BSPro - A First Bachelor Semester Project in BiCS-land

Tuesday 6th December, 2022 - 14:45

Motivated Student
University of Luxembourg
Email: motivated.student.001@student.uni.lu

This report has been produced under the supervision of:
Motivated Tutor
University of Luxembourg
Email: motivated.tutor@uni.lu

Abstract

This document is a template for the scientific and technical (S&T for short) report that is to be delivered by any BiCS student at the end of each Bachelor Semester Project (BSP). The Latex source files are available at: ***https://github.com/nicolasguelfi/lu.uni.course.bics.global***

This template is to be used using the Latex document preparation system or using any document preparation system. The whole document should be in 8000 words (± 20%) [1] *for S2 to S6 students and 6000 (± 20%) for S1 students (excluding the annexes) and the proportions must be preserved. The other documents to be delivered (summaries, ...) should have their format adapted from this template.*

A tutor (or any person having contributed to the BSP work) is not a co-author per se for a student's work. It is possible to exploit a BSP report to produce a scientific and technical publication. In this case, the authors list has to be discussed and agreed with the concerned parties.

1. Plagiarism statement

This 350 words section without this first paragraph must be included in the submitted report and placed after the conclusion. This section is not counting in the total words quantity.

I declare that I am aware of the following facts:

- As a student at the University of Luxembourg I must respect the rules of intellectual honesty, in particular not to resort to plagiarism, fraud or any other method that is illegal or contrary to scientific integrity.
- My report will be checked for plagiarism and if the plagiarism check is positive, an internal procedure will be started by my tutor. I am advised to request a precheck by my tutor to avoid any issue.
- As declared in the assessment procedure of the University of Luxembourg, plagiarism is committed whenever the source of information used in an assignment, research report, paper or otherwise published/circulated piece of work is not properly acknowledged. In other words, plagiarism is the passing off as one's own the words, ideas or work of another person, without attribution to the author. The omission of such proper acknowledgement amounts to claiming authorship for the work of another person. Plagiarism is committed regardless of the language of the original work used. Plagiarism can be deliberate or accidental. Instances of plagiarism include, but are not limited to:

1) Not putting quotation marks around a quote from another person's work
2) Pretending to paraphrase while in fact quoting
3) Citing incorrectly or incompletely
4) Failing to cite the source of a quoted or paraphrased work
5) Copying/reproducing sections of another person's work without acknowledging the source
6) Paraphrasing another person's work without acknowledging the source
7) Having another person write/author a work for oneself and submitting/publishing it (with permission, with or without compensation) in one's own name ('ghost-writing')
8) Using another person's unpublished work without attribution and permission ('stealing')
9) Presenting a piece of work as one's own that contains a high proportion of quoted/copied or paraphrased text (images, graphs, etc.), even if adequately referenced

Auto- or self-plagiarism, that is the reproduction of (portions of a) text previously written by the author without citing that text, i.e. passing previously authored text as new, may be regarded as fraud if deemed sufficiently severe.

[1] i.e. approximately 16 pages double columns excluding the Plagiarism Statement

2. Introduction (± 5% of total words)

This paper presents the bachelor semester project made by Motivated Student together with Motivated Tutor as his motivated tutor. It presents the scientific and technical dimensions of the work done. All the words written here have been newly created by the authors and if some sequence of words or any graphic information created by others are included then it is explicitly indicated the original reference to the work reused.

This report separates explicitly the scientific work from the technical one. In deed each BSP must cover those two dimensions with a constrained balance (cf. [BiCS(2021)]). Thus it is up to the Motivated Tutor and Motivated Student to ensure that the deliverables belonging to each dimension are clearly stated. As an example, a project whose title would be "PLAYTOUCH - A multi-user game for multi-touch devices" could define the following deliverables:

- Possible scientific [Armstrong and Green(2017)] deliverables:
  - What are concurrency models and how are they implemented?
  - How is measured ergonomics in human-computer interaction?
  - How to model the concurrency of a multi-touch devices?
  - Can PLAYTOUCH enter in a blocking state?
  - How to model the design of PLAYTOUCH?
- Possible Technical deliverables:
  - PLAYTOUCH Implementation
  - PLAYTOUCH Tests implementation
  - Hardware end system configuration for PLAYTOUCH

The length of the report should be from 8000 words (± 20%) for S2 to S6 students and 6000 (± 20%) for S1 students excluding images and annexes. The sections presenting the technical and scientific deliverables represent ± 80% of total words of the report.

3. Project description (± 10% of total words)

3.1. Domains

3.1.1. Scientific . Provide a short description of the scientific domain(s) in which the project is being made.

3.1.2. Technical. Provide a short description of the technical domain(s) in which the project is being made.

3.2. Targeted Deliverables

3.2.1. Scientific deliverables. Provide a synthetic and abstract description of the scientific deliverables that have been produced. Each BSP must contain some work done according to the principles of the scientific method. It basically means that you should define at least one question related to the knowledge domain of your BSP and follow part of the scientific method process to answer this question. The description of the work done to answer this question is a scientific deliverable.

Other examples of question could be:

- Is Python an adequate language for concurrent programs?
- How can we measure the ergonomics of a graphical user interface?
- How can we ensure that a program will not fail?

An answer to such question should be the result of applying partly or totally the scientific method according to its standard definition which can be found in the literature.

As you can see in this template, the scientific deliverable is entirely separated from the technical deliverable. It the default case it addresses a question closely related to the technical deliverable.

3.2.2. Technical deliverables. Provide a synthetic and abstract description of the technical deliverables that were targeted to be produced. A technical deliverable in this report is the description of a product build by the student using software or hardware technologies.

4. Pre-requisites ($[5\%..10\%]$ of total words)

Describe in these sections the main scientific and technical knowledge that is required to be known by you before starting the project. Do not describe in details this knowledge but only abstractly. All the content of this section shall not be used, even partially, in the deliverable sections. It is important not to include in this section all the knowledge you have been obliged to acquire in order to produce the deliverable. It should only state the knowledge the student possessed before starting the project and that was mandatory to possess to be capable to produce the deliverables. It explicitly defines the technical and scientific pre-condition for the project. It is also useful to avoid project failures due to over or under complex subjects.

4.1. Scientific pre-requisites

4.2. Technical pre-requisites

5. A Scientific Deliverable 1

For each scientific deliverable targeted in section 3
2 provide a full section with all the subsections described below.

5.1. Requirements (± 15% of section's words)

Describe here all the properties that characterize the deliverables you produced. It should describe, for each main deliverable, what are the expected functional and non functional properties of the deliverables, who are the actors exploiting the deliverables. It is expected that you have at least one scientific deliverable (e.g. “Scientific presentation of the Python programming language”, “State of the art on quality models for human computer interaction”, ...) and one technical deliverable (e.g. “BSProSoft - A python/django website for IT job offers retrieval and analysis”, . . .).

5.2. Design (± 30% of section’s words)

Provide the necessary and most useful explanations on how those deliverables have been produced.

5.3. Production (± 40% of section’s words)

5.3.1. What are the most useful criteria for designing a Web Development Library (WDL)?. This scientific deliverable aims to provide a reference of the most useful criteria for designing a Web Development Library (WDL). It successfully classifies the analyzed criteria by justifying the Usefulness Level (USL) of each quality criteria presented in the following paragraphs.

Usefulness Level (USL). The Usefulness Level (USL) is the tool used in this scientific deliverable to classify the different quality criteria of which there is provided a reference. They are called levels because of the different and distinct quantities of Usefulness characterizing the elements to which they are mapped that they express. Each quality criteria is mapped by the Usefulness Classification Function (USCF) onto a set of distinct Usefulness Levels:

- **Not required**: The quality criteria mapped by USCF to this USL does not improve the usefulness of any instance of WDL design problems.
- **Desired**: The quality criteria mapped by USCF to this USL improves the quality of WDL design, but it can be disregarded in every instance of WDL design problems.
- **Important**: The quality criteria mapped by USCF to this USL improves the quality of WDL design, but it can be disregarded in particular edge cases of WDL design problems.
- **Critical**: The quality criteria mapped by USCF to this USL must be followed in every possible instance of WDL design problems.

Functional Suitability. Functional stability is the degree to which a product or system provides functions that meet stated and implied needs when used under specified conditions. Functional suitability is relevant in the context of WDL because of the way in which WDLs are designed to address generalized development principles. Functional suitability is especially useful because the specified conditions of a functionally suitable library are defined by the end goal of the utilizing software. The end software could be of any form and any kind. Therefore, the developers must follow very precise and abstract guidelines for defining the functions of the library in order to satisfy the needs of higher-order design in the process of Web Development, through the libraries themselves. However, the degree of specificity of functionality can vary as the scope of the goals of the product vary, even if in minimal part. These are the reasons the Quality Criteria of functional suitability has an evaluation of: USF 9.5.

Performance Efficiency. Performance Efficiency is the performance relative to the amount of resources used under stated conditions.
Performance Efficiency is relevant in the context of WDL because of the classic architecture of a Web App, involving a client and a server. Performance Efficiency is especially useful when the library is designed for data-intensive applications. By its very definition, a library is a set of behaviors and data structures that can be used inside of a third-part implementation. This implies that in order to obtain a higher degree of performance efficiency, the library itself must have it, and it does so by providing the correct behavior in the specified amount of time, and with the specified amount of resources. It is useful to develop with this quality criteria in mind because of the vast applications of such technologies in the real world which require a lot of data. However, in most cases, on the Web, it is acceptable to not completely adhere to the principle of efficiency because Performance Efficiency is inversely proportional to complexity of codebase. Therefore, it decreases the usability and maintainability from the developer’s perspective. These are the reasons the Quality Criteria of functional suitability has an evaluation of: USF 6.5.

Compatibility. Compatibility is the degree to which a product, system or component can exchange information with other products, systems or components, and/or perform its required functions, while sharing the same hardware or software environment.
Compatibility is relevant to the context of WDL design, because of the very nature of the infrastructures on which the Web relies: APIs are of paramount importance for communicating with different products, systems, or components. Compatibility is especially useful to WDL design because compatibility features increase the usability of the code base, and, if the compatible technologies are also well- mainained, maintainability of itself. In the dynamic field of Web Development it is of paramount importance. These are the reasons the Quality Criteria of functional suitability has an evaluation of: USF= 9.75.

Usability. Usability is degree to which a product or system can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use.
Usability is relevant to WDL design since it allows intuitive and clear practices for development using the WDL. It is absolutely important in order to provide the developers with: Appropriateness; Recognizability; Learnability; Operability; User-Error Protection (which increases Security) and Accessibility (which increases Usability). Usability is the most useful Quality Criteria, therefore it holds an evaluation of: USF= 10.0.

Reliability. Reliability is the degree to which a system, product or component performs specified functions under specified conditions for a specified period of time.

Reliability is of fundamental usefulness in every possible software development context, expecially given the classic extended architecture of Web applications, which require multiple components to work together, and therefore hold and extremely high standard for the reliability of each component. Since the components themselves depend on the WDL, the WDL must have a strong degree of reliability. This is the reason the Quality Criteria of functional suitability is given an average evaluation of: USF= 10.0.

Maintainability. Maintainability is the degree of effectiveness and efficiency with which a product or system can be modified by the intended maintainers.
Maintainability is relevant in the context of WDL because libraries are inherently subject to time-bound improvement and evolution. Maintainability is especially useful for the design of a WDL since there is a vast plethora of hardware/software platforms on which any give Web project can be developed, and it is the responsibility of the developers to guarantee correct functioning and complete functionality to each one of those afro mentioned platforms. Since the difference in many of these target platforms is the temporal release of the host technologies used (I.e rendering and JS engines), the development team must absolutely ensure a maintainable product that is able to quickly evolve and adapt to all of the options present as target technologies. Maintainability is definitely a useful quality criteria to have, but there is one case in which it does not represent a major requirement for the development of a proper library: The lifecycle of the product is predetermined to be relatively short. In which case, there may be no need to focus on maintainability because the developers would rather focus on the functional suitability of the specific use-case coupled with the specific time-frame. These are the reasons the Quality Criteria of maintainability has an evaluation of: USF 8.5.

Portability. Portability is the degree of effectiveness and efficiency with which a system, product or component can be transferred from one hardware, software or other operational or usage environment to another.
Portability is relevant in the context of WDL because in the field of WD there can be present multiple inter-crossing combinations of technological solutions for a given problem. Portability is especially useful for the design of a WDL since there can be multiple hardware/software platform options in the requirements of the given project, and the ability to utilize the library in the native environment of one's choice is advantageous for testing purposes. Ex. react-native react-dom react. However, portability is not an essential feature condusive to the preliminary segment of the lifecycle of the product, and often times it can be completely disregarded as there are multiple potential external solutions to bring the WDL development on one's hardware/software platform (I.e. Emulation, Virtualization). Therefore it cannot be regarded as of the maximum usefulness. This are the reasons the Quality Criteria of portability has an evaluation of: USF = 5.0.

Rankings.

Functional Suitability	9.5
Performance Efficiency	6.5
Compatibility	9.75
Usability	10.0
Reliability	10.0
Security	5.0
Maintainability	8.5
Portability	5.0

Conclusion.

5.4. Assessment (± 15% of section's words)

Provide any objective elements to assess that your deliverables do or do not satisfy the requirements described above.

6. A Technical Deliverable 1

For each technical deliverable targeted in section 3
2 provide a full section with all the subsections described below. The cumulative volume of all deliverable sections represents 75% of the paper's volume in words. Volumes below are indicated relative the the section.

6.1. Requirements (± 15% of section's words)

cf. section 6 applied to the technical deliverable

6.2. Design (± 30% of section's words)

cf. section 6 applied to the technical deliverable

6.3. Production (± 40% of section's words)

cf. section 6 applied to the technical deliverable

6.4. Assessment (± 15% of section's words)

cf. section 6 applied to the technical deliverable

Acknowledgment

The authors would like to thank the BiCS management and education team for the amazing work done.

7. Conclusion

The conclusion goes here.

References

[BiCS(2021)] BiCS Bachelor Semester Project Report Template. https://github.com/nicolasguelfi/lu.uni.course.bics.global University of Luxembourg, BiCS - Bachelor in Computer Science (2021).

[BiCS(2021)] Bachelor in Computer Science: BiCS Semester Projects Reference Document. Technical report, University of Luxembourg (2021)

[Armstrong and Green(2017)] J Scott Armstrong and Kesten C Green. Guidelines for science: Evidence and checklists. *Scholarly Commons*, pages 1–24, 2017. https://repository.upenn.edu/marketing_papers/181/

8. Appendix

All images and additional material go there.

8.1. Source Code

The following environment shows the correct and mandatory way to insert your code.

Listing 1: Caption example.

```
import numpy as np

def incmatrix(genl1,genl2):
    m = len(genl1)
    n = len(genl2)
    M = None #to become the incidence matrix
    VT = np.zeros((n*m,1), int) #dummy variable

    #compute the bitwise xor matrix
    M1 = bitxormatrix(genl1)
    M2 = np.triu(bitxormatrix(genl2),1)

    for i in range(m-1):
        for j in range(i+1, m):
            [r,c] = np.where(M2 == M1[i,j])
            for k in range(len(r)):
                VT[(i)*n + r[k]] = 1;
                VT[(i)*n + c[k]] = 1;
                VT[(j)*n + r[k]] = 1;
                VT[(j)*n + c[k]] = 1;

                if M is None:
                    M = np.copy(VT)
                else:
                    M = np.concatenate((M, VT), 1)

                VT = np.zeros((n*m,1), int)

    return M
```