21 May. 2015

Chapt. 3 SENG 422 TA Lab Log Progress Report

TA: Philip B. Alipour

General Log and Notes on lab session 2:

This session started with 9 students and was finalized to have the following groups as agreed between group members: Group 1 consists of 3 students. Group 2, 4 students, and group 3, 2 students (under exceptional circumstances, this group will be evaluated with more leniency as other groups insisted to remain within the same number of group members without adding/receiving any members from group 3).

Notes for the Attending Students on the project part 1 and 2 deliverables:

I have received questions on whether you need to include for your first part deliverable a sequence diagram:

1- In order to evaluate your architecture, comparisons must be made as your system evolves and events occur, hence requiring an **events/scenarios diagram** i.e. a **sequence diagram**. Although this is not mentioned in the first part project requirements, it is needed when events are studied relative to performance concerns and criteria. Such essential requirements are asked of to meet the criteria I have stated in order to gain top marks relevant to all sections of the project document.

2- As you strategize and come up solutions on mainly performance concerns as the system grows, thereby future enhancements/utilities etc. are also questioned in the first part of your project which are key points to stipulate in your report.

3- Each group must submit one report. As I have explained for Group # 2 in this lab, you will be evaluated individually once the report is about to receive final marks, e.g. each person contributing say between 20 to 25% in a group of 4, in order to finalize the grade for the portion assigned out of the total of your lab.

4- For **part 1** you may or may not depending on the progress of your system for part 1 include: A very general sequence diagram explaining the communication without code (hypothetically speaking) pertaining to **events** and **rough estimates** (**not numbers**, **comparative timing**, e.g., **{first time to register > time to sign in > 0 sec.}** in form of bars or timeframes relative to tasks) for each task/event being executed e.g., as shown to you above, just one task takes longer than the other i.e. one vertical bar longer than the next as you know, and so on. This

is for **part 1** (what is expected to see for best case vs. worst case scenarios addressing performance concerns in the given criteria).

Note: If you cannot submit a sequence diagram for part 1, then it is necessary to submit a detailed version for part 2. If you have already submitted and adequately elucidated in part 1, no need to resubmit the same diagram or anything similar for part 2. Just you may comment on the parts where time sequences have been updated or added to your system.

5- For **part 2**, the **detailed version** (more accurate one as code-dependent which is actual) will be OK to include if you deem it fit (see **note** above), then I will mark your sequence diagrams based on both parts as the evolved one compared to the first version (it can change and does not to be absolutely correct as far as it is logical and part of your discussion).

6- Communication diagrams in addition can also be used representing the events as far as they correspond to time values: comparative values not exactly **x** seconds unless you have them all generated based on code,... just again expected to see one task takes longer than the other starting from second 0, upto the time where an interruption occurs, **concurrency**, **synchronicity asynchronicity** of events occur as you can point them out between services and your system.

a. For example, a server communicates through a request made by your system, and at some point, events (internal as well as external over a communication channel between ends i.e. DBs, memories and CPUs) happen on both sides which takes time. On the other had, if another service is communicating based on a concurrent request by another user, there is an asynchronized time and level of communication (event). What will happen to your data request as a task to process and respond to... and which task/request is prioritized and why should it be prioritized compared to the other request? This is important when the number of tasks increase simultaneously and thread management to address overhead communication becomes a problem to consider (see e.g., **sec. 3.6.1** of the document)...

Summary:

- **Stick to the minimum requirements of the project as the project document was from last year and it will be updated as I will require something that is more reasonable than an unclear description given on the necessary deliverables. I have requested and will use a completed project sample (with minimum standards met already) as my indicator which I haven't currently got any!** To this account, I have

forwarded a request from the main course instructors on the quality of the deliverables as an indicator to assess your submissions! The rest is extra once you meet the minimum requirements (project outline), and as far as you fulfill the documentation you are in the clear.

- Note that, I must reiterate and emphasis further on performance concerns which will always remain in software architecture evaluation process for sure 100% i.e. according to http://www.viewpoints-and-perspectives.info/home/perspectives/performance-and-scalability/ and as far as you convince in your designs as clearly wanted such as discussions/analysis on specific issues, you are OK.
- I am not asking anything out of the scope of the deliverables (except for bonus marks in order to compensate any potential marks you lose in different areas, this is when your system evolves such as its future architectural outcomes), thus giving you the incentive to discuss matters and establish a clear argument behind your rational of e.g., your **design pattern** or anything you are proposing to be logical to mitigate a set of events that occur within this system. This is relevant to software architecture qualities and conflicts between them, and it is to make sure you got it right as a partial fulfillment of your course requirements.
- The verbal discussions is for you to argue your choices and design once you add or remove a component relative to the overall system performance/efficiency prior to implementation (**what will be the impact once done, and the consequences of your design decision?**). This is not a programming course nonetheless, but the application of code onto micro and macro levels of systems, concerns that will arise as outlined above and have to be addressed once they have impact on your code expectably. This could simply be a hypothetical argument as your design expands.

Keep up the good work!

Cheers,

Philip

==========

Philip B. Alipour,
Ph.D. Candidate in Electrical, Computer Engineering and Quantum Physics,
Dept. of Electrical and Computer Engineering, University of Victoria, V8W 3P6, Canada
Office: ELW Room # A358,
Homepage: http://web.uvic.ca/~phibal12/