TELE 3118: Network Technologies: Mini-project [10 points]

Note 1: This assignment can be done in pairs (i.e. groups of two people); but if so, you are required to inform the lab demonstrators about the pairing by email no later than the end of week 7. Partners for the mini-project need not be from the same lab session. The working code is to be demonstrated in lab during week 10.

Note 2: This assignment can be done in any programming language; however, we strongly recomend that it be done in Python, for two reasons: (a) Python is very easy to learn and is increasingly being used for simple network programs, and (b) Python provides many support libraries for network functions (such as Flask web-server and JSON encoding routines). You are encouraged to use resources (including sample code) from the web that can help you with your project.

This lab introduces you to writing network applications, by exposing you to the socket API for UDP comunication and REST-based APIs for TCP communication. The application you develop will require you to maintain (in-memory) a small database of student marks. The first part of the project requires you to populate this database by querying a server using a UDP socket, while the second part of the project requires you to run a web-server so that a web-browser can query your database over a TCP connection.

Part 1 [5 points]: Populating your database: In this part you will write a UDP client program that sends a message to a UDP server requesting the list of students and their marks. Please use the sample UDP server code - you can run it locally on your machine for testing purposes, and once you have your client working with your local server you can connect to the real UDP server running on 149.171.36.238. Your message to the UDP server should have the following content:

- The command string "studentmarklist" (should fit in exactly in 16 bytes, including a null terminator)

Your registration message should contain nothing more and nothing less than the string above, and should be compatible with the following C structure "ReqMsg_t":

```
typedef struct { char passwd[16]; } ReqMsg_t; /* request message */
```

In response the UDP server will send you back a message containing a list of students and their marks, in the format below:

- The first 4-bytes (unsigned integer, in network order) will indicate how many students are on the list.
- The rest of the mesage will be an array (of at most 50 items) of a structure that includes the following information for each student: 16-byte user-name (character string including a null terminator), followed by a 4-byte mark (unsigned integer, in network order).

The message you receive will be compatible with the following C structure "RespMsg_t":

```
typedef struct { char studentName[16]; unsigned long studentMark; } StudentRec_s; /*
info about each student */
```

```
 typedef struct { unsigned long nStudents; /* number of users (no more than 50) */
StudentRec_t studentRec[50]; /* record for each student */ } RespMsg_t; /* response
message */
```

Part 2 [5 points]: Querying the database using a web-browser: In this part you will develop a web-server that allows the database to be queried using any web-browser. Your web-server will support the following HTTP GET operations:

- "/api/studentlist" should return the list of students as an array, encoded in JSON.
- "/api/studentmark/[student-name]" should return the mark for the student whose name is specified as part of the URL string.

The outputs of the queries above should be viewable using any browser. it is recommended that you use the Python Flask framework for developing your web-server, and Python's JSON encoder library for sending JSON formatted strings.

Bonus [up to 3 points]: Bonus points will be given to students who develop and demonstrate a web front-end for displaying the results of the queries (instead of just text output on the web-browser). It is recommended that the Bootstrap JavaScript front-end framework be used, though other front-end frameworks (e.g. AngularJS) are also acceptable.