:01:11.509: A/invoke method(7591): onLeScan
07-01 11:01:11.509: A/device found(7591): null @ -75
07-01 11:01:11.529: A/invoke method(7591): onLeScan
07-01 11:01:11.529: A/device found(7591): null @ -92
07-01 11:01:11.559: A/invoke method(7591): onDescriptorWrite
07-01 11:01:11.569: A/invoke method(7591): enableNextSensor
07-01 11:01:11.569: A/enable accelerometer(7591): ACCELEROMETER_PERIOD_CHAR
07-01 11:01:11.659: A/invoke method(7591): onCharacteristicWrite
07-01 11:01:11.659: A/invoke method(7591): readNextSensor
```

```
07-01 11:01:11.669: A/sensors read(7591): mState = 1
07-01 11:01:11.699: A/invoke method(7591): onLeScan
07-01 11:01:11.709: A/device found(7591): null @ -91
07-01 11:01:12.339: A/invoke method(7591): onCharacteristicChanged
07-01 11:01:12.349: A/values(7591): x=-0.3125, y=-1.25, z=0.65625
07-01 11:01:12.349: A/invoke method(7591): SaveData
07-01 11:01:12.540: A/invoke method(7591): onCharacteristicChanged
07-01 11:01:12.540: A/values(7591): x=0.171875, y=-0.25, z=-0.765625
07-01 11:01:12.540: A/invoke method(7591): SaveData
07-01 11:01:12.730: A/invoke method(7591): onCharacteristicChanged
07-01 11:01:12.740: A/values(7591): x=-0.21875, y=-0.65625, z=-1.0625
07-01 11:01:12.740: A/invoke method(7591): SaveData
07-01 11:01:12.930: A/invoke method(7591): onCharacteristicChanged
07-01 11:01:12.930: A/values(7591): x=0.625, y=-0.828125, z=0.03125
07-01 11:01:12.930: A/invoke method(7591): SaveData
07-01 11:01:13.170: A/invoke method(7591): onCharacteristicChanged
07-01 11:01:13.170: A/values(7591): x=-0.171875, y=-0.359375, z=-1.984375
07-01 11:01:13.170: A/invoke method(7591): SaveData
07-01 11:01:13.370: A/invoke method(7591): onCharacteristicChanged
07-01 11:01:13.370: A/values(7591): x=-0.71875, y=-0.609375, z=0.0625
07-01 11:01:13.370: A/invoke method(7591): SaveData
07-01 11:01:13.521: A/invoke method(7591): onLeScan
07-01 11:01:13.531: A/device found(7591): null @ -98
07-01 11:01:13.551: A/invoke method(7591): onCharacteristicChanged
07-01 11:01:13.561: A/values(7591): x=0.453125, y=0.078125, z=0.9375
07-01 11:01:13.561: A/invoke method(7591): SaveData
07-01 11:01:13.751: A/invoke method(7591): onCharacteristicChanged
07-01 11:01:13.751: A/values(7591): x=0.03125, y=0.21875, z=-0.109375
07-01 11:01:13.751: A/invoke method(7591): SaveData
07-01 11:01:13.941: A/invoke method(7591): onCharacteristicChanged
07-01 11:01:13.951: A/values(7591): x=0.75, y=-0.65625, z=-1.09375
07-01 11:01:13.951: A/invoke method(7591): SaveData
07-01 11:01:14.151: A/invoke method(7591): onCharacteristicChanged
07-01 11:01:14.151: A/values(7591): x=-0.484375, y=-0.640625, z=-0.671875
07-01 11:01:14.161: A/invoke method(7591): SaveData
07-01 11:01:14.341: A/invoke method(7591): onCharacteristicChanged
07-01 11:01:14.341: A/values(7591): x=-0.3125, y=-0.484375, z=-1.390625
07-01 11:01:14.341: A/invoke method(7591): SaveData
07-01 11:01:14.532: A/invoke method(7591): onCharacteristicChanged
07-01 11:01:14.542: A/values(7591): x=0.34375, y=1.265625, z=-0.03125
07-01 11:01:14.542: A/invoke method(7591): SaveData
07-01 11:01:14.732: A/invoke method(7591): onCharacteristicChanged
07-01 11:01:14.732: A/values(7591): x=-0.921875, y=-0.1875, z=2.0
07-01 11:01:14.732: A/invoke method(7591): SaveData
07-01 11:01:14.972: A/invoke method(7591): onCharacteristicChanged
07-01 11:01:14.982: A/values(7591): x=-2.0, y=1.109375, z=-0.484375
07-01 11:01:14.982: A/invoke method(7591): SaveData
```

```
07-01 11:01:15.152: A/invoke method(7591): onLeScan
07-01 11:01:15.152: A/device found(7591): null @ -96
07-01 11:01:15.162: A/invoke method(7591): onCharacteristicChanged
07-01 11:01:15.162: A/values(7591): x=-0.53125, y=1.109375, z=0.40625
07-01 11:01:15.172: A/invoke method(7591): SaveData
07-01 11:01:15.342: A/invoke method(7591): onLeScan
07-01 11:01:15.352: A/device found(7591): null @ -97
07-01 11:01:15.362: A/invoke method(7591): onCharacteristicChanged
07-01 11:01:15.372: A/values(7591): x=-1.0, y=-1.203125, z=-0.265625
07-01 11:01:15.372: A/invoke method(7591): SaveData
07-01 11:01:15.553: A/invoke method(7591): onCharacteristicChanged
07-01 11:01:15.553: A/values(7591): x=-0.640625, y=0.359375, z=0.40625
07-01 11:01:15.563: A/invoke method(7591): SaveData
07-01 11:01:15.703: A/invoke method(7591): onLeScan
07-01 11:01:15.713: A/device found(7591): null @ -101
07-01 11:01:15.753: A/invoke method(7591): onCharacteristicChanged
07-01 11:01:15.753: A/values(7591): x=-0.703125, y=-0.734375, z=0.4375
07-01 11:01:15.753: A/invoke method(7591): SaveData
07-01 11:01:15.953: A/invoke method(7591): onCharacteristicChanged
07-01 11:01:15.963: A/values(7591): x=-0.765625, y=-0.515625, z=0.4375
07-01 11:01:15.973: A/invoke method(7591): SaveData
07-01 11:01:15.993: A/invoke method(7591): onConnectionStateChange
07-01 11:01:16.003: A/invoke method(7591): connectionState
07-01 11:01:16.003: A/connection state change(7591): 0 -> Disconnected
```

- File Transfer (using Web Service) (Refer to Tutorial 5)

This step is not functioning correctly. I'm not really sure what the path to a file in Linux.
This does not work:
glassfishIP:8080/group.seven/rest/hbase/a0_accelerometer/insert-sensor-txt//Documents/sensor.txt

- HBase CRUD operations (Create, Retrieve, Update, Delete) (Using Web Services)

These are fully implemented

```
package group.seven.hbase;

import java.io.FileNotFoundException;
import java.io.IOException;

import javax.ws.rs.DELETE;
import javax.ws.rs.GET;
import javax.ws.rs.POST;
import javax.ws.rs.PUT;
import javax.ws.rs.Path;
```

```
import javax.ws.rs.PathParam;
import javax.ws.rs.Produces;
import javax.ws.rs.core.MediaType;

import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.hbase.HBaseConfiguration;
import org.apache.hadoop.hbase.HColumnDescriptor;
import org.apache.hadoop.hbase.HTableDescriptor;
import org.apache.hadoop.hbase.KeyValue;
import org.apache.hadoop.hbase.MasterNotRunningException;
import org.apache.hadoop.hbase.ZooKeeperConnectionException;
import org.apache.hadoop.hbase.client.HBaseAdmin;
import org.apache.hadoop.hbase.client.HTable;
import org.apache.hadoop.hbase.client.Put;
import org.apache.hadoop.hbase.client.Result;
import org.apache.hadoop.hbase.client.ResultScanner;
import org.apache.hadoop.hbase.client.Scan;
import org.apache.hadoop.hbase.util.Bytes;

@Path("/hbase")
public class HBaseService {

    private final String HBASE_ZOOKEEPER_QUORUM_IP = "localhost.localdomain";
    private final String HBASE_ZOOKEEPER_PROPERTY_CLIENTPORT = "2181";
    private final String HBASE_MASTER = HBASE_ZOOKEEPER_QUORUM_IP + ":60010";

    /**
     * CREATE TABLE
     * http://localhost.localdomain:8080/goup.seven/rest/hbase/create
     * /tablename/column1:column2:column3:column4:column5
     *
     * @param tablename
     * @param columnFamilies
     * @return
     */
    @PUT
    @Produces(MediaType.APPLICATION_JSON)
    @Path("/create/{tablename:.+}/{columnFamilies:.+}")
    public String createTable(@PathParam("tablename") String tablename,
                    @PathParam("columnFamilies") String columnFamilies) {
            String line = "{'status':'init'}";
            HBaseAdmin hba = null;
            Configuration config = getHBaseConfiguration();
            try {
                    // create a table
                    HTableDescriptor ht = new HTableDescriptor(tablename);
                    // add columns
```

```
                for (String columnFamily : columnFamilies.split(":")) {
                        ht.addFamily(new HColumnDescriptor(columnFamily));
                }
                try { // save the table
                        hba = new HBaseAdmin(config);
                        hba.createTable(ht);
                        line = "{'status':'ok'}";
                } catch (Exception ex) {
                        line = exceptionToJson(ex);
                }
        } finally { // close the connection
                try {
                        hba.close();
                } catch (IOException e) {
                        // do nothing
                }
        }
        return line;
}

/**
 * READ ALL FROM TABLE
 * http://localhost:8080/group.seven/rest/hbase/fetch/tablename
 *
 * @param table
 * @return
 */
@GET
@Produces(MediaType.TEXT_PLAIN)
@Path("/fetch/{tablename:.+}")
public String readAll(@PathParam("tablename") String table) {
        String line = "";
        Configuration config = getHBaseConfiguration();
        try {
                HTable ht = new HTable(config, table);
                Scan s = new Scan();
                ResultScanner ss = ht.getScanner(s);
                for (Result r : ss) {
                        for (KeyValue kv : r.raw()) {
                                line = line + new String(kv.getRow()) + " ";
                                line = line + new String(kv.getFamily()) + ":";
                                line = line + new String(kv.getQualifier()) + " ";
                                line = line + kv.getTimestamp() + " ";
                                line = line + new String(kv.getValue());
                                line = line + "\n\n";
                        }
                }
```

```
		} catch (IOException e) {
			e.printStackTrace();
		}
		return line;
	}

	/**
	 * UPDATE AT QUALIFIER
	 * http://localhost.localdomain:8080/goup.seven/rest/hbase
	 * /insert/tablename/row/family/qualifier
	 *
	 * @param value
	 *		- passed in the header message
	 * @param table
	 * @param row
	 * @param family
	 * @param qualifier
	 * @return
	 */
	@POST
	@Produces(MediaType.APPLICATION_JSON)
	@Path("/insert/{table:.+}/{row:.+}/{family:.+}/{qualifier:.+}")
	public String insertSingle(String value, @PathParam("table") String table,
			@PathParam("row") String row, @PathParam("family") String family,
			@PathParam("qualifier") String qualifier) {

		String line = "{'status':'init'}";
		Configuration config = getHBaseConfiguration();
		HTable ht = null;
		try {
			ht = new HTable(config, table);
			Put put = new Put(Bytes.toBytes(row));
			put.add(Bytes.toBytes(family), Bytes.toBytes(qualifier),
					Bytes.toBytes(value));
			ht.put(put);
			line = "{'status':'ok'}";
		} catch (Exception ex) {
			line = exceptionToJson(ex);
		}
		return line;
	}

	/**
	 * DELETE TABLE
	 *
	 * @param table
	 * @return
```

```
 */
@DELETE
@Produces(MediaType.APPLICATION_JSON)
@Path("/delete/{table:.+}")
public String deleteTable(@PathParam("table") String table) {
	String line = "{'status':'init}";
	try {
		Configuration config = getHBaseConfiguration();
		HBaseAdmin admin = new HBaseAdmin(config);
		admin.disableTable(table);
		admin.deleteTable(table);
		line = "{'status':'ok'}";
	} catch (Exception ex) {
		line = exceptionToJson(ex);
	}
	return line;
}

/**
 * INSERT FILE
 * http://localhost:8080/group.seven/rest/hbase/insert-sensor-txt
 * /ttjj_sensor_txt//path
 *
 * @param tablename
 * @param filepath
 * @param columnFamilies
 * @return
 */
@GET
@Produces("application/json")
@Path("/insert-sensor-txt/{tablename:.+}/{filepath:.+}")
public String insertFile(@PathParam("tablename") String table,
		@PathParam("filepath") String filePath) {

	String line = "{'status':'init'}";
	try {
		HBase.insertSensorsTxt(table, filePath);
		line = "{'status':'ok'}";
	} catch (Exception ex) {
		line = exceptionToJson(ex);
	}
	return line;
}

/**
 * RETURN THE HBASE CONFIGURATION
 *
```

```
     * @return
     */
    private Configuration getHBaseConfiguration() {
            Configuration config = HBaseConfiguration.create();
            config.clear();
            config.set("hbase.zookeeper.quorum", HBASE_ZOOKEEPER_QUORUM_IP);
            config.set("hbase.zookeeper.property.clientPort",
                            HBASE_ZOOKEEPER_PROPERTY_CLIENTPORT);
            config.set("hbase.master", HBASE_MASTER);
            return config;
    }

    /**
     * HANDLE ALL EXCEPTIONS
     *
     * @param ex
     * @return
     */
    private String exceptionToJson(Exception ex) {
            String json = "{'status':'fail','exception':'";
            String exception = "";
            String message = "";
            if (ex instanceof MasterNotRunningException) {
                    exception = MasterNotRunningException.class.toString();
                    message = ex.getMessage();
            } else if (ex instanceof ZooKeeperConnectionException) {
                    exception = ZooKeeperConnectionException.class.toString();
                    message = ex.getMessage();
            } else if (ex instanceof FileNotFoundException) {
                    exception = FileNotFoundException.class.toString();
                    message = ex.getMessage();
            } else if (ex instanceof IOException) {
                    exception = IOException.class.toString();
                    message = ex.getMessage();
            } else if (ex instanceof NullPointerException) {
                    exception = NullPointerException.class.toString();
                    message = ex.getMessage();
            } else {
                    exception = Exception.class.toString();
                    message = ex.getMessage();
            }
            json += exception + "','msg':'" + message + "'}";
            return json;
    }
}
```