

# Design and Implementation of Baccalaureate Grade Processing Web Application for Niger

Ali Hassan Kachalo<sup>1</sup>, Oumarou Abdou Arbi<sup>2\*</sup>, Ibrahim Mamane Souley<sup>3</sup>

<sup>1</sup>Rennes 1 University, Rennes, France

<sup>2</sup>Department of Mathematics and Computer Science, Dan Dicko Dankoulodo University, Maradi, Niger

<sup>3</sup>Department of Mathematics and Computer Science, Abdou Moumouni University, Niamey, Niger

Email: ka\_chaloali@gmail.com, \*oabdouarbi@gmail.com, ibrah\_dz@yahoo.fr

**How to cite this paper:** Kachalo, A.H., Arbi, O.A. and Souley, I.M. (2018) Design and Implementation of Baccalaureate Grade Processing Web Application for Niger. *Journal of Software Engineering and Applications*, 11, 121-128.

<https://doi.org/10.4236/jsea.2018.113007>

**Received:** February 18, 2018

**Accepted:** March 24, 2018

**Published:** March 27, 2018

Copyright © 2018 by authors and Scientific Research Publishing Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

## Abstract

In this work, we have developed a web application for Baccalaureate grades processing in Niger. This online application allows to replace Excel based manual grades processing with fully automated one. Thus, the application helps to minimize the errors caused by the Excel manipulation. It also allows rapid centralization of all results at the exam service level. For the development, we adopted the agile method. This method is very efficient in allowing us to prioritize our real needs and the application is enriched as these needs arise. For the design, we used the UML Modeling language and for the implementation, we used a number of computer tools such as PHP, JavaScript, HTML, JQuery, AJAX, Bootstrap and CSS.

## Keywords

Baccalaureate in Niger, Web Application, Agile Method, UML, PHP, JavaScript, AJAX, JQUERY, The Front-End Bootstrap

## 1. Introduction

The Baccalaureate is the first university degree leading to the completion of secondary studies. It is organized in Niger by a central service through several examination centers.

Ten years ago, the calculation of candidates' scores was done manually using calculators. But for some time, the service in charge of the organization had initiated the use of Excel software in order to speed up the calculation methods. However, many cases of errors occur very often highlighting problems related to the use of Excel software for such calculations. Indeed, Excel has a lot of heaviness. And so, computer beginners remain reluctant about its use.

Thus, at the level of the Faculty of Science and Technology (FST) of the University of Dan Dicko Dankoulodo at Maradi (UDDM) through its department of Mathematics and Computer Science, we have proposed a web application that limits the errors that cause the use of Excel software. It also allows rapid centralization of all results at the exam service level. A web application is a software that can be run via an Internet browser. Many technologies are used to develop these kinds of applications [1] [2] [3].

Unlike a conventional website, usually having an informative or promotional role, a web application is a website that provides its users with online services. Thanks to the Internet, web applications can be accessed from a browser at any time to facilitate the processing and exchange of information with a minimum of equipment. Thus, thanks to the information collected at the level of the service of baccalaureate, we have developed a web application called “Baccalaureate calculator” for an efficient treatment of the deliberations of the baccalaureate exams. For the design we used the UML language [4] [5]. For the implementation we used a number of tools such as: PHP, JavaScript, HTML, JQuery, AJAX, Bootstrap and CSS [6] [7] [8].

## 2. System Analysis

A rigorous analysis led us to identify the possibilities of the system and the needs of the users. We have tried to project these needs into comprehensive and detailed use case diagrams to structure the information discoveries. Use cases determine the expectations of users at the system level and thus allow the modeling of their needs [9].

After identifying the actors, we categorized them according to their system access privileges. An actor represents a role played by someone who interacts with the system. Thus, the grouping by access rights has generated two types of users: administrators and the end users. The administration represents in our case of study the service of baccalaureate and the end users are the presidents of exams across the country.

### 2.1. Use Case

A use case defines the behavior of a system or the semantics of any other entity without revealing its internal structure. Each use case specifies an action sequence, including variants that the entity realizes by interacting with the actors of the other entities [9].

**Figure 1** and **Figure 2** represent respectively two use cases of end user’s and administrator’s requirements.

### 2.2. Data Flow Diagrams (DFDs)

A DFD is a cartographic representation of the flow of information between actors participating in a field of study known as an information system. A field of study is a coherent subset of a company or organization, well defined and

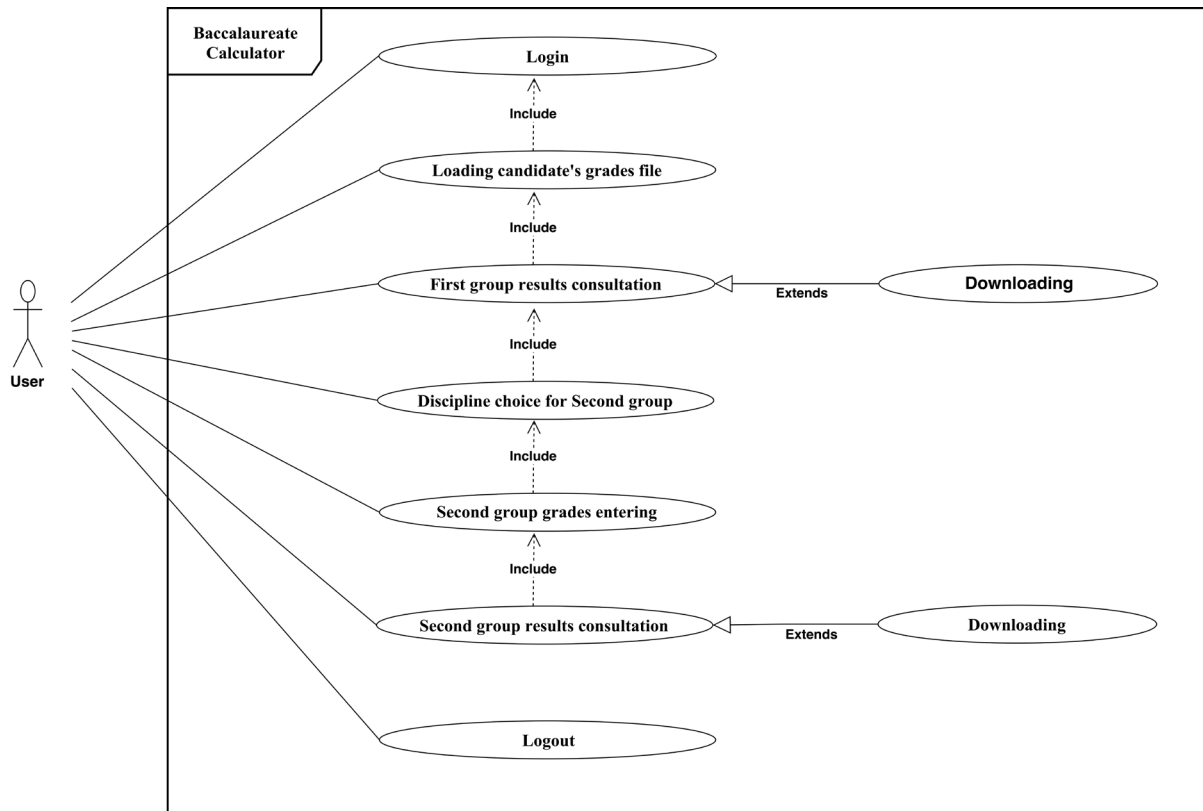


Figure 1. End user use case diagram.

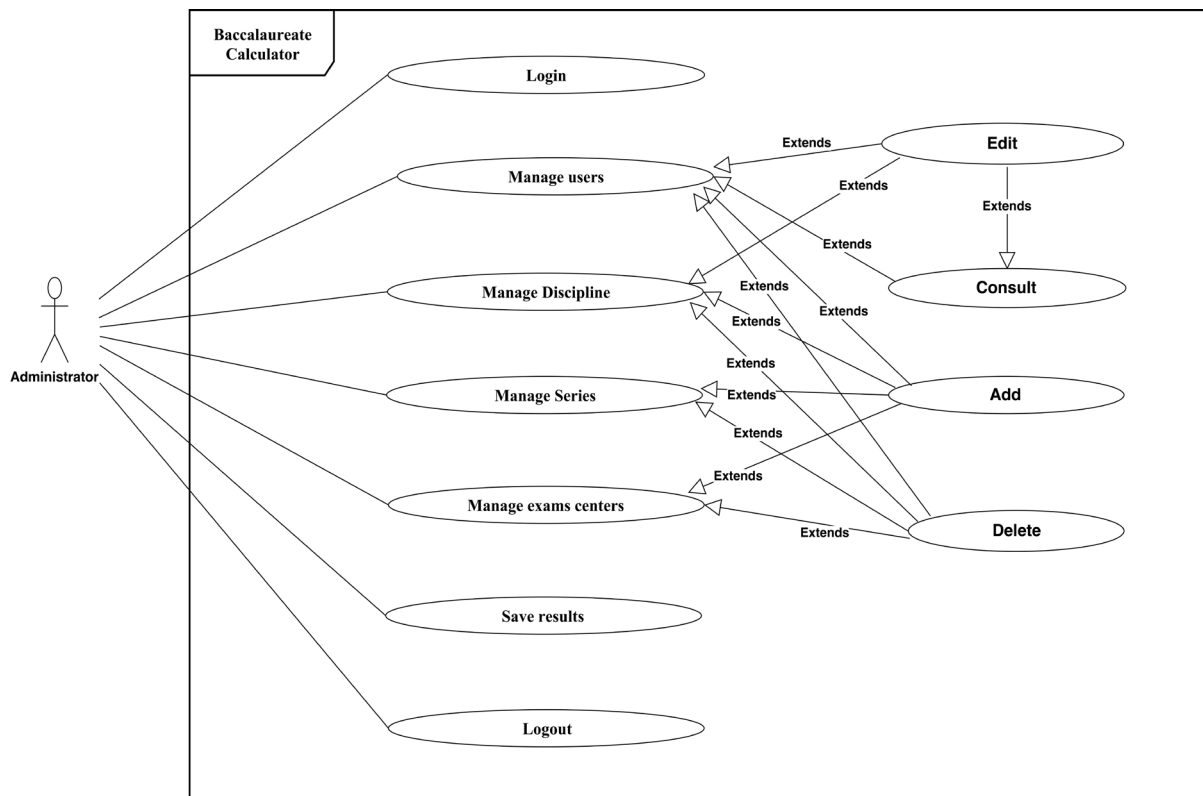


Figure 2. Administrator use case diagram.

forming the content of the subject to study. A DFD describes the main inputs and outputs of the domain as well as the main users [10]. It allows to have an overview of the system to build.

There are two main types of DFD: context-level DFD (level 0), where the study domain is seen as a black box, only flows outside the domain are represented and DFD exploded (level 1), where we detail the activities including flows internal to the field of study [11].

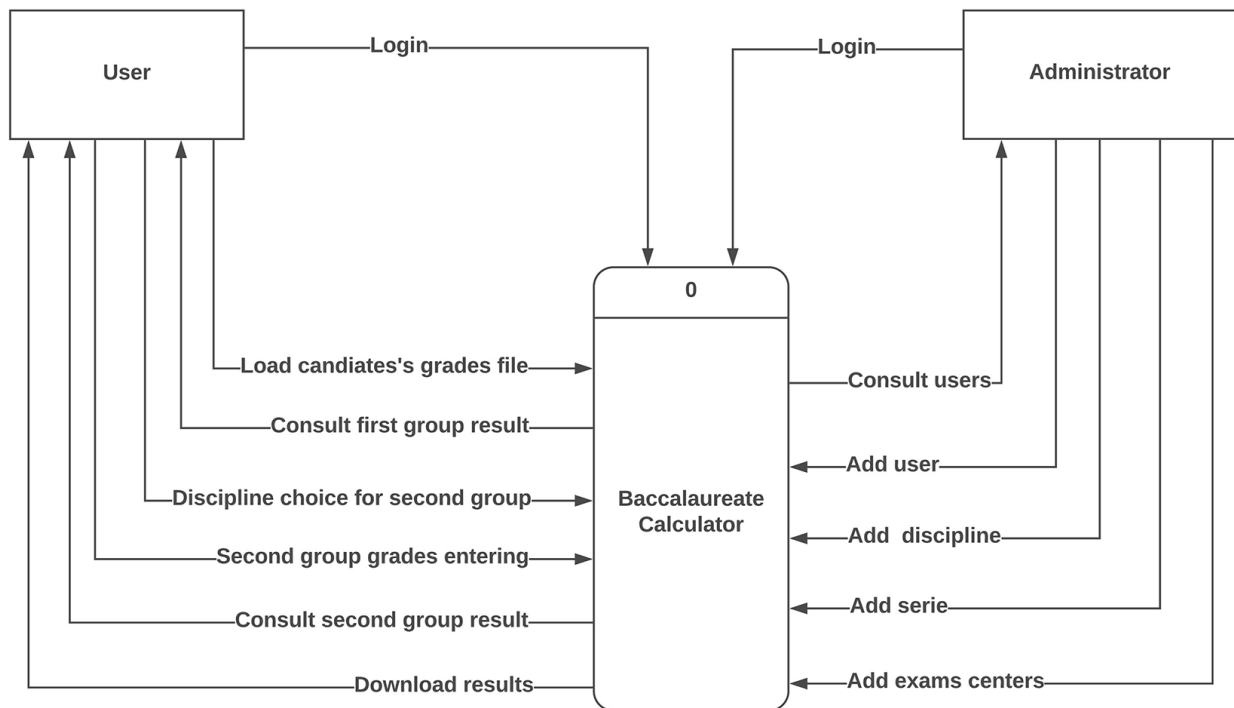
**Figure 3** represents a DFD level 0 (context diagram) for the *Baccalaureate calculator* showing the two main users of the application.

### 3. System Design

The design of a system brings more detail to the proposed solution in the analysis phase and seeks to clarify the technical aspects. Functional requirements and constraints stemming from system requirements are translated into more professional and intelligible language to the team that collaborates in the realization of the system. The essential steps in a design project are: the design of the interface, the technical architecture and the database [12]. In this section, we describe two steps that are most critical to the design of the system. These steps condition the performance and robustness of the application.

#### 3.1. Application Architecture

An architecture is an infrastructure composed of active modules, a mechanism of interaction between these modules and a set of rules that govern this



**Figure 3.** Data Flow Diagram (Context Diagram) for *Baccalaureate calculator*.

interaction [13]. Defining the technical architecture of an application relies on the choice of technologies and organization of components software. The design of the *Baccalaureate calculator* consisted of two independent parts: part of administration and part of end use. Each of these two parts adopts a clear programming style of the Model-View-Controller (MVC) model. The MVC model is a programming style that separates an application into 3 main layers [14]:

The Model Layer: It represents the part of the application responsible for database management, data processing and manipulation [15]. For the *Baccalaureate calculator* application, at the administration part level, the Model layer essentially deals with the data management system stored in the database. As for the end users part, this layer deals with the calculations to be performed and the handling of the files.

The View layer: This layer returns a display of the data that comes from the Model [16]. As any web application, this layer is represented for the *Baccalaureate calculator* by web browsers capable of parsing HTML, CSS, JavaScript, JQuery and interpreting Bootstrap.

The controller layer: This layer manages the requests of the users by introducing the necessary resources (Models & views) for accomplish the desired task [17]. For the case of our application, the controllers are coded scripts using JavaScript, JQUERY and AJAX.

### 3.2. Database Design

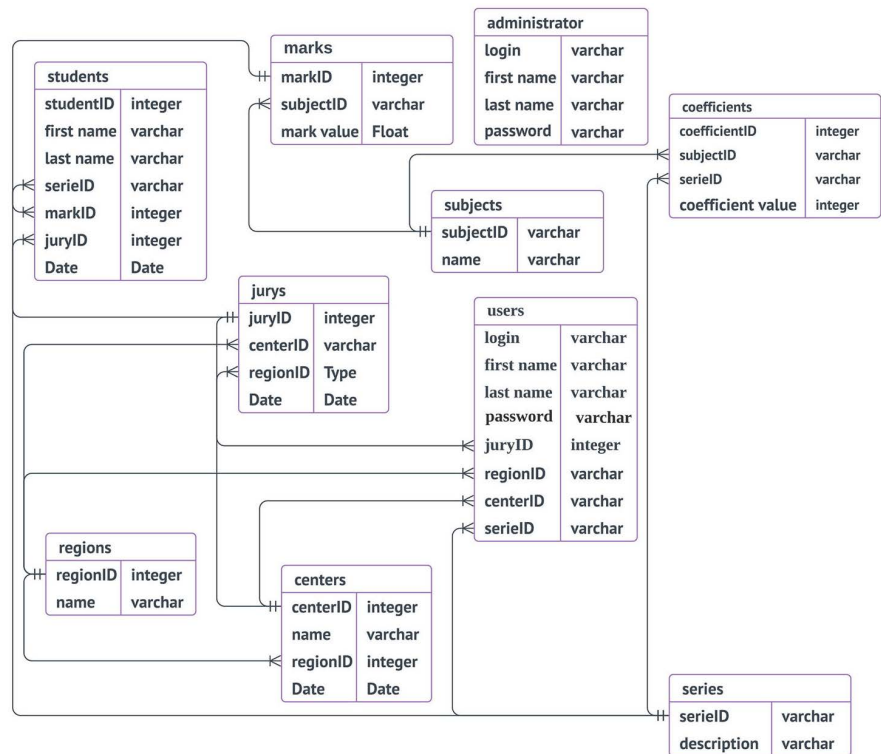
The design process of a database is technical called Entity-Relationship (ER) model. An entity is a table that is essentially characterized by a name and a set of attributes. A relationship is an association that links the entities to each other most often subject to a number of constraints [18]. Relational databases are represented by graphical formalisms called Entity-Relationship diagrams [19]. **Figure 4** is the ER diagram of the database of our application *Baccalaureate calculator*.

## 4. System Implementation

There are usually three indissociable steps when implementing a system: incorporating data (coding), checking their quality (testing), and documentation [20]. We will describe in this section the coding and testing steps for the *Baccalaureate calculator*.

### 4.1. Coding

The implementation is the phase in which we make sure that the system is ready to be exploited by the end users. Thus, to develop the application, we used on the server side, the languages HTML, CSS, Ajax, PHP and SQL. On the client side, we used the JavaScript, JQuery and Bootstrap languages. The LucidChart software environment served as modeling tools. MySQL served database management system, and Apache as web server.



**Figure 4.** Database ER diagram for *Baccaalaureate calculator*.

## 4.2. Testing

Automated functional tests are done throughout this project using the PHPUnit framework, since the implementation of the first functionality. Functional testing is a type of testing which verifies that each function of the software application operates in conformance with the requirement specification. PHPUnit is an open source unit testing framework dedicated to the PHP programming language [21]. The purpose of these tests was to check the quality of the code, to detect any errors that may exist and then to proceed with the correction.

Another test method called integration test was used. In this method, we combined the modules and functions together to examine the collective workings of the modules. We also performed some number of hands-on manual tests, including application access control, user registration, database data modification and update, usability testing in different browsers, and testing to see if the application behaves well in responsive.

## 5. Conclusions

In the organization of the baccaalaureate in Niger, one often finds errors after the deliberations of the examinations in the various centers across the country. Some of these errors are due to mishandling of the excel software. Also, the transcripts of the candidates are established manually. This also causes a lot of errors at this level. To remedy all these problems we developed the web application *Baccaalaureate calculator*.

This online application solved some of the disadvantages of using the Excel software, such as solving computational errors, automatically generating candidate lists and transcript, and statistical results. It also allows rapid centralization of all results at the exam service level.

We have also developed good documentation on how to use and operate the application to help users use it effectively.

## References

- [1] Williams, N.S. (2014) Professional Java for Web Applications. Wrox, Hoboken, NJ, 936 p.
- [2] Japikse, P. (2017) Building Web Applications with Visual Studio 2017: Using .NET Core and Modern JavaScript Frameworks. Apress, New York, 393 p.
- [3] Chris, L. (2018) Progressive Web Application Development by Example: Develop Fast, Reliable and Engaging User Experiences with Progressive Web Applications. Packt Publishing, Luxembourg, 419 p.
- [4] Unhelkar, B. (2017) Software Engineering with UML. Auerbach Publications, New York, 406 p.
- [5] Oestereich, B. (2002) Developing Software with UML: Object-Oriented Analysis and Design in Practice. Addison Wesley, Boston, 320 p.
- [6] Spurlock, J. (2013) Bootstrap. O'Reilly, Sebastopol, CA, 128 p.
- [7] Holdener, A.T. (2008) Ajax: The Definitive Guide. Sebastopol, CA, O'Reilly, 980 p.
- [8] Duckett, J. (2014) JavaScript and JQuery: Interactive Front-End Web Development. John Wiley Sons, Hoboken, NJ, 640 p.
- [9] Schneider, G. and Winters, J.P. (2001) Applying Use Cases: A Practical Guide. Pearson. Education, India.
- [10] Chen, Y.L. (2009) Data Flow Diagram. In: *Modeling and Analysis of Enterprise and Information Systems*, Springer, Berlin, Heidelberg, 85-97.
- [11] Ward, P.T. (1986) The Transformation Schema: An Extension of the Data Flow Diagram to Represent Control and Timing. *IEEE Transactions on Software Engineering*, **2**, 198-210.
- [12] Hoffer, J.A. (2012) Modern Systems Analysis and Design. 6th Edition, Pearson Education, India.
- [13] Boasson, M. and Signaalapparaten, H. (1995) The Artistry of Software Architecture. *IEEE Software*, 13-16 November 1995.
- [14] Selfa, D.M. and Carrillo, M. and Boone, M.D.R. (2006) A Database and Web Application Based on MVC Architecture. *16th International Conference on IEEE Electronics, Communications and Computers, CONIELECOMP 2006*, Puebla, MexR., 1 March-27 February 2006, p. 48.
- [15] Deacon, J. (2009) Model-View-Controller (MVC) Architecture. Online [Citado em: 10 de março de 2006]. <http://www.jdl.co.uk/briefings/MVC.pdf>
- [16] Sanderson, S. (2008) Architecture. Apress, New York, 51-99.
- [17] Leff, A. and Rayfield, J.T. (2001) Web-Application Development Using the Model/View/Controller Design Pattern. *Proceedings Fifth IEEE International Enterprise Distributed Object Computing Conference*, Seattle, WA, 4-7 September 2001, 118-127.
- [18] Isakowitz, T., Stohr, E.A. and Balasubramanian, P. (1995) RMM: A Methodology

for Structured Hypermedia Design. *Communications of the ACM*, **38**, 34-44.

- [19] Thalheim, B. (2013) Entity-Relationship Modeling: Foundations of Database Technology. Springer Science & Business Media, Berlin.
- [20] Tilley, T., Cole, R., Becker, P. and Eklund, P. (2005) A Survey of Formal Concept Analysis Support for Software Engineering Activities. *Formal Concept Analysis*, **3626**, 250-271.
- [21] Bergmann, S. (2018) Manuel PHPUnit.  
<https://phpunit.de/manual/current/en/phpunit-book.pdf>