PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA

الجمهورية الجزائرية الديمقراطية الشعبية

MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH

وزارة التعليم العالي و البحث العلمي

UNIVERSITY OF BISKRA

جامعة محمد خيضر بسكرة

FACULTY OF SCIENCE EXACTES AND SCIENCE OF NATURE AND LIFE

كلية العلوم الدقيقة وعلوم الطبيعة والحياة

DEPARTMENT OF MATHEMATICS AND COMPUTER SCIENCES

قسم الرياضيات و الإعلام الآلي

Master Thesis

Domain Mathematics and Computer Science

Field Computer Science

Option Software Engenieering and Distributed Systems

By:

Mohammed Ilyes Khelil, Mahfoudh Kamel and Akram Khelal

Topic

Web application for municipality budget management
studied case: municipality of djemoura

Defended Publicly in Front of the Jury Composed of

M_r.	H.Bennoui	PRESIDENT	UMKB
M_s.	Z.Hemidi	EXAMINER	UMKB
M_r.	L.KAHLOUL	SUPERVISOR	UMKB
M_s.	A.MOHAMMEDI	SUPERVISOR	UMKB

Academic Year 2022/2023

Dedication

This thesis is dedicated to

- **Mohammed Ilyes Khelil:** Dedicated to my parents, whose unwavering love and support are my driving force. To my family, colleagues, and friends, thank you for your understanding and encouragement throughout my life. Your presence has made the journey more meaningful.
- **Mahfoudh Kamel:** I give special thanks to my dear parents, who are one of the blessings I have and to all the friends who have been with me from a young age and at university, as well as to my teachers, especially
- **Akram Khelal:** This work is dedicated to My beloved Parents who always supported and believed in me, To My brother and sisters and my family , Last but not least, My friends and classmates and everyone I studied with and supported me.

Acknowledgements

We would like to take this opportunity to extend our heartfelt appreciation and gratitude to the following individuals and organizations who have played a significant role in the successful development and completion of this project:

- **Supervisors:** are immensely grateful to our dedicated supervisors, Laid Kahloul and Ameria Mohamedi, for their unwavering guidance, valuable insights, and continuous support throughout the entire duration of this project. Their expertise, encouragement, and constructive feedback have been instrumental in shaping our work and pushing us towards excellence.
- **Professor Meftah Zouai:** We would like to express our sincere thanks to Professor Meftah Zouai for his invaluable assistance and mentorship. His extensive knowledge, enthusiasm, and support have been crucial in introducing us to the Laravel framework and providing us with the necessary guidance to embark on this innovative path. His dedication to our academic growth and commitment to fostering a dynamic learning environment have been truly inspiring.
- **Municipality of Djemoura:** We would like to express our gratitude to the Municipality of Djemoura for providing us with the invaluable opportunity to collaborate and work on this project. Their trust, encouragement, and cooperation have allowed us to gain firsthand experience in real-world municipal budget management and have served

as a driving force behind our motivation to deliver a practical and impactful solution.

- **Startup Committee and Computer Science Department Committee:** We would like to express our sincere appreciation to both the Startup Committee and the Computer Science Department Committee for accepting our graduation project for discussion. We are grateful for the opportunity to present our project before these esteemed committees and for the insightful discussions that followed.
- **Entrepreneurs House of Biskra University:** We would like to extend a special thanks to the Entrepreneurs House of Biskra University for their invaluable support and resources in fostering our entrepreneurial journey. Their guidance, mentorship, and provision of the necessary skills and knowledge related to startups have been instrumental in shaping our project. We are grateful for their commitment to nurturing entrepreneurial talent and for equipping us with the essential skills required to transform our ideas into a reality.
- **All Those Who Contributed:** We extend our gratitude to all individuals who have directly or indirectly contributed to the success of this project. Your support, assistance, and encouragement have been instrumental in overcoming obstacles and achieving our objectives. Whether it was through sharing valuable insights, providing technical expertise, offering words of encouragement, or providing startup-related skills, your contributions have made a significant impact on the outcome of this project.

Thank you all for being an integral part of our journey.

Abstract

Our objective is to develop a municipal budget Manage web application. This application will help users view, track and manage budgets in real time, Budget allocation, expense tracking, and more Reports. The app focuses on user experience, This makes it easily accessible to all users, even those without prior budget Learn how to navigate and use the tool. The web application will contains a dashboard that provides an overview of the budget, including Visualization of budget allocations, expenses and income. this Let users quickly and clearly understand the process marital status. By incorporating a powerful budget optimization tool, our web application enables users to strategically allocate financial resources, track progress towards goals, make transparent decisions, optimize resources.our application also ensures efficient use of funds, informed decision-making, and a stronger sense of financial accountability within the community.

keywords: Develop, Municipal budget manage, Web applications, Users view, real time, User experience.

مُلخّص

هدفنا هو تطوير تطبيق ويب لإدارة الميزانية البلدية. هذا التطبيق لمساعدة المستخدمين على عرض وتتبع وإدارة الميزانيات في الوقت الحقيقي، تخصيص الميزانية، تتبع النفقات و المزيد من التقارير. يركز التطبيق على تجربة المستخدم وهذا يجعله في متناول جميع المستخدمين بسهولة حتى أولئك الذين ليس لديهم ميزانية مسبقة يتعلمون كيفية التنقل واستخدام الأداة. يحتوي تطبيق الويب على لوحة تحكم توفر نظرة عامة على الميزانية، بما في ذلك تصور مخصصات الميزانية والمصروفات والدخل. يتيح ذلك للمستخدمين فهم الحالة الاجتماعية للعملية بسرعة ووضوح. من خلال دمج أداة قوية لتحسين الميزانية، يمكن تطبيق الويب الخاص بنا المستخدمين من تخصيص الموارد المالية بشكل استراتيجي، تتبع التقدم نحو الأهداف، اتخاذ القرارات الشفافة تحسين الموارد، كما يضمن تطبيقنا الاستخدام الفعال للأموال، اتخاذ القرارات المستنيرة و إحساس أقوى بالمساءلة المالية داخل المجتمع.

الكلمات المفتاحيّة: تطوير، إدارة الميزانية البلدية، تطبيقات الويب، واجهة المستخدمين، الوقت الحقيقي، تجربة المستخدم.

Résumé

Notre objectif est de développer une application web pour la gestion du budget municipal. Cette application aide les utilisateurs à visualiser, suivre et gérer les budgets en temps réel, l'allocation budgétaire, le suivi des dépenses et plus de rapports. L'application se concentre sur l'expérience utilisateur, Ce qui la rend facilement accessible à tous les utilisateurs, même ceux qui n'ont pas de budget préalable Apprennent à naviguer et à utiliser l'outil. L'application Web contiendra un tableau de bord qui donne un aperçu du budget, y compris la visualisation des allocations budgétaires, des dépenses et des revenus. cela a permis aux utilisateurs de comprendre rapidement et clairement le processus de l'état matrimonial. En intégrant un puissant outil d'optimisation budgétaire, notre application Web permet aux utilisateurs d'allouer stratégiquement des ressources financières, de suivre les progrès vers les objectifs, de prendre des décisions transparentes, d'optimiser les ressources.notre application garantit également une utilisation efficace des fonds, une prise de décision éclairée et un sens accru de la responsabilité financière au sein de la communauté.

mots-clés: Développement, gestion du budget municipal, applications Web, vue des utilisateurs, temps réel, expérience utilisateur.

Table of Contents

List of Figures

List of Tables

General Introduction

In today's rapidly evolving society, effective budget management is crucial for the sustainable development and progress of municipalities. However, many municipalities face challenges in streamlining their financial processes, ensuring transparency, and engaging their communities. In recognition of these issues, our project endeavors to develop a web application dedicated to municipal budget management. This report outlines the key problems faced by municipalities, presents the objectives of our project, and introduces the innovative solutions our web application provides to address these challenges.

The municipality is considered the basic unit of the people and the state. This definition reflects the many functions of the municipality and its various tasks. However, considering most Algerian municipalities, we find that they rely on manual struggle in their administration in general and the budget in particular. This reflects many disadvantages and problems, including:
-Resource consumption such as (papers, pens).
-Managed more slowly.
-Lack of analysis and overview.
-Limited cooperation and access in Budget Management.
-Increased probability of errors.

And because the study of the existing is one of the most important factors in building science and projects, through our visit to the studied municipality to get acquainted with the way of managing the municipal budget, they have presented to us the budget management system that exists there, which is a desktop application that works on one device.

After a simple experience of the application, we found many disadvantages and shortcomings, including:
-The application can be accessed without authentication, which threatens its confidentiality.
-Difficulty in use which can lead to user frustration, decreased efficiency, and a poor user experience. It can also make a product or service less attractive to potential customers, resulting in a decrease in sales.
-Using the application and budgeting can only be done by one user, which leads to completion delays.
-The application runs on one platform (Windows) which makes it impossible to use it on another platform like Linux or android. This limitation greatly reduces its user base and hinders its adoption.
-The process of municipal budget management is incomplete in the application.

Technologies of information and communication become one of the most evolution field nowadays, such as web applications which has become an integral part of our lives, as well as its users are increasing continuously.

Web applications provide the use of the platform on any device without taking into consideration the possibility of compatibility between them. As well as being able to be used from any place without the need to use a particular one, this can also guarantee that employers are able to add modifications or perform some repairs.

Our project in this report represents the development of a web application with multiplatform that ensures a smooth and effective management of municipality budgets. It is therefore a question of designing and implementing a solution allowing to:
-Budget management

-Export the documents
-General overview
-Preview statistics

This report is organized into four chapters:
-The first chapter is about the municipality accounting department which displays general structure of municipality and information about its budget such as the various documents used in management and the preparation of them.
-The second one contains general definitions of the domain, such as information systems, computer systems and types of software applications. Moreover, it provides information about the web applications.
-The third chapter contains the analysis and the design of the project as well as the different diagrams that were utilized.
-The final one displays the implementation of the project and the different technologies that have been employed.

Chapter I

Municipal Accounting Department

1 Introduction

The budget is not just a theoretically empty framework, rather it is a tool and software used to manage, purchase, and spend resources to provide services to residents.
The Algerian municipal budget is a schedule of anticipated municipal yearly revenues and expenses. It is a licensing and management agreement that permits the handling of municipal affairs and the execution of its investment and equipment program. Through regulation, it establishes the format and substance of the budget.

2 General presentation of the Algerian municipalities

2.1 Brief history

In general, we can divide the stages of municipal development as follows:

- **Municipalities in Algeria at the stage of colonization(1830 - 1962):** The municipality was one of the tools to achieve the colonial administration's ambitions and impose its dominance and influence. It served the European element in general and the French in particular.
  These mixed municipalities were run by a colonial administration official, the civil administrator, assisted by Algerian officials under French administration.

- **Municipalities in transition stage after 1962:** At this stage, the municipality was exposed to the same crisis as the rest of the state institutions due to its difficult financial, technical and human conditions. This made it difficult for the municipality to set and achieve its goals. To address these issues, the municipality implemented several reforms to improve its operations. These reforms included improved financial management and increased investment in human capital.

- **The stage of thinking about the creation of a municipal law:** The Constitution of 1963, the Charter of Algeria and the Charter of Tripoli had a significant impact in highlighting the status of the municipality at the official level and recognizing its role on the scene, for reasons that called for its speedy issuance.

- **The stage of the municipal law of 1967:** This stage witnessed the issuance of Municipal Law No. 24-67 issued on January 18, 1967, and this law was characterized by being influenced by two different models, namely the French and the Yugoslav model.
- **The stage of administrative division of 1974:** Pursuant to this division issued pursuant to order 74 - 69 dated July 02, 1974. Many points were agreed upon, including replacing the term province with state, and the number of provinces increased from 17 to 31 states. The number of municipalities reached 704 from 676 in 1967.
- **The stage of administrative division of 1984:** Municipalities were characterized by political and economic stability until 1984, where old municipalities were divided and new ones were created. In order to bring the administration closer to the citizens and to adhere to the principle of decentralization and independence, the number of municipalities doubled to reach 1541 municipalities.
- **The stage of the municipal law of 1990:** This stage was marked by the submission of new principles and provisions established by the 1989 Constitution, on top of which the abolition of the one-party system and the adoption of multipartyism, and the workers and farmers no longer have any priority in the field of candidacy as before after the abandonment of the socialist system was proven[1].

2.2 General organization of municipality

According to the following organizational chart, we can describe the general structure of the municipality:

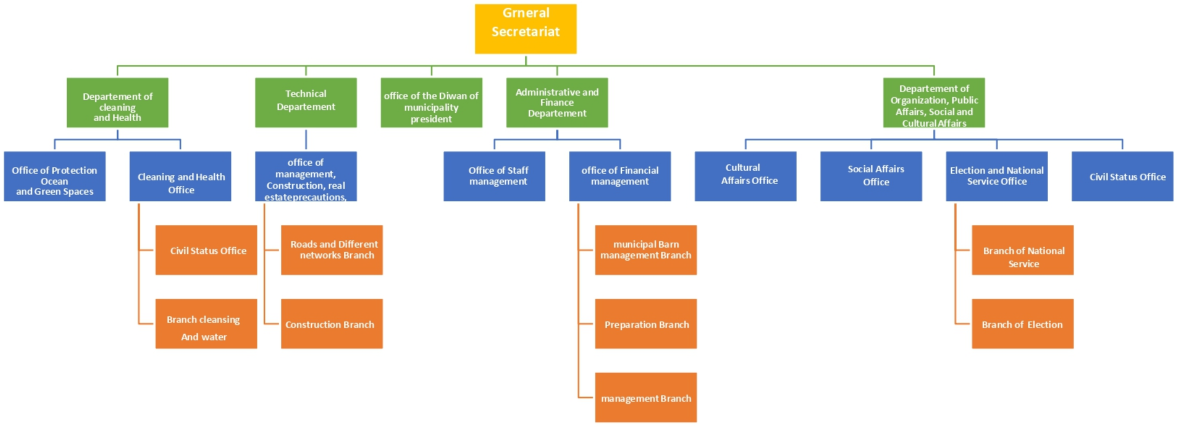

Grneral Secretariat
Departement of cleaning and Health
Technical Departement
office of the Diwan of municipality president
Administrative and Finance Departement
Departement of Organization, Public Affairs, Social and Cultural Affairs
Office of Protection Ocean and Green Spaces
Cleaning and Health Office
office of management, Construction, real estateprecautions,
Office of Staff management
office of Financial management
Cultural Affairs Office
Social Affairs Office
Election and National Service Office
Civil Status Office
Civil Status Office
Branch cleansing And water
Roads and Different networks Branch
Construction Branch
municipal Barn management Branch
Preparation Branch
management Branch
Branch of National Service
Branch of Election

Figure I.1: Municipality Structure.

The Directorate of Finance and Accounting Affairs is an official service responsible for the municipal budget implementation and management. Its tasks include preparing the general budget of the municipality, directing expenses and revenues related to the municipality, conducting financial audits, and submitting financial reports to the relevant authorities.

2.3 Budget execution officers

As officers we can find in that service to execute budget management are Cashiers and Public Accountants.

- **Cashiers:** It is any person who is legally qualified to operate the state's funds or its institutions and public groups. This is whether it is a revenue or spending operation.

- **Public Accountants:** A public accountant is any person legally appointed to perform operations related to the state's finances, whether directly or through other accountants. They are responsible for ensuring that public funds are managed correctly and that the government is able to meet its financial obligations. They are also responsible for auditing businesses and other organizations to ensure compliance with the law.

3 Identifying the various municipal budget documents in Algeria

Five (05) detailed budget papers, each with the following definitions, make up the municipality's budget:

- **The initial budget:** The basic financial record is one that shows estimates of revenues and costs for a fiscal year. on the tentative budget, before the end of the fiscal year previous to the year it is implemented

- **Pre-financial accreditation :** In the event of a need, opening separately certified credits in advance is known as opening credits in advance; if it occurs before the extra budget, the additional budget is determined through discussion.

- **Additional budget :** It is an amended document of expenses and revenues (whether increased or decreased) for the first budget of the current fiscal year based on the results of the previous fiscal year, and it must be voted on before June 15 of the year in which it is implemented; the balance in the administrative account of the previous fiscal year is carried over.

- **Special license:** A special license is used to access credits that must be certified separately if necessary. If it arrives after the additional budget, then any new revenues registered after the additional budget are recorded using a special license and subsequently settled by the administrative account.

- **Administrative account:** At the conclusion of the fiscal year's additional period (March 31), the chairman of the municipal People's Council prepares the municipal administrative account and submits it to the municipal People's Council for approval[1].

4 Preparation of budget documents

The preparation of municipal budget documents takes into account the following:

- The municipal budget is split into two sections: management & equipment and investment.
- Each part is mandatory and is divided into balanced revenues and expenses.
- Deducting from operating income an amount set aside to pay processing and investment costs.
- The initial budget is created before the commencement of the fiscal year, and throughout the fiscal year, modifications are made to expenditures and revenues based on the prior fiscal year's outcomes using a supplementary budget.
- The municipal Secretary General prepares the draft budget under the authority of the chairman of the municipal People's council, and the chairman of the municipal People's council presents the draft budget to the council for approval.
- Before the beginning of the fiscal year when the budget will take effect, the first budget will be subject to voting prior to October 31. Also, the supplementary budget will be voted on before June 15 of the fiscal year it will be implemented in.
- When the municipal budget is executed with a deficit, the municipal council must take all the required measures. If the municipal People's Assembly does not take the necessary corrective measures to ensure the balance of the additional budget, they are taken by the governor to enable

the deficit to be absorbed over two or more years of the fiscal year.

- The municipal budget is created for the fiscal year, and its execution is extended until March 15 of the fiscal year for liquidation and payment of expenses, and until March 31 of the fiscal year for payment of expenses.
- Practice the procedures of revenue liquidation, collection, and payment of expenses.
- The chairman of the municipal People's Council also prepares the municipal administrative account at the conclusion of the fiscal year's additional period.and submits it to the municipal People's Council for approval.

5 Study of the existing

The study of the existing is an approach used very often in the design cycle of an IT solution. It consists in analyzing the existing competitor or comparable, to extract positive and/or negative points. This allows designers to identify and address any gaps in the existing solutions, as well as to create unique features or experiences. Through this approach, designers can also draw conclusions about user behaviors and preferences.

And through our visit to the studied municipality to get acquainted with the way of managing the municipal budget, they have presented to us the budget management system that exists there, which is a desktop application that works on one device.
These are some of its interfaces :

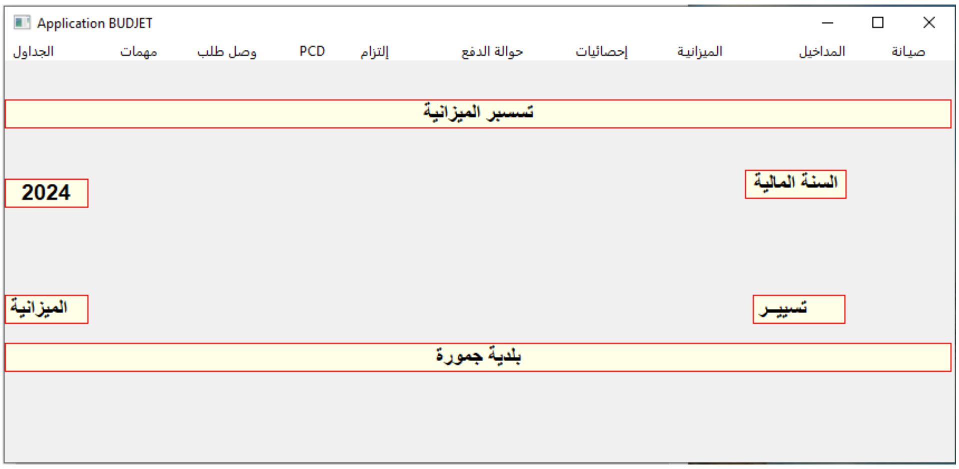
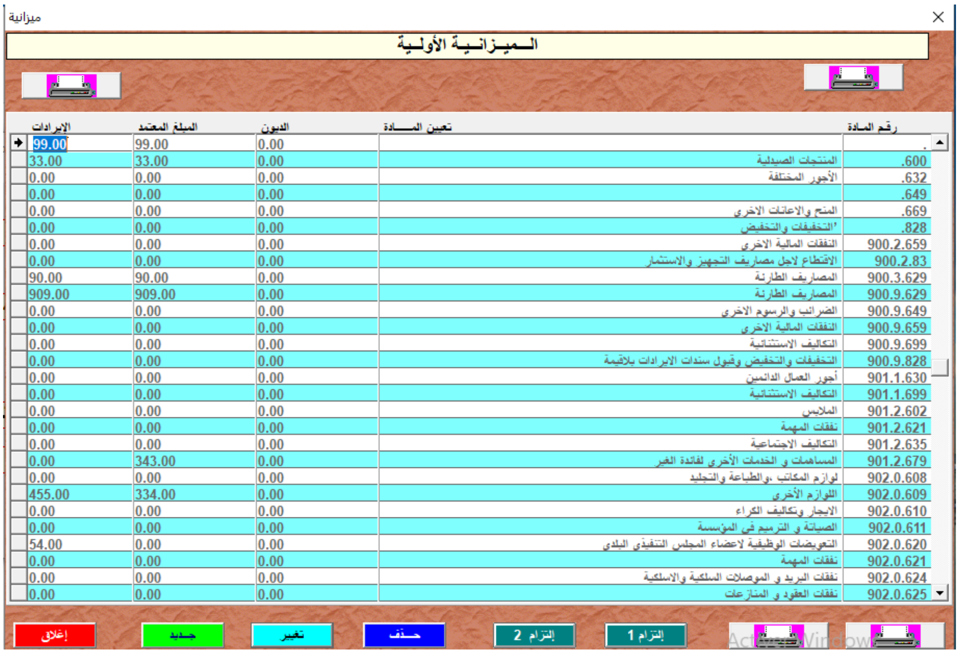
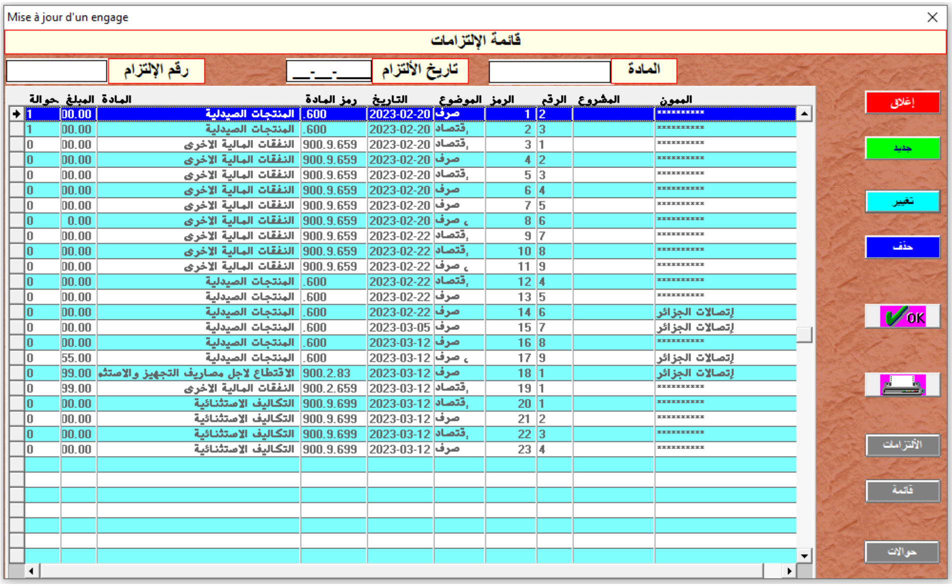


Figure I.2: welcome page .



الإيرادات	المبلغ المعتمد	الديون	تعيين المـــادة	رقم المـادة
99.00	99.00	0.00		.
33.00	33.00	0.00	المنتجات الصيدلية	.600
0.00	0.00	0.00	الأجور المختلفة	.632
0.00	0.00	0.00		.649
0.00	0.00	0.00	المنح والاعانات الاخرى	.669
0.00	0.00	0.00	التخفيفات والتخفيض'	.828
0.00	0.00	0.00	النفقات المالية الاخرى	900.2.659
0.00	0.00	0.00	الاقتطاع لاجل مصاريف التجهيز والاستثمار	900.2.83
90.00	90.00	0.00	المصاريف الطارئة	900.3.629
909.00	909.00	0.00	المصاريف الطارئة	900.9.629
0.00	0.00	0.00	الضرائب والرسوم الاخرى	900.9.649
0.00	0.00	0.00	النفقات المالية الاخرى	900.9.659
0.00	0.00	0.00	التكاليف الاستثنائية	900.9.699
0.00	0.00	0.00	التخفيفات والتخفيض وقبول سندات الايرادات بلاقيمة	900.9.828
0.00	0.00	0.00	أجور العمال الدائمين	901.1.630
0.00	0.00	0.00	التكاليف الاستثنائية	901.1.699
0.00	0.00	0.00	الملابس	901.2.602
0.00	0.00	0.00	نفقات المهمة	901.2.621
0.00	0.00	0.00	التكاليف الاجتماعية	901.2.635
0.00	343.00	0.00	المساهمات و الخدمات الأخرى لفائدة الغير	901.2.679
0.00	0.00	0.00	لوازم المكاتب ،والطباعة والتجليد	902.0.608
455.00	334.00	0.00	اللوازم الأخرى	902.0.609
0.00	0.00	0.00	الايجار وتكاليف الكراء	902.0.610
0.00	0.00	0.00	الصيانة و الترميم في المؤسسة	902.0.611
54.00	0.00	0.00	التعويضات الوظيفية لاعضاء المجلس التنفيذي البلدي	902.0.620
0.00	0.00	0.00	نفقات المهمة	902.0.621
0.00	0.00	0.00	نفقات البريد و الموصلات السلكية والاسلكية	902.0.624
0.00	0.00	0.00	نفقات العقود و المنازعات	902.0.625

Figure I.3: Initial Budget .



Mise à jour d'un engage

قائمة الإلتزامات

رقم الإلتزام

تاريخ الألتزام __-__-____

المادة

حوالة	المبلغ	المادة	رمز المادة	التاريخ	الموضوع	الرمز	الرقم	المشروع	الممون
1	00.00	المنتجات الصيدلية	.600	2023-02-20	صرف	1	2		**********
1	00.00	المنتجات الصيدلية	.600	2023-02-20	إقتصاد	2	3		**********
0	00.00	النفقات المالية الاخرى	900.9.659	2023-02-20	إقتصاد	3	1		**********
0	00.00	النفقات المالية الاخرى	900.9.659	2023-02-20	صرف	4	2		**********
0	00.00	النفقات المالية الاخرى	900.9.659	2023-02-20	إقتصاد	5	3		**********
0	00.00	النفقات المالية الاخرى	900.9.659	2023-02-20	صرف	6	4		**********
0	00.00	النفقات المالية الاخرى	900.9.659	2023-02-20	صرف	7	5		**********
0	0.00	النفقات المالية الاخرى	900.9.659	2023-02-20	صرف ،	8	6		**********
0	00.00	النفقات المالية الاخرى	900.9.659	2023-02-22	إقتصاد	9	7		**********
0	00.00	النفقات المالية الاخرى	900.9.659	2023-02-22	إقتصاد	10	8		**********
0	00.00	النفقات المالية الاخرى	900.9.659	2023-02-22	صرف ،	11	9		**********
0	00.00	المنتجات الصيدلية	.600	2023-02-22	إقتصاد	12	4		**********
0	00.00	المنتجات الصيدلية	.600	2023-02-22	صرف	13	5		**********
0	00.00	المنتجات الصيدلية	.600	2023-02-22	صرف	14	6		إتصالات الجزائر
0	00.00	المنتجات الصيدلية	.600	2023-03-05	صرف	15	7		إتصالات الجزائر
0	00.00	المنتجات الصيدلية	.600	2023-03-12	صرف	16	8		**********
0	55.00	المنتجات الصيدلية	.600	2023-03-12	صرف ،	17	9		إتصالات الجزائر
0	99.00	الاقتطاع لاجل مصاريف التجهيز والاستثه	900.2.83	2023-03-12	صرف	18	1		إتصالات الجزائر
0	99.00	النفقات المالية الاخرى	900.2.659	2023-03-12	إقتصاد	19	1		**********
0	00.00	التكاليف الاستثنائية	900.9.699	2023-03-12	إقتصاد	20	1		**********
0	00.00	التكاليف الاستثنائية	900.9.699	2023-03-12	صرف	21	2		**********
0	00.00	التكاليف الاستثنائية	900.9.699	2023-03-12	إقتصاد	22	3		**********
0	00.00	التكاليف الاستثنائية	900.9.699	2023-03-12	صرف	23	4		**********

Figure I.4: commitment.

Through these interfaces and the use of the application, we find many disadvantages and shortcomings, including:

-The application can be accessed without authentication, which threatens its confidentiality.

-Difficulty in use which can lead to user frustration, decreased efficiency, and a poor user experience. It can also make a product or service less attractive to potential customers, resulting in a decrease in sales.

-Using the application and budgeting can only be done by one user, which leads to completion delays.

-The application runs on one platform (Windows) which makes it impossible to use it on another platform like Linux or android. This limitation greatly reduces its user base and hinders its adoption.

-The process of municipal budget management is incomplete in the application.

6 problems

After discussing about the system of managing the municipal budget, some problems and obstacles appear to us, including:

- Resource consumption such as (papers, pens).
- Managed more slowly.
- Lack of analysis and overview.
- Limited cooperation and access in Budget Management.
- Increased probability of errors.

7 Solutions

Developing a web application for municipal budget management can address many of the problems associated with traditional paper-based budgeting processes.
Our platform works to solve the problems of this management and provide some privileges to facilitate the process by :

- Providing transparency in the management, allowing to see the latest changes.
- Providing stakeholders with the opportunity to work together.
- Facilitate the measurement of results by providing various data and analytics.
- Digitization of management and preservation of certain resources.
- Automating and accelerating processes.

8 Conclusion

In this chapter, we presented a detailed study of the municipal accounting department and how to prepare budget documents, as well as the problems and solutions for this municipal budget system. This chapter presents an important stage in what constitutes the management of the municipal budget.

Chapter II

Generalities

1 Introduction

Nowadays, the use of the Web is undergoing an inevitable development, as we do not find an area untouched by Web applications (public and private institutions, financial and banking, education, trade, etc). Developing web applications dedicated to the budget management system can be a great idea, as it provides an effective way to manage financial resources for both individuals and companies. In this chapter, we will introduce some concepts and definitions related to the domain Web development and application of our project. We will also introduce the methodology They are applied while studying and developing an IT solution.

2 Information system

A systemic definition of IS seeks to place it in connection to other existing systems. (operating system and control system). Information about the device is processed and stored by IF.

making it accessible to the "control system" via the "operating system." A definition from the perspective of the company's goal certifies that IS is The system must make sure that the appropriate information is accessible at the appropriate moment and place. Another way to define "point of view" The idea views an information system as a database for the communication between applications and procedures.[2]

3 Computer system

A computer system is a collection of computing tools and communications channels used to create, process, store, route, present, or even delete data. The computer system, which consists of hardware, software, networks, and operational processes, is the computer portion of the system information. Hardware and software are the two components that make up a computer system. [3]

4 types of software applications

The market's three dominant families of computer applications are as follows:

- **A web application :** An application that utilizes web technologies and can be accessed through a browser is referred to as a web application. (Firefox, Google Chrome, IE, etc.). Its primary benefit is its affordable cost. Additionally, a web application can be accessed from any station (PC, smartphone, tablet, etc.) and location. Furthermore, operating system compatibility is not required. Web applications also provide better security management. Access is restricted by identification, and everything is centrally located on a server. Because updates are automatic and transparent, the risk of obsolescence is greatly reduced while evolution and innovation are ongoing. The biggest disadvantage of this kind of application is that it can only be used if there is an Internet network.
- **Desktop application :** A desktop application is an application that runs entirely on a single computer, so they does not require network access to function. One of its advantages is that it's faster and more stable since it doesn't depend on the performance of any other application. According to Cons, its downside is its portability as it needs to be installed on every machine to be available. It also has compatibility issues with operating systems.[3]
- **Mobile application :** A program "written in view of a specific utilisation (calcul, management, game, etc.)" is referred to as an application in the field of computer science. A mobile application is used via a smartphone, "portable telephone that provides computing and multimedia functions," or a tablet, "computer or portable computer terminal whose tactile display covers the entire surface." The key features of mobile applications include mobility and portability, whether they are on a smartphone or tablet.

The word "appli" can also be used to describe a mobile application, whereas the name "App" is used both in English and German. Despite the fact that the definitions that follow should only be viewed as summaries and because technology is always evolving, the review of literature agrees to divide mobile applications into three families.[4]

5 Web application architectures

5.1 1-tier Architecture :

1 tier architecture also known as single-tier architecture, is related to that form of software architecture in which all the needed components for the working of application are offered under the same package.

It means that the user interface, business, layers are accessible by the application under the same local drive. Both the client and server sit in the same computer. It is the simplest application architecture used. However, this tier is not appropriate for a web application. As it can only access data available in a single computer or server.

Microsoft Office is a prime example of a one-tier architecture. This is a cost-efficient architecture and applications based on this are much easier to build. The fundamental downside of this architecture is that it cannot share information from one client machine to others. Sometimes the applications based on 1 tier are unable to work if some changes are done in the machine.[5]

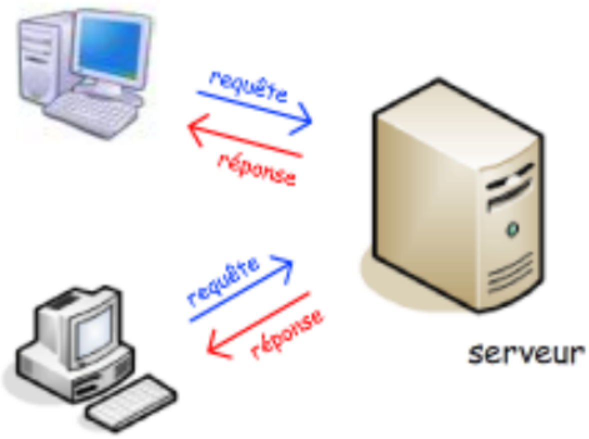


Figure II.1: Architecture 1-tiers.[6]

5.2 2-tier Architecture :

A 2 tier architecture involves separating the user interface layer and database layer into different machines. The client device sends instructions to the server which stores all the data and has the business layer. Communication between the presentation layer and database layer occurs by the internet using Transfer Control Protocol and Internet Protocol. This architecture is simple to maintain and modify, with fast communication between client and server. However, if the number of clients exceeds capacity, the server may become unresponsive and decrease productivity. Changes made to the application require reinstallation on the client, and since the business layer is on the client-side, high processing power is necessary.[5]

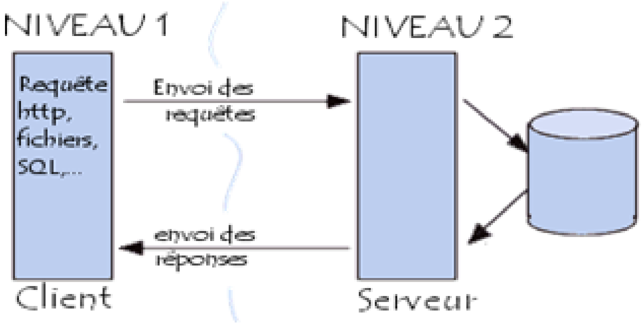


Figure II.2: Architecture 2-tiers.[6]

5.3 3-tier Architecture :

The presentation layer usually resides at the client level in this design, as it did in the previous one. Program logic and resource management pertaining to them are contained in the middle-tier (middle-tier) and data management layer, respectively. (back-end).

The main advantages of the three-tier architecture are:

- From a development point of view, the separation that exists between client and server Database Management System (DBMS) allows specialization Architect developers at all levels.
- Greater flexibility in server-level portability, allowing Changes according to development.

The primary disadvantage is:

- The time required to acquire development expertise, which is longer than in the case of a 2-tier architecture.[7]

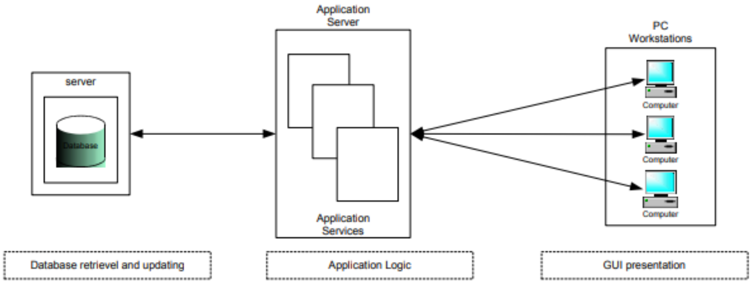


Figure II.3: Architecture 3-tiers.[8]

5.4 n-tier Architecture :

Multi-tier architecture, also called deployment architecture, where n is the number of tiers. The difference between 3 tier and n tier is that there are multiple application servers that mediate between the UI tier and the database tier. It distributes your business logic across multiple servers, allowing each to run individually. This architecture is very flexible as the number of database servers increases. n Tier -based applications are more secure than other layers of the architecture.[5]

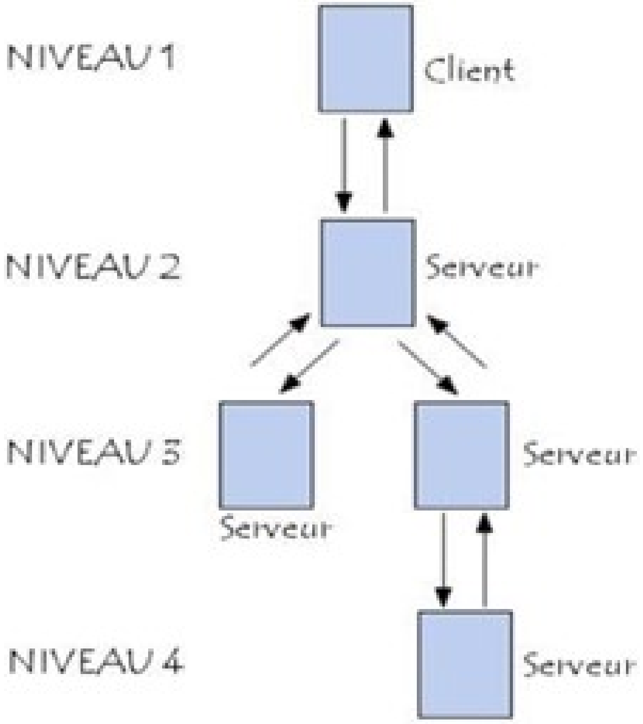


Figure II.4: Architecture N-tiers.[6]

6 CONCEPT OF MODEL VIEW CONTROLLER(MVC):

MVC has 3 models to separate the functionality of the system, namely model, view and controller. A model is a module in the system business logic and the core of a system. View is a user interface model that comes from the controller yang to produce the initial appearance of the system in response to user commands. Although the controller is a module in the system implementation, it later flows as a control between the view module and the model module.[9]

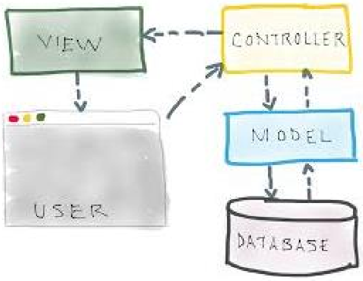


Figure II.5: CONCEPT OF MODEL,VIEW,CONTROLLER ARCHITECTURE (MVC)

[10]

7 Our project architecture

In our municipality budget management web application, we have utilized both the MVC (Model-View-Controller) architecture and the 3-tier architecture to ensure a well-organized and maintainable codebase.

7.1 MVC Architecture

We implemented the MVC pattern to separate the concerns of the application into three distinct components:

- **Model:** The Model component represents the data and business logic of the application. In our project, the Model encapsulate the budget data, interacte with the database. The Model layer also included validation logic and data integrity checks to ensure the accuracy and consistency of the budget data.
- **View:** The View component was responsible for rendering the user interface and presenting the budget information to the users. We designed

intuitive and user-friendly views to display budget summaries, charts, and reports. The View component interacted with the Controller to retrieve the necessary data from the Model and present it in a visually appealing manner.

- **Controller:** The Controller component handle user interactions, receive requests from the View, and coordinate the flow of data between the Model and the View. It implemente the business logic to handle budget allocation, expenditure tracking, and financial reporting. The Controller also ensure the integrity of the data by validating user input before updating the Model.

By separating these concerns, the MVC architecture allowed us to maintain a clean codebase with clear responsibilities for each component. It facilitated code reusability, modular development, and easier collaboration among team members.

7.2 3-tier Architecture

In addition to the MVC architecture, We incorporated the 3-tier architecture to further organize the application into separate layers, each with its own responsibility:

- **Presentation Layer:** This layer encompassed the View and the Controller components of the MVC architecture. It handled the user interface, user input, and presentation logic. The View component was responsible for rendering the HTML templates and displaying the budget-related information to the users. The Controller component received user requests, processed them, and coordinated the interaction between the View and the Model.

- **Application Layer:** The Application Layer included the Model component from the MVC architecture, along with the business logic specific to budget management. This layer ensured that the budget data and its associated operations were well-abstracted and separated from the presentation layer.

- **Data Layer:** The Data Layer represented the database and data access functionality. It provided an interface to interact with the database. The Data Layer was responsible for retrieving and persisting the budget data, ensuring its integrity, and implementing any necessary data transformations or optimizations.

By adopting the 3-tier architecture, We achieved better separation of concerns, scalability, and maintainability. Each layer had its own specific responsibilities, allowing for easier updates, enhancements, and troubleshooting.

Overall, by combining the MVC and 3-tier architectures in our municipality budget management web application, we achieved a clear separation of concerns, modularity, and maintainability. These architectural choices facilitated efficient development, testing, and future enhancements of the application.

8 The different types of websites

- **Static websites :** the initial design of a website [4]. Figure 1 provides an illustration of the enlarged definition utilized in the current study. a static site is made up of HTML pages, documents, and media that can be read by the server from persistent storage and served to the client (often a web browser) without further customization. The server doesn't perform any runtime modifications or other actions on the saved page. A static website may contain HTML, CSS, multimedia files, and perhaps client-side JavaScript or other programming language scripts or programs that can be executed by the browser. Typically content-driven (read-only), these websites can also be interactive using tools or games.[11]

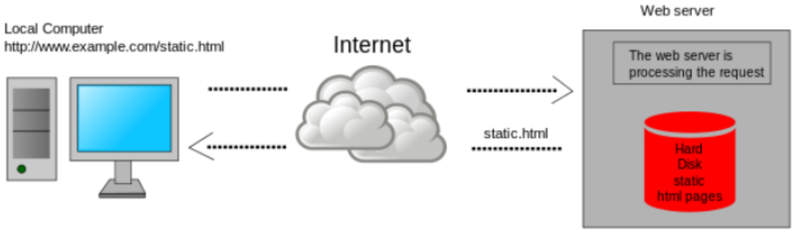

Local Computer
http://www.example.com/static.html
Internet
Web server
The web server is processing the request
static.html
Hard Disk static html pages

Figure II.6: An example how browser requests and server responds with a static page.[11]

- **dynamic websites :** Dynamic sites are those that are stored in a format that the web browser cannot directly render. After performing some intermediate steps, the server receives a client request and produces a response that the browser can render. Figure 2 shows how operations are specified in a scripting or programming language, with some of the most popular examples being PHP, Python, Perl, Ruby, Java, C#, and JavaScript by Node js. Operations frequently contain database requests. It is possible for a website to be inherently dynamic, offering interactive client-server activity, or it is also possible for it to merely translate statically stored data to an HTML page.[11]

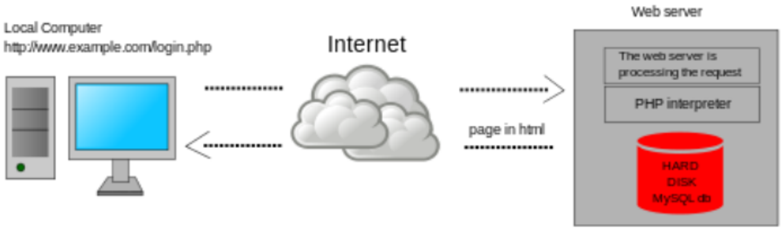


Figure II.7: Server gets HTML through PHP interpreter which uses MySQL database[11]

9 Conclusion

In this chapter of generalities, we have defined some general concepts related to the design of web applications. We used UML diagrams and a detailed description of this language is given in the next chapter.

Chapter III

Analysis and Design

1 Introduction

Before developing an information system, it is essential to select a good, well-defined method for organizing the work to be done by naming the various tasks and stages. We chose the UML technique for the representation it provides through its various diagrams. This chapter will identify the various actors and their duties, then elaborate the use case diagrams, sequence diagrams, activity diagrams, and class diagrams.
This approach can be represented graphically as follows :

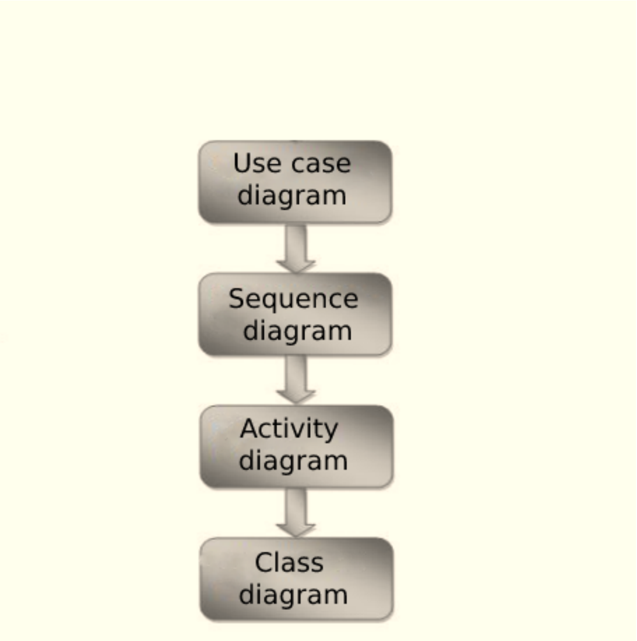


Figure III.1: The application modeling approach

2 Description of needs

In computer science, it is necessary to determine the needs well. In fact, it represents the most delicate and important work.
These needs divided into two type (the functional needs and non-functional needs).
The envisaged application must meet the functional needs that the system will implement and non-functional needs that will improve the quality of the

system software.

2.1 Functional needs

-Contains a database for storing budget documents.
-Manipulate and update the database.
-An overview and detailed description of the completed work.
-The designed application will have to work in 3-third-party mode (client, data server, application server).
-Authentication access to the application is required.
-A list of the budget of previous years.

2.2 Non-functional needs

-Prevent direct access with URL links and set a time limit for non-active logoff.
-The interface of this application must be ergonomic, user-friendly and even able to help the user better manage his workspace.
-It is necessary to ensure the security of the various manipulations and updates made in the database.
-The code must be clear and organized to allow future evolutions or improvements.

3 UML Presentation

3.1 Definition

UML (Unified Modeling Language) Allows you to visualize and manipulate object concepts and employ analytical methods, allows you to iterate design solutions with the help of diagrams and visually express object solutions. It is a graphical and textual modeling language, an important step in the systems development cycle, used to visualize, understand, and define requirements, specify and create the documentation needed to properly develop object-oriented software, sketch software architecture, and design communication Solutions and Perspectives. These models must be realistic.[12]

3.2 strong points of UML

- **Standardization:** UML diagrams can be understood by developers from different backgrounds.
- **Abstraction:** UML provides simple and abstract concepts for representing complex systems.
- **Flexibility:** UML diagrams can be customized to meet the specific needs of a project.
- **efficiency:** UML model can automatically generate code to improve efficiency.
- **Communication:** UML diagrams are easy to read and understand, making them a useful tool for communication between developers.

3.3 Static representation of the system

- Class diagram.
- Object diagram.
- Component diagram.
- Deployment diagram.
- Package diagram.
- Use case diagram.
- Diagram of composite structure.

3.4 dynamic representation of the system

- activity diagram.
- sequence diagram.
- transition-state diagram.
- collaboration diagram.
- communication diagram.

4 The Different UML Diagrammes

4.1 Use case diagram

4.1.1 Definition

The use case diagram depicts the structure of the primary system functionalities. The first diagram in the UML model ensures the relationship between the user and the items that the system implements.[13]

4.1.2 Identification of the actors

- **Definition of an actor:** Actors represent external entities (human users or systems) which interact directly with the application and have the ability to consult or modify its state.
- **The actors of our system:**
  -Accountant: Which represents the accountant's post as defined in the previous chapter.
  -Users: They represent the various Cashiers of the service that we have seen previously.

4.1.3 Use case diagram

The following figure describe the use case diagram according to the actors:

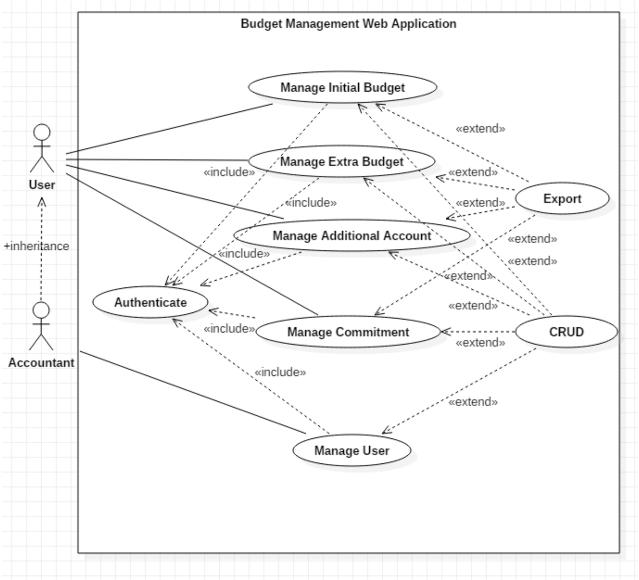


Figure III.2: Use Case Diagram

4.2 Sequence diagram

4.2.1 Definition

The sequence diagram depicts the chronological order of the operations performed by the actor. It denotes the items that the actor will manipulate as well as the actions that will be performed to go from one object to another. [13]

4.2.2 Sequence diagrams

These are the most usefull sequences in the application according to each functionality:

- **Sequence Diagram of authentication process:**

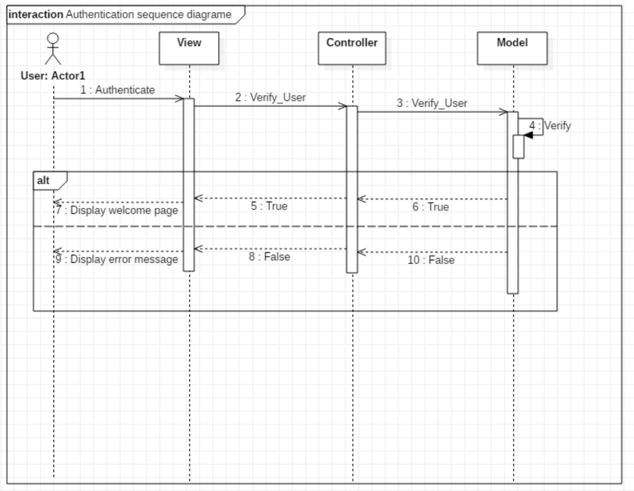


Figure III.3: Authentication sequence Diagram

- **Sequence Diagram of add users process:**

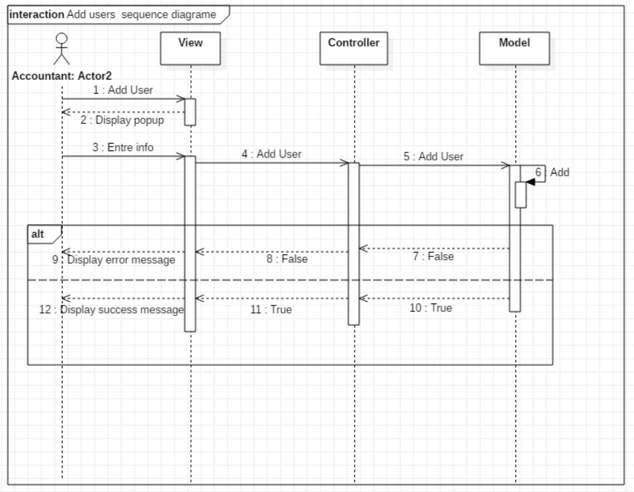


Figure III.4: Add users Sequence Diagram

- **Sequence Diagram of initial budget process:**

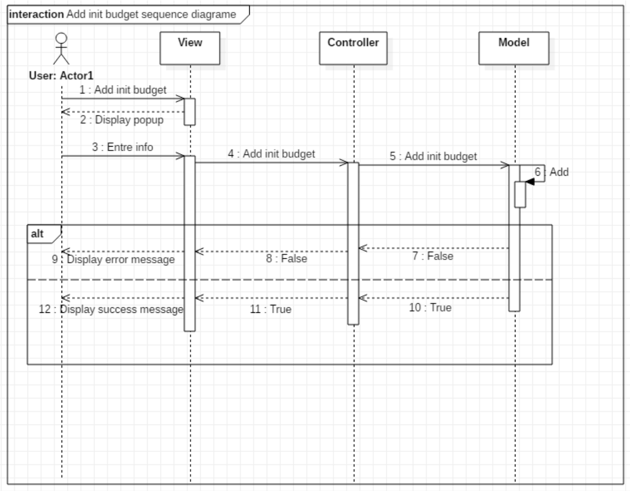


Figure III.5: Add initial budget sequence Diagram

- **Sequence Diagram of export process:**

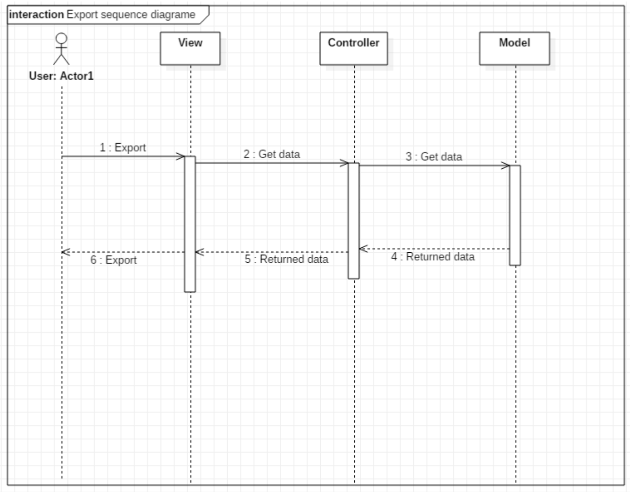


Figure III.6: Export sequence Diagram

4.3 Activity diagram

4.3.1 Definition

The activity diagram depicts the sequence of activities that take part in the process and its behavior in terms of actions. The diagram can also show how activities are divided into different groups and how the flow of control is directed. It is useful in understanding the logic of complex systems or documenting existing processes.

4.3.2 Activity diagrams

The following activity diagrams illustrate the most significant processes in the application:

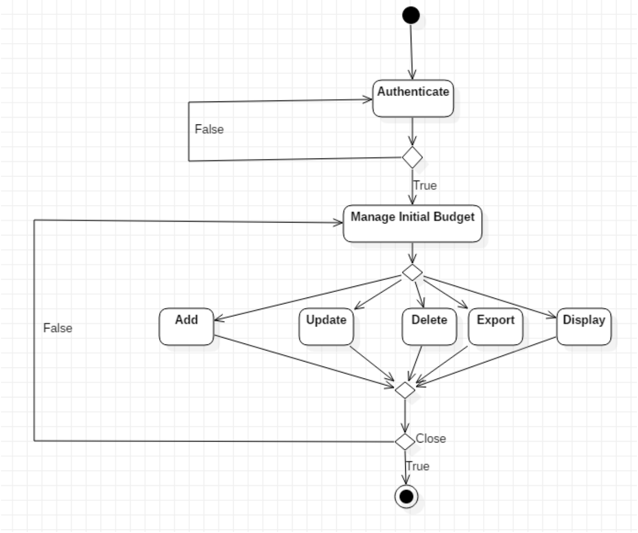


Figure III.7: Initial Budget Management Activity Diagram

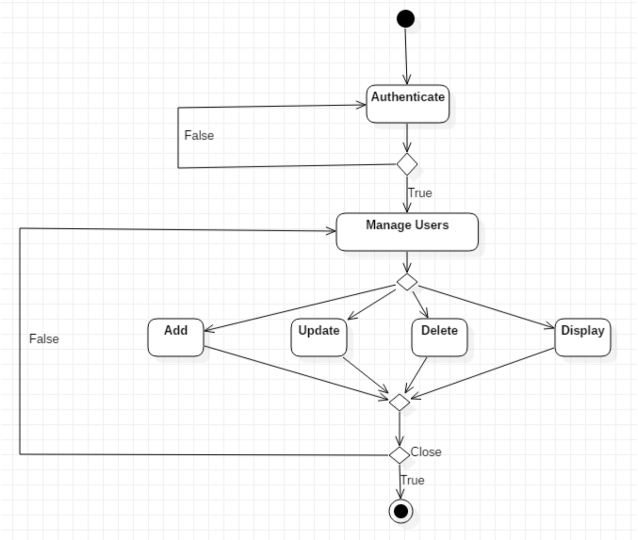


Figure III.8: Users' Management Activity Diagram

4.4 Class diagram

4.4.1 Definition

The class diagram is often regarded as the most important in object-oriented development. It displays the system's conceptual architecture, it explains the classes that the system uses, as well as their relationships, which reflect a conceptual interlocking (inheritance) or an organic link(aggregation).[13]

It mainly contains classes, which contain attributes and operations. It demonstrates the links between these classes and their cardinalities.

4.4.2 Data dictionary

The description of classes and their data is given in the following table:

Table III.1: The description of classes and their data

Class	Attribute	Code	Type	Mehtod
Caterer	The identifier of the class	Id	Integer	
	Name of the caterer	Nom	String	
	Address of the caterer	Address	String	
	Number of caterer phone	Phone	Integer	
	Commercial Registre of caterer	Cr_number	Integer	
Commitment	The identifier of the class	Id	Integer	Add()
	Token of Commitment	Token	Integer	Delete()
	Number of Commitment	Number	Integer	Update()
	Subject of commitment	Subject	String	Search()
	Amount of commitment	Amount	Integer	Export()
	Assignment of the commitment	Assignment	String	Display()
	Date of commitment	Date	Date	
Account	The identifier of the class	Id	Integer	
	Number of account	Account_number	Integer	
	Name of the bank	Bank	String	
Users	The identifier of the class	Id	Integer	
	Name of the user	Username	String	
	Password of the users	Password	String	
	Language of the user	Language	String	
Accountant				Add_User()
				Delete_User()
				Display_Users()
Articles	The identifier of the class	Id	Integer	
	Numero of the article	Num_Article	String	
	Name of the article	Nom	String	
	Type of the article	Type	String	
Extra budget	The identifier of the class	Id	Integer	Add()
	The Amount of previous initial budget	Initial_Budget	Integer	Delete()
	The Amount of increase	Increase	Integer	Update()
	The Amount of Decrease	Decrease	Integer	Search()
	Proposition amount	Proposition	Integer	Export()
	Confirmation amount	Confirmation	Integer	Display()
	Year of the extra budget	Year	Year	
Initial budget	The identifier of the class	Id	Integer	Add()
	Reminder of previous amount	Account_reminder	Integer	Delete()
	Proposition amount	Proposition	Integer	Update()
	Confirmation amount	Confirmation	Integer	Search()
	Year of the initial budget	Year	Year	Export()
				Display()
Sub chapters	The identifier of the class	Id	Integer	
	Number of port	Port	Integer	
	Quantity of sub chapter	Quantity	Integer	
	Year of the sub chapter	Year	Year	
Chapters	The identifier of the class	Id	Integer	
	Number of chapter	Ch_Numero	Integer	
	Nom of the chapter	Nom	String	
Additional Account	The identifier of the class	Id	Integer	
	The amount of previous extra budget	Extra_Budget	Integer	Add()
	Special licenses amount	Special_licenses	Integer	Delete()
	Total of the additional account	Total	Integer	Update()
	Amount of selections	Selections	Integer	Search()
	Fulfilment of the additional account	Fulfilment	Integer	Export()
	Rest of fulfilment	Rest_F	Integer	Display()
	year of the additional account	Year	Year	

4.4.3 class diagram

After the description of the existing data, the following figure represents our class diagram:

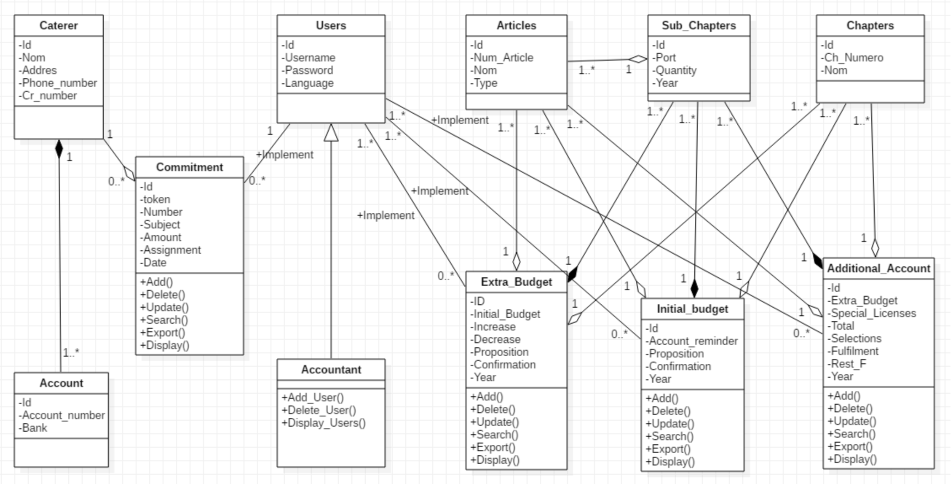


Figure III.9: Class Diagram

5 Database design

5.1 Transition from the class diagram to the relational model

- Each class becomes a table, with the primary key often corresponding to the class identification.

- The many-to-many association results in the creation of a table whose main key is made up of the primary keys of the tables corresponding to the connected classes.

- The relationship of one to several (father-son) leads the main key to migrate from the father to the son, becoming a foreign key.

- The specialized classes are transformed into tables whose primary key is the primary key of the table corresponding to the general class.

5.2 The relational model

- **Remarque :**
  The word in **Grass**: primary key.

word underline: secondary key.

account (**id**, caterer_id, account_number, bank);

ad_account (**id** ,article_id,extra_budget,special_licenses,total,selections, fulfilment ,rest_fulfilment,year);

article(**id**,num_article,nom_fr,nom_ar,chapter_id,type_id);

article_types (**id**,nom_fr,nom_ar);

budget_types (**id**,nom_fr,nom_ar);

caterer (**id**,tax_code,statistical_number,cr_number,address,phone_number, fax_number,institution_nature,nom_fr,nom_ar);

chapters (**id**,ch_numero,nom_fr,nom_ar);

chapters_types (**id**,chapters);

commitment (**id**,token,number,article_id,sub_chapter,subject_id,caterer_id, amount,assignment,date);

commitment_subject (**id**, nom_fr,nom_ar);

extra_budget (**id**, article_id,initial_budget,increase,decrease,proposition, confirmation,year);

initial_budget (**id**,article_id,account_reminder,proposition,confirmation,year);

rules (**id**,article_id,port);

sub_chapters (**id**, article_id,port,quantity,budget_type,year);

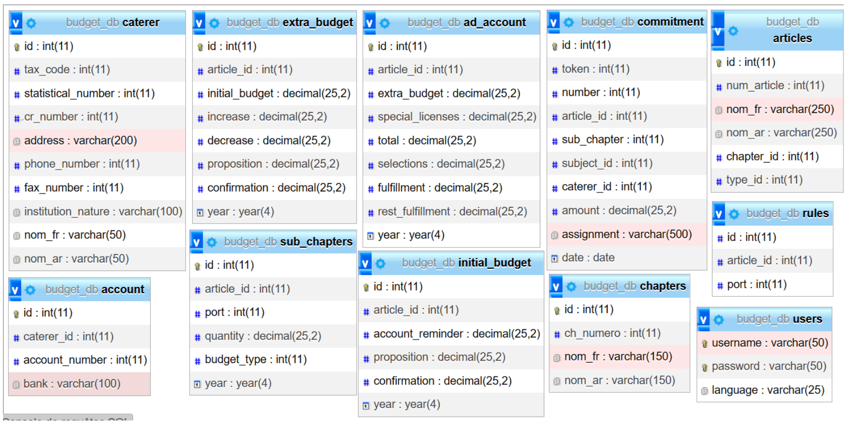
users (**id**, username,passwordss);

5.3 Databse tables

We executed SQL statements to define the table structures, including column names, data types, and any constraints. Through this process, we ensured the tables were accurately created and aligned with our database design.

To generate the database tables, we utilized XAMPP, a popular software package that provides an easy-to-use development environment. With XAMPP, we set up a local server environment that includes MySQL. We leveraged the MySQL command-line interface or a graphical user interface (phpMyAdmin) to create the necessary tables.

The following figure gives an overview of our database tables:



budget_db caterer
id : int(11)
tax_code : int(11)
statistical_number : int(11)
cr_number : int(11)
address : varchar(200)
phone_number : int(11)
fax_number : int(11)
institution_nature : varchar(100)
nom_fr : varchar(50)
nom_ar : varchar(50)

budget_db account
id : int(11)
caterer_id : int(11)
account_number : int(11)
bank : varchar(100)

budget_db extra_budget
id : int(11)
article_id : int(11)
initial_budget : decimal(25,2)
increase : decimal(25,2)
decrease : decimal(25,2)
proposition : decimal(25,2)
confirmation : decimal(25,2)
year : year(4)

budget_db sub_chapters
id : int(11)
article_id : int(11)
port : int(11)
quantity : decimal(25,2)
budget_type : int(11)
year : year(4)

budget_db ad_account
id : int(11)
article_id : int(11)
extra_budget : decimal(25,2)
special_licenses : decimal(25,2)
total : decimal(25,2)
selections : decimal(25,2)
fulfillment : decimal(25,2)
rest_fulfillment : decimal(25,2)
year : year(4)

budget_db initial_budget
id : int(11)
article_id : int(11)
account_reminder : decimal(25,2)
proposition : decimal(25,2)
confirmation : decimal(25,2)
year : year(4)

budget_db commitment
id : int(11)
token : int(11)
number : int(11)
article_id : int(11)
sub_chapter : int(11)
subject_id : int(11)
caterer_id : int(11)
amount : decimal(25,2)
assignment : varchar(500)
date : date

budget_db chapters
id : int(11)
ch_numero : int(11)
nom_fr : varchar(150)
nom_ar : varchar(150)

budget_db articles
id : int(11)
num_article : int(11)
nom_fr : varchar(250)
nom_ar : varchar(250)
chapter_id : int(11)
type_id : int(11)

budget_db rules
id : int(11)
article_id : int(11)
port : int(11)

budget_db users
username : varchar(50)
password : varchar(50)
language : varchar(25)

Figure III.10: Database tables.

6 Application architecture

The following figure gives an overview of our 3-tier architecture with existing functions. The architecture is designed to be scalable and efficient. It allows for new functions to be added easily and enables the system to be customized for different user requirements. This architecture has enabled us to create an efficient and reliable system.

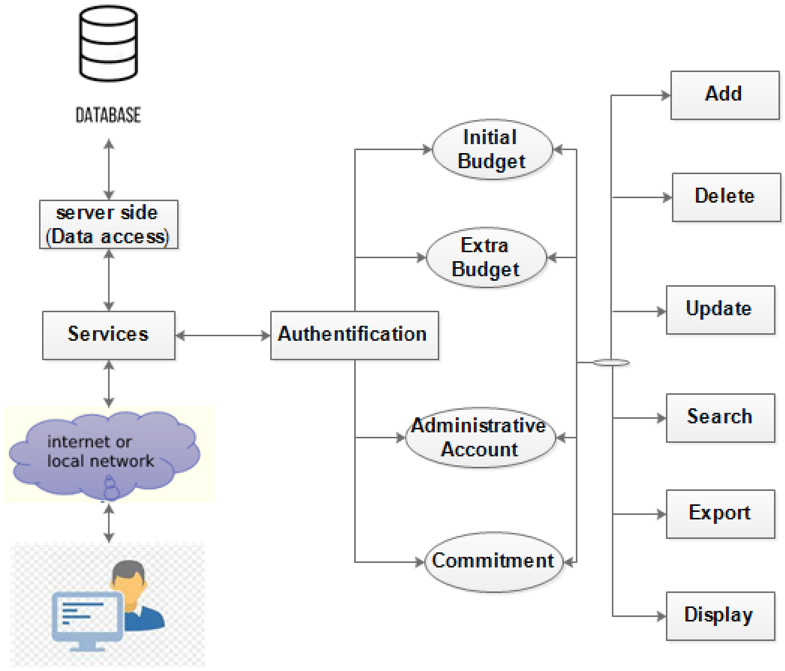


Figure III.11: Architecture three-tier of the application.

7 Conclusion

We were able to finalize and detail the design and modeling of our system for the multi-platform application of municipal budget management in this second semester. By first adapting to the modeling language and based on the UML formalism diagrams, beginning with the diagram describing our system with these different functionalities, such as the use case diagram, then the sequence diagrams that show the dialogue that takes place between the users, our system, and the database, and the activity diagrams describing the system behavior on their side. Finally, the class diagram displays the classes that interact with the system as well as the final components.

Chapter IV

Implementation

1 Introduction

The objective of this chapter is to present the working environment and the tools to be used.

2 Technologies

2.1 PHP

- **Definition:** PHP is a server-side scripting language that was created expressly for the Internet. PHP code can be embedded in an HTML page and run every time the page is viewed. At the Web server, your PHP code is interpreted and produced HTML or other output that the visitor will view. PHP was created in 1994 by Rasmus Lerdorf, who worked alone at the time. It was embraced by other gifted individuals and has undergone three major rewrites to produce the broad, mature product we see today.[14]

Figure IV.1: PHP-Logo

2.2 MySQL

- **Definition:** MySQL, a popular open source SQL database management system, was built, distributed, and supported by Oracle Corporation. MySQL is a database that supports structured data collections. A MySQL database makes it easier to add, access, and handle data contained in the database. MySQL stores information in different tables. The database structures are organized into physical files that are performance-optimized. A diverse programming environment is provided by the logical model, which comprises objects like as databases, tables, views, rows, and columns. SQL stands for "Structured Query Language," and it is the most extensively used standardized language for accessing databases. MySQL software is free and open source, licensed under the GNU General Public License. (GNU General Public License). [15]

Figure IV.2: My-SQL Logo

2.3 Apache

- **Definition:** It is an HTTP server created and maintained within the Apache Foundation. Used to serve static and dynamic web pages, it has emerged as the first practical alternative to the Netscape web server since 1995 by Robert McCall Used Mainly on Internet hosting under Linux, although it can also be used in Windows.

Figure IV.3: apache-logo

2.4 Xampp

- **Definition:** XAMPP is a tiny and lightweight Apache distribution that includes the most commonly used web development tools in a single package. Its content, modest size, and mobility make it a great tool for students designing and testing PHP and MySQL applications. XAMPP is available for free download in two flavors: full and lite. While the entire package download includes a wide range of development tools, this post will concentrate on using XAMPP Lite, which includes the technologies required to achieve the Ontario Skills Competition guidelines. The light version, as the name implies, is a compact package that includes Apache HTTP Server, PHP, MySQL, phpMyAdmin, Openssl, and SQLite.[16]

Figure IV.4: Xampp-logo

2.5 Laravel

- **Definition:** Laravel is a PHP-based web framework for building rich web applications with its sensible and clean syntax. Laravel was developed by Taylor Otwell in July 2011 and released more than five years after the release of Codeigniter. It comes with a powerful toolkit and provides an application architecture for . also includes various features of technologies like ASP.NET MVC, CodeIgniter, Ruby on Rails and many more.Dix Framework Jest Framework Open Source . This helps reduce the thought and planning involved in creating a complete website from inception and saves developers a ton of time. The security of the application is also taken care of by Laravel. This allows all of its features to quicken the rate at which your website is developing. Laravel will build your work with ease if you are familiar with PHP scripting and its fundamentals. Model-View-Controller (MVC) architecture is used by Laravel.[10]

Figure IV.5: Laravel logo [17]

2.6 Bootstrap

- **Definition:** The well-known HTML, CSS, and JavaScript framework known as Bootstrap is used to create responsive and mobile-friendly websites. The most widely used HTML is Bootstrap. Building a responsive and mobile-friendly website with CSS and JavaScript.

  Use and download are both completely free.

  It is a front-end framework used to make web development simpler and quicker.

It offers design templates for typography, forms, buttons, tables, navigation, modals, image carousels, and many other things that are based on HTML and CSS.

Plug-ins for JavaScript are also an option.

Figure IV.6: bootstrap-logo-vector

You can easily construct responsive designs thanks to it. Bootstrap, Why Use It:[18]

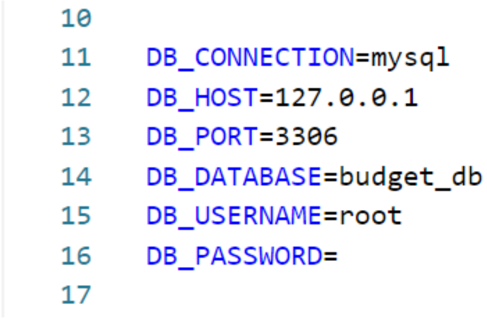
- Bootstrap 3 is mobile-first architecture includes mobile-first styles throughout the entire library rather than in separate files.
- It is compatible with all widely used browsers.
- Simple to get started with. Anyone can use Bootstrap with just a basic understanding of HTML and CSS. The official Bootstrap website also provides excellent info.
- Design that is mobile, tablet, and desktop-responsive thanks to Bootstrap's flexible CSS. The chapter Bootstrap Responsive Design has more information about responsive design.
- And the greatest part is that it is open source.[18]

3 Understanding Laravel

The purpose of this chapter is to introduce the fundamental principles and components of the Laravel framework. We aim to provide you with a comprehensive understanding of how Laravel works by providing code snippets and accompanying descriptions.

3.1 Database connection

Laravel provides a specific file .env for storing and retrieving data, allowing seamless integration with various databases. In this file, instructions are provided for configuring the database connection, which is an important step in the setup of a Laravel application.



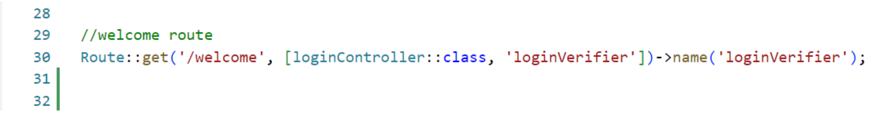
```

DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=budget_db
DB_USERNAME=root
DB_PASSWORD=

```

Figure IV.7: Database connection

In this code snippet, we specify the database connection type as MySQL. We provide the host, port, database name and username for the database we want to connect to.

3.2 Routing

In any web application, routing is a crucial component, and Laravel provides a clean and expressive way to define routes. Here's an example of defining a basic route in Laravel:



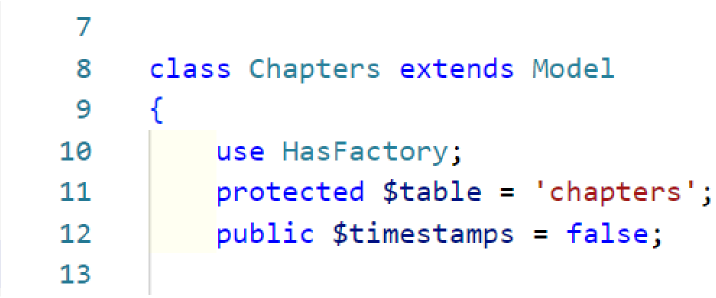
```

//welcome route
Route::get('/welcome', [loginController::class, 'loginVerifier'])->name('loginVerifier');


```

Figure IV.8: Routing method

In this code snippet, we define a GET route for the /welcome URL. This route returns the results of the loginVerifier method performed by the loginController controller. We name it loginVerifier to be used in our views.

3.3 Models

Once the database configuration is set, you can use Laravel's Eloquent ORM to interact with the database which is called models. Here's an example of retrieving data from a user's table using it.



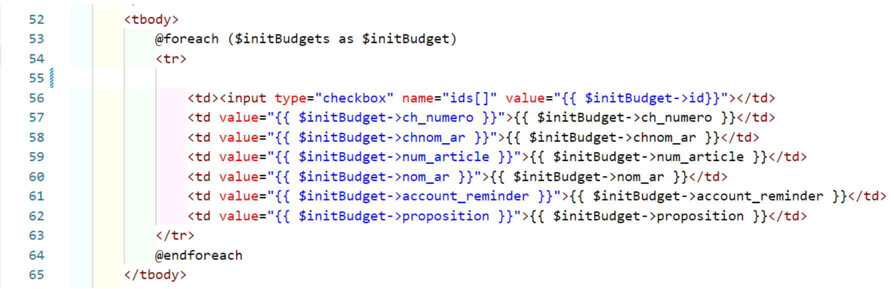
```

class Chapters extends Model
{
    use HasFactory;
    protected $table = 'chapters';
    public $timestamps = false;

```

Figure IV.9: Laravel Model

In this example we used the Chapters class that extends the model class provided by Laravel to represent the chapter table of the database. In this way, each table in the database is represented. We can specify the target table name with the property $table.

3.4 Views

Views present data to the user. Laravel uses the Blade templating engine, which offers concise syntax for creating dynamic views. Here's an example of a Blade template:



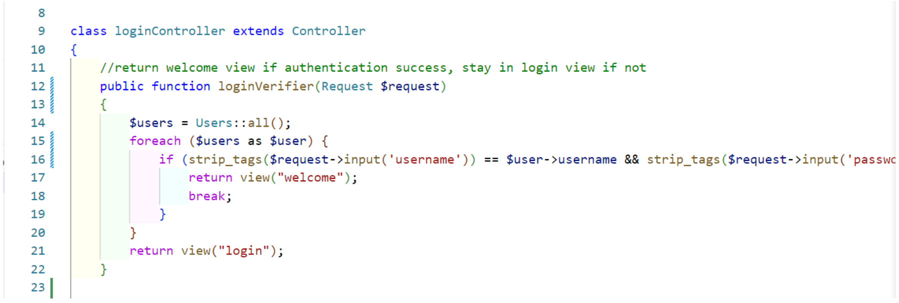
```
<tbody>
    @foreach ($initBudgets as $initBudget)
    <tr>

        <td><input type="checkbox" name="ids[]" value="{{ $initBudget->id}}"></td>
        <td value="{{ $initBudget->ch_numero }}">{{ $initBudget->ch_numero }}</td>
        <td value="{{ $initBudget->chnom_ar }}">{{ $initBudget->chnom_ar }}</td>
        <td value="{{ $initBudget->num_article }}">{{ $initBudget->num_article }}</td>
        <td value="{{ $initBudget->nom_ar }}">{{ $initBudget->nom_ar }}</td>
        <td value="{{ $initBudget->account_reminder }}">{{ $initBudget->account_reminder }}</td>
        <td value="{{ $initBudget->proposition }}">{{ $initBudget->proposition }}</td>
    </tr>
    @endforeach
</tbody>
```

Figure IV.10: Laravel view

In this code we catch the data returned by the controller and display it in a table using double curly braces. We loop through the items in the data and create a row for each item. Finally, we add styling to the table to make it look neat and organized.

3.5 Controllers

Controllers serve as the central point for handling requests and performing logic. Here is an example of a basic controller in Laravel:



```

class loginController extends Controller
{
    //return welcome view if authentication success, stay in login view if not
    public function loginVerifier(Request $request)
    {
        $users = Users::all();
        foreach ($users as $user) {
            if (strip_tags($request->input('username')) == $user->username && strip_tags($request->input('passwc
                return view("welcome");
                break;
            }
        }
        return view("login");
    }

```

Figure IV.11: Laravel controller

In this code snippet, we define a loginController class that extends the Laravel base Controller class. Within the loginverifier method, we fetch all users from the database using the User model. We test if the given data matches the existing one to pass the next view.

4 Application interfaces

The budget management of a municipality is based on 4 functionality: the execution of the initial budget, extra budget, administrative account and the different commitments.

Besides supporting these functions, the application offers a number of features designed to facilitate budget management, as shown in the following interfaces:

4.1 Authentication interface:

When the application starts, the authentication interface is displayed.

The user must enter his username and password to authenticate The application then verifies the credentials against a database. If the credentials are valid, the user is granted access. If not, the authentication process is repeated.

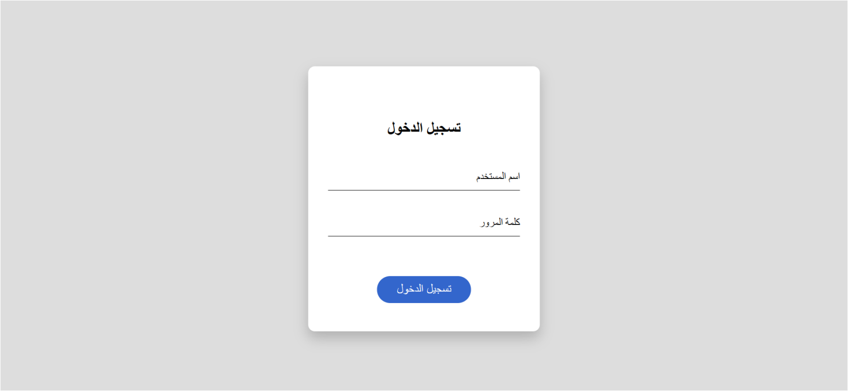


Figure IV.12: Login View

4.2 Welcome interface:

After the authentication succeeds, the user passes to the main interface which contains some statistics about the current budget state and the other functionality to start working in it. Moreover, there are several features like changing the language or theme mode of the application and making it full-screen.

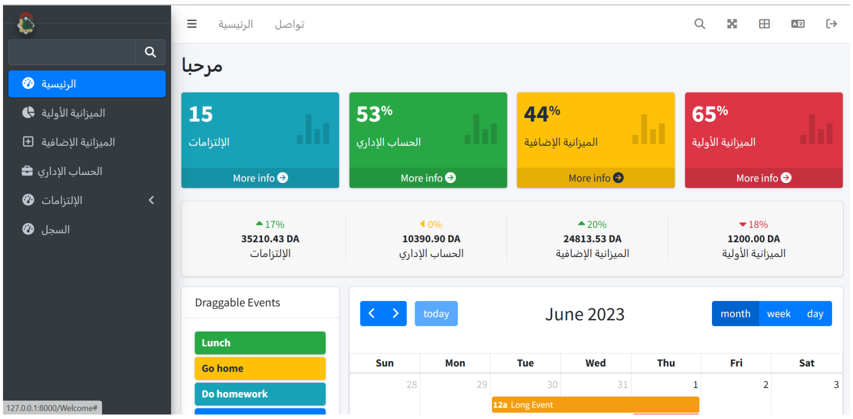
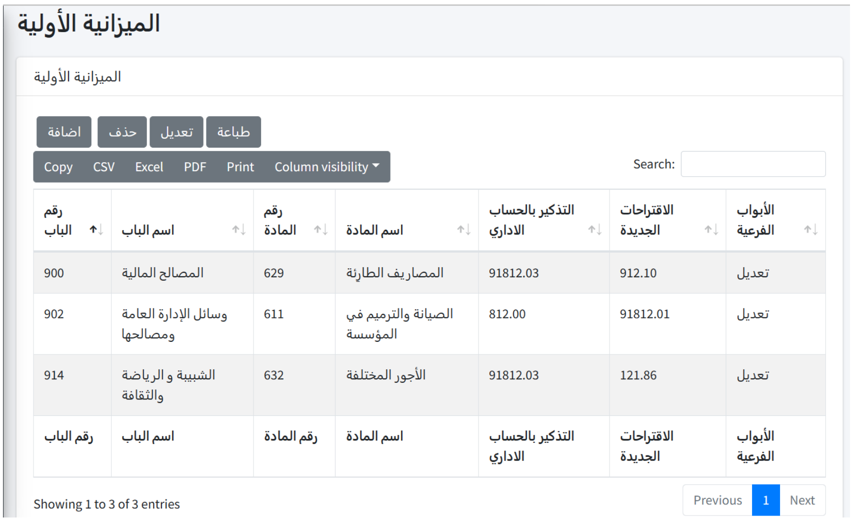


Figure IV.13: Welcome View

4.3 initial Budget interfaces:

One of the first parts of budget management is the initial budget. The following interfaces simplifies working with it, with a well divided table to display its different items that have been done. It also has several functions such as add, update, search, delete and export it in diffrent format.

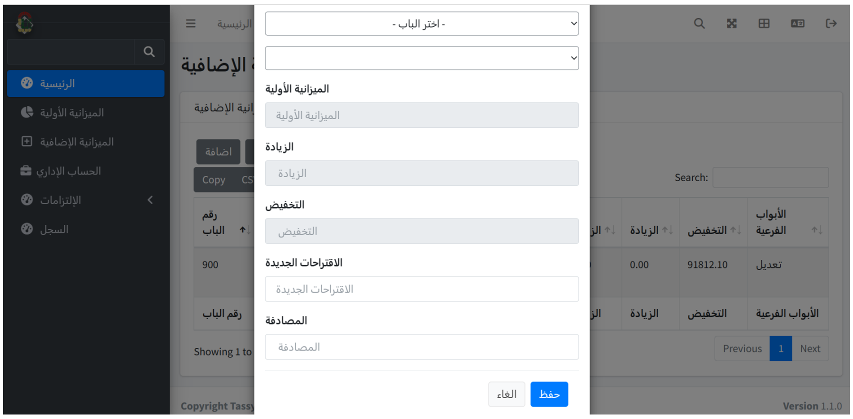
All of these can be found in the extra budget and administrative account interfaces



رقم الباب	اسم الباب	رقم المادة	اسم المادة	التذكير بالحساب الاداري	الاقتراحات الجديدة	الأبواب الفرعية
900	المصالح المالية	629	المصاريف الطارئة	91812.03	912.10	تعديل
902	وسائل الإدارة العامة ومصالحها	611	الصيانة والترميم في المؤسسة	812.00	91812.01	تعديل
914	الشبيبة و الرياضة والثقافة	632	الأجور المختلفة	91812.03	121.86	تعديل
رقم الباب	اسم الباب	رقم المادة	اسم المادة	التذكير بالحساب الاداري	الاقتراحات الجديدة	الأبواب الفرعية

Showing 1 to 3 of 3 entries

Figure IV.14: initial Budget View



- اختر الباب -

الميزانية الأولية: الميزانية الأولية

الزيادة: الزيادة

التخفيض: التخفيض

الاقتراحات الجديدة: الاقتراحات الجديدة

المصادقة: المصادقة

الغاء | حفظ

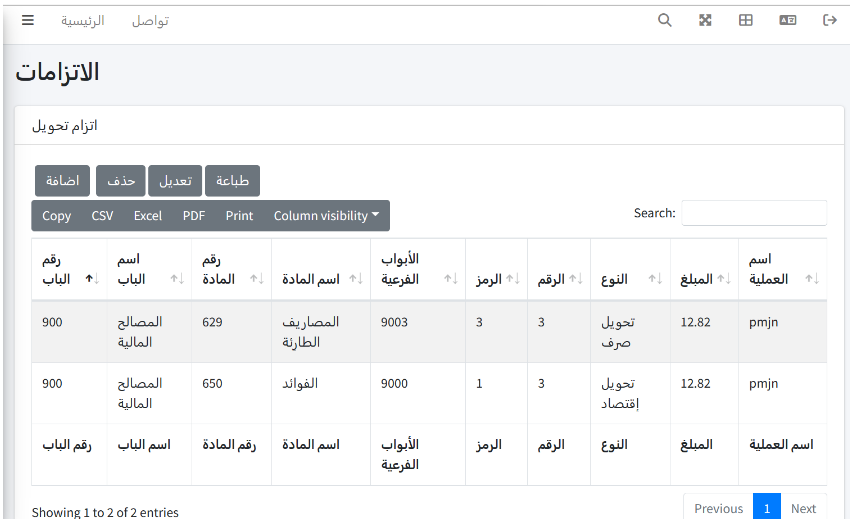
رقم الباب	الزيادة	التخفيض	الأبواب الفرعية
900	0.00	91812.10	تعديل
رقم الباب	الزيادة	التخفيض	الأبواب الفرعية

Figure IV.15: Adding Extra Budget View

4.4 Commitments interfaces:

Moreover there are the commitment functions which can be categorized into 2 types, normal commitment and Transfer commitment.

The following interfaces simplify the working of these functions in addition to some features.



الالتزامات

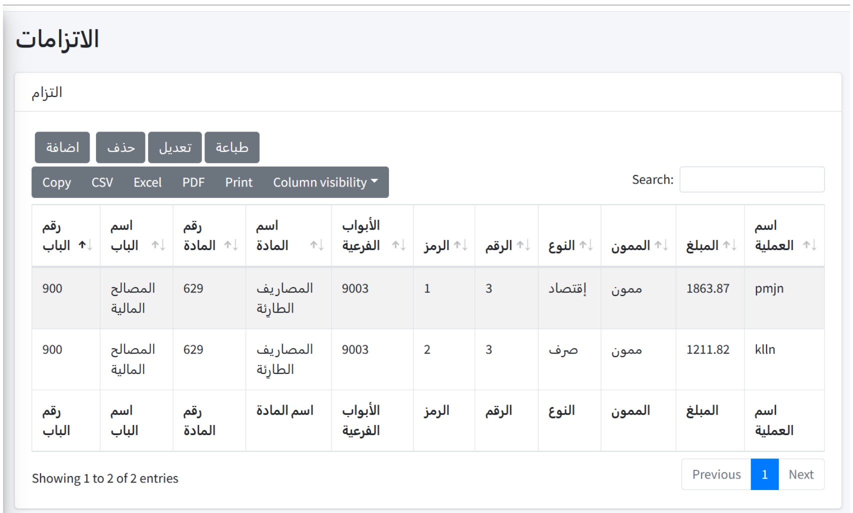
اتزام تحويل

Search:

رقم الباب	اسم الباب	رقم المادة	اسم المادة	الأبواب الفرعية	الرمز	الرقم	النوع	المبلغ	اسم العملية
900	المصالح المالية	629	المصاريف الطارئة	9003	3	3	تحويل صرف	12.82	pmjn
900	المصالح المالية	650	الفوائد	9000	1	3	تحويل إقتصاد	12.82	pmjn
رقم الباب	اسم الباب	رقم المادة	اسم المادة	الأبواب الفرعية	الرمز	الرقم	النوع	المبلغ	اسم العملية

Showing 1 to 2 of 2 entries

Figure IV.16: normal Commitment View



الالتزامات

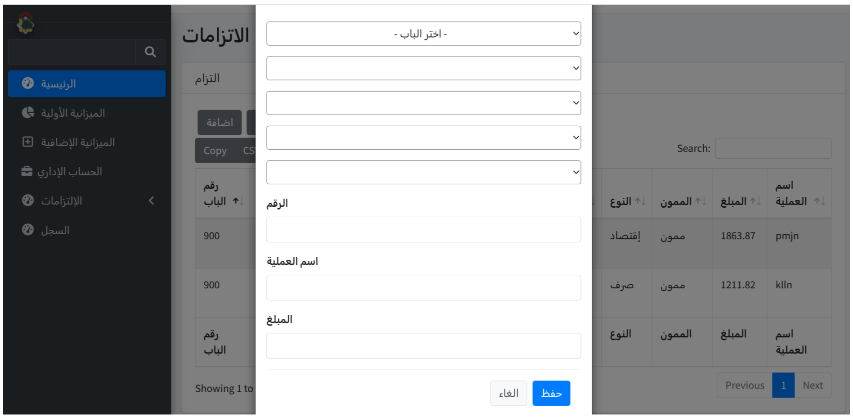
التزام

Search:

رقم الباب	اسم الباب	رقم المادة	اسم المادة	الأبواب الفرعية	الرمز	الرقم	النوع	الممون	المبلغ	اسم العملية
900	المصالح المالية	629	المصاريف الطارئة	9003	1	3	إقتصاد	ممون	1863.87	pmjn
900	المصالح المالية	629	المصاريف الطارئة	9003	2	3	صرف	ممون	1211.82	klln
رقم الباب	اسم الباب	رقم المادة	اسم المادة	الأبواب الفرعية	الرمز	الرقم	النوع	الممون	المبلغ	اسم العملية

Showing 1 to 2 of 2 entries

Figure IV.17: Transfer Commitment View



- اختر الباب -

الرقم

اسم العملية

المبلغ

Figure IV.18: Adding Transfer Commitment View

5 Conclusion

During this chapter we have introduced several development tools that It allows us to implement our project, so we introduced as many concepts as possible related to the development environment, such as MySQL, Xampp, Laravel, PHP, ctc. We've provided our database tables as relatioanl model and some code explanations, and finished with some examples From the interfaces of the application that we have created.

General Conclusion

The objective of our work was to design and realize a web application multiplatform for municipality budget management. This will allow us to simplify, numerize and decentralize its features. This should make the process more efficient and cost-effective. It should also provide more transparency and accountability in the management of public funds. Finally, it should enable better decision-making by providing real-time access to budget data.

To complete our project, we briefly introduced the municipality structure and budget documents, which is our first goal. We also introduced the used software in budget management to identify the problem, which led us to draw from it the objectives to be achieved in our application.

Then we have given a general overview of the web application systems and the technologies that make it possible to build multiplatform applications.

For this realization, we have become aware of several areas such as: the 3-tier and MVC architecture, budget management and the accounting work in the municipality. It also allowed us to familiarize ourselves with a number of IT and development tools, such as the Laravel and Bootstrap frameworks, Javascript, Ajax and Jquery for the treatment and interactions, Apache server for the database connection.

While our project has made significant strides in enhancing municipal budget management through the development of a web application. However, we recognize that the journey towards comprehensive and efficient budget management is ongoing. As with any project, there is always room for improvement and development.

At this stage, our web application has successfully addressed key challenges such as streamlining financial processes, improving transparency and accountability, optimizing resource allocation, and enhancing community engagement. However, we acknowledge that important aspects of budget management procedures remain to be integrated into the application.

Moving forward, our roadmap includes several essential steps to further enhance our web application's capabilities and impact. These include:
-Completing the Budget Management Procedure
-Implement security and Data Protection
-Provide statistical Views and Analytics
-Deployment with the Latest Tools and Technologies

In order to continue working on this project we have to become aware of several other areas such as the network, remote access to databases and their manipulations and security implementation.

References

[1] Ben-Aiach and Samir. The contribution of municipal property revenues in algeria to the revenues of their budgets for management. *El-Tawassol" "*, 29(9): 1–15, 2023. 6, 9

[2] Jihed Touzi. *Aide à la conception de système d'information collaboratif, support de l'interopérabilité des entreprises.* PhD thesis, 2007. 15

[3] Guerri 2022. Conception et réalisation d'une application web pour la gestion de stock et de vente et d'achat des dattes.master 2 thesis university of biskra. 15, 16

[4] Elisabeth Hermann. Les applications mobiles.article. 2017. 17

[5] Appsierra. 1 tier architecture, 2 tier architecture, 3 tier architecture - appsierra — appsierra.com. https://www.appsierra.com/blog/tiers-in-software-architecture, 2022. 17, 18, 20

[6] Houssameddine Mellal. Développer une application 3-tiers. 2016. x, 18, 20

[7] Mohamed Yacine Saidi and Laurent Gizon. *Developpment d'une interface logicielle pour le calcul scientifique a distance : Application Al'HELIOSISMOLOGIE.* PhD thesis, Universités Paris-Sud XI, 2006. 19

[8] Channu Kambalyal. 3-tier architecture. *Retrieved On*, 2(34):2010, 2010. x, 19

[9] Zulfahmi Syahputra. Website based sales information system with the concepto of mvc (model view controller): Website based sales information system with the concept of mvc (model view controller). *Jurnal Mantik*, 4(2):1133–1137, 2020. 20

[10] MI Kausar Bagwan and PD Swati Ghule. A modern review on laravel-php framework. *Ire Journals*, 2(12):1–3, 2019. 21, 43

[11] Hillar Petersen. From static and dynamic websites to static site generators. *University of tartun, Institute of Computer Science*, 2016. x, 23, 24

[12] Samira Kadouci 2015 et al. *Conception et réalisation d'une application des inventaires avec code-barres pour l'entreprise portuaire de ghazouat tlemcen.* PhD thesis. 27

[13] Laurent Audibert. Uml 2. *Institut universitaire de technologie de villetaneuse–département informatique*, 2007. 29, 30, 34

[14] Luke Welling and Laura Thomson. *PHP and MySQL web development.* Sams publishing, 2003. 40

[15] Binildas Christudas and Binildas Christudas. *MySQL.* Springer, 2019. 41

[16] Dalibor D Dvorski. Installing, configuring, and developing with xampp. *Skills Canada*, 2007. 42

[17] Robert Labus. *Social network for publishing and editing image files.* PhD thesis, Josip Juraj Strossmayer University of Osijek. Faculty of Electrical . . . , 2018. xi, 43

[18] Suraj Shahu Gaikwad and Pratibha Adkar. A review paper on bootstrap framework. *IRE Journals*, 2(10):349–351, 2019. 44