Marking guidelines – ECE318

Guidance for demo functionality tests:

- Ask the student to unzip the file she/he submitted originally and run the demo based on that
    - If the software cannot easily run and some path finding needs to be done at first due to bad configurations etc with MySQL, then 5 marks should be automatically removed
    - If the software cannot run at all, then observe the code/architecture/diagrams and the student should get 15 out of 60 for the overall functionality demo. If the implementation is really bad and basic then the student should get only 1 out of 60 for the overall functionality

Checks/tests

1) Assess the time taken for the software to load a large sample:
    a. **Test**: ask for 100 000 samples
        i. If these cannot be loaded and only a smaller portion can then remove 5 marks from f.
            1. If this doesn't work at all, then remove all 10 marks from f.
        ii. If there is no general summary for the sample and just a printout then remove 2 points in f. on the table
        iii. Ask to see only the TCP-based incidents – if not functionality on this, then remove 2 marks
        iv. Ask to see a range of destination ports related to the sample , try 3000<range<5500
            1. If it doesn't work remove 2 marks from g.
        v. Ask to see a specific port , try destination port 443
            1. If it doesn't work, remove 2 marks from g.
        vi. Ask to see from the sample only Android_SMS_Malware
            1. If this functionality doesn't work, remove 2 marks from g.
        vii. Ask for the top 15 Android_Adware → if they only did for the whole sample and not individual class – then remove 3 points in i. on the table
        viii. Ask to insert a new event with incomplete columns (I will provide the label of the event on the day of assessment)
            1. If they cannot insert it → remove 5 marks from j.
            2. If they can insert it but cannot access it → remove 5 marks from j.
            3. If this functionality doesn't exist → remove all marks from j.
2) Check the code where random samples are picked and how the jdbc connection is done – if no jdbc then reduce the marks within class dependencies in Code explanation on based functionalities – out of 10 it should be less than 5 or maximum to 5 if the implementation is overall good
3) Run erroneous input (e.g., instead of 1000, ask for 1051 records) when you first try the software

4) If all is on command prompt and no GUI or there is only very basic GUI (e.g., a single button) – should be less than 5 in GUI-based incident views
5) Ask them how their sorting works for ranking the events and ask them to show you the code
6) Check if they have comments in their code – if they don't remove 2 marks for presentation → this should go under e. in methods
7) In general, you should also consider your own academic judgement on the overall mark;
    a. If for instance, a student has developed it as a web app, then you should approach it with higher tolerance and be more generous since higher effort was placed
    b. If advanced GUI as single desktop application was developed, also be generous