People's Democratic Republic of Algeria

Ministry of Higher Education and Scientific Research

University August 20, 1955 Skikda

Faculty of Science

Department of Computer Science

End-of-study dissertation

For obtaining a Master's degree

Sector: Computer science

Speciality: Networks and Distributed Systems

Theme here

Integration electronic payment

Presented by:

Hichem kadissa

Supervised by:

Dr. Boulehouache Soufiane

Academic year: 2024 / 2025

General introduction

With globalization and the emergence of new information technologies, electronic payment is emerging as a new Internet service that improves customer service while reducing customer and supplier service expenses. Thus, quality and efficiency can be .improved

Online electronic payment is actually a service that takes place over the Internet, partially or completely automated, using a computer, smartphone or electronic tablet. The electronic payment system via the Internet can perform financial transactions remotely without the need to go in person to carry out financial transactions

Electronic payment, often referred to as e-payment, is a modern and convenient method of conducting financial transactions using digital technology and electronic systems. This form of payment has become increasingly popular in recent years, revolutionizing the way individuals, businesses, and organizations exchange money for goods and services. Unlike traditional payment methods, such as cash or checks, electronic payments rely on electronic devices and online networks to transfer funds securely and efficiently. This introduction will provide an overview of electronic payment, highlighting its key features, benefits, and significance in today's digital economy.

This dissertation, structured around 4 chapters, presents the stages of carrying out the work. :

Chapter One: Related to electronic payment systems and what follows Characteristics, advantages and disadvantages of these systems and why they are We also talk about some online appointment booking ،connected to the Internet systems in Algeria, electronic payment and Introducing us to the electronic payment system on our website rdv-rendezvous.net

The second chapter: In this chapter, we will define the key concepts on on which our web application is based, we talk about Web and these different technologies such as development languages, frameworks, databases web protocols...etc، let's also talk about client-server architectures

General introduction

The third chapter: in this chapter we present the design of ours web application, using the different tools of the UML design method.

fourth chapter: constitutes the realization and implementation part of our web application and the presentation of the hardware and software environment used, with the presentation of the technical choices and the production tools and the different parts of the web application.

In addition to these chapters my dissertation must contain a general conclusion and perspective to summarize our work.

Chapter 01
electronic payment

Chapter 01: electronic payment

1.Introduction:

Electronic payment refers to the use of digital technologies to transfer funds from one bank account to another, and it has become an increasingly popular way to make financial transactions in recent years.

The rise of electronic payment methods has transformed the way that people conduct business, as it allows for faster, more secure, and more convenient transactions. With the growing popularity of online shopping and other digital services, electronic payment has become an essential part of modern commerce, making it easier than ever before to transfer money between individuals and businesses across the globe.

This report will examine the benefits of electronic payment, the different types of electronic payment available, and the challenges that must be overcome to ensure its widespread adoption.

Chapter 01: electronic payment

2-What is Electronic payment?

Figure1.1: An image showing the different types of electronic payment.

2.1Definition 01:

Electronic payment is a means of carrying out commercial transactions for the exchange of goods or services on the Internet

Currently, it is very well established and used by the majority of people and companies with an internet business [Web1]

2.2Definition 02:

electronic payment is the transfer of funds through electronic or digital medium via the internet. There are different forms of online payments, including mobile pay, digital wallets, e-banking, bank transfers, and many more.[Web1.2]

2.3Definition 03:

Payment is the last step in the business transaction processing cycle. Given the significant effort put into the initial phase of processing accounts payable, which includes receiving invoices, organizations need to be able to exchange payments in a timely and cost-effective manner. Electronic payment methods provide faster, cheaper and more convenient processing than checks. [Web1.3]

2.4 Definition 04:

E-payments are an electronic or digital way of transferring funds. Essentially, you can utilise electronic payment methods to transfer funds as an alternative to cash.

3.Advantage of Electronic payment

The following some of critical advantages of E-payment

3.1.Time-Saving

E-payments enable you to make purchases with a simple tap or swipe. Transactions are processed and completed within a couple of minutes. While it is faster than paying with a paper check or other instruments, it also saves you the time and hassles associated with arranging cash.

3.2Efficient

With electronic payment systems, you do not have to wait in long queues at ATMs or bank branches to withdraw cash. The lines at checkout counters are also shorter, with each transaction taking less time. You can also use these online payment systems to pay for a wide variety of products on online shopping websites, thus eliminating the need to visit stores physically.

3.3.Cashless Economy

Another advantage of e-payments is that it helps build a cashless economy, especially in the urban areas of the country, by reducing the reliance on cash. Reduced cash usage in the urban sectors enables banks to distribute more cash in the rural parts of the nation where e payments are uncommon.

3.4 Security

Cash transactions bring their own set of risks, such as robbery, misplacement, or other similar incidents. However, electronic payment systems come equipped with security protocols that ensure the safety of your funds. Banks use highly secure practices like two-factor authentication, PIN (Personal Identification Numbers) and OTPs (One Time Passwords) to protect your funds from thefts or fraudulent activities.

3.5 Certainty

The payments made using e-payment methods reflect in your bank statement or digital wallets. You also receive instant e-mails and SMS alerts after every transaction. You can check for the credit/debit of funds in your account based on the chosen method of e-payment. In case funds are debited wrongly, the transaction is reversed within 24-48 hours.[Web1.4]

4.Types of electronic payment ?

There are essentially three types of electronic payments:

4.1 Card payments

Cards are the most familiar type of electronic payment the world over, though they are preferred mainly by older millennials and up, according to a 2016 study. Cards remain attractive in part due to the rewards they offer, which have been increasing for the past several years. Major credit & debit cards in all corners of the world include Visa, Mastercard, American Express, Discover, and Diners Club.

Figure1.2:master card for bank of bnp Paribas.

Figure1.3:visacard for bank of paysersa.

4.2 Bank transfer payments

The transfer of money from one bank account to another can be done in several ways. ACH transfers are one type of bank transfer payment specific to the U.S. Similar to direct deposit, money is electronically withdrawn from the shopper's bank account, transferred to an ACH network, and then deposited into a merchant's bank account. Bank transfers happen over the world and are referred to by different names. Sofort is a popular real-time bank transfer payment method for several countries in Europe, and Giropay is Germany's most-used payment method.

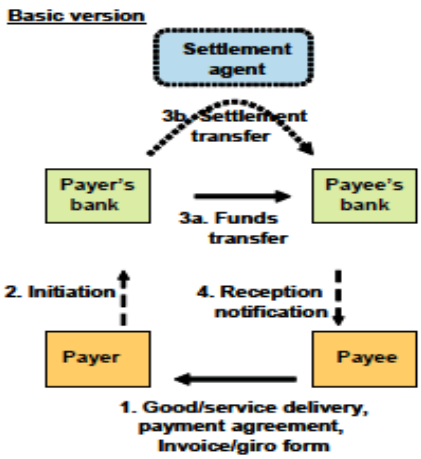


Figure1.4: An image showing a method of transferring money from one bank to another.

4.3 EWallet payments

eWallets are poised to take over the payments scene, according to Citibank's Michael Corbat (and lots of other people in the know). EWallets store your credit or debit card information in a mobile device; you can use the eWallet either by holding your phone near a special payment terminal in-store or by using the eWallet app for in-app or web purchases. Varying forms of eWallets are used all over the world, so if you're not offering them as a payment method, you're missing out.

These are online accounts or platforms where the customer can either deposit money into the account or link it to a bank account. Some of the most popular e-wallets are PayPal, Neteller, and Skrill.[Web2]

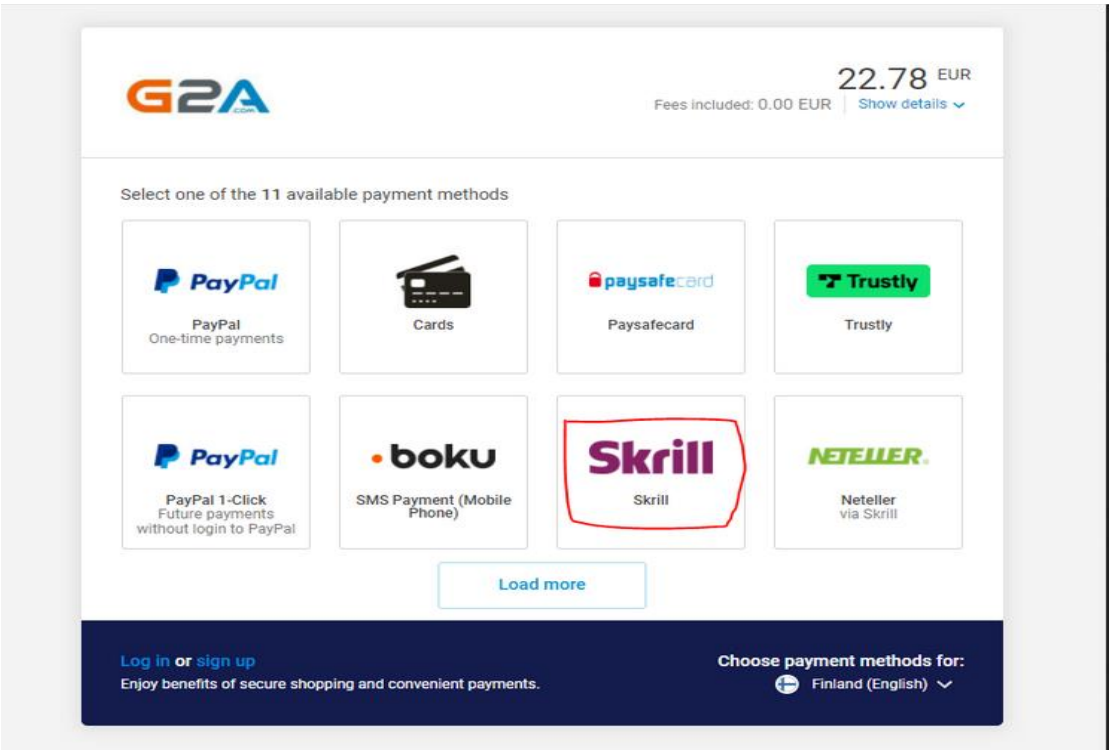


Figure1.5: An image showing the types of ewallet used in electronic payment.

5.The Electronic payment how It's work?

There are essentially six steps to process a payment…

- **Purchase:** Customer finds something they want to buy and enters their payment information.
- **Encryption:** A **payment gateway** encrypts the data as part of PCI compliance.
- **Verification:** The **payment processor** then verifies the customer's data and transaction information is correct.
- **Authorization:** The **payment processor** sends the transaction information to the customer's bank or credit card to approve or deny the funds.
- **Transfer:** Once the bank or credit card approves the transaction, the **payment processor relays that information to the payment gateway** and requests the transfer of funds from the customer's bank to the merchant's bank.
- **Transaction complete:** The merchant's bank receives the funds and notifies the customer that the transaction is complete via the payment gateway.[2.2]

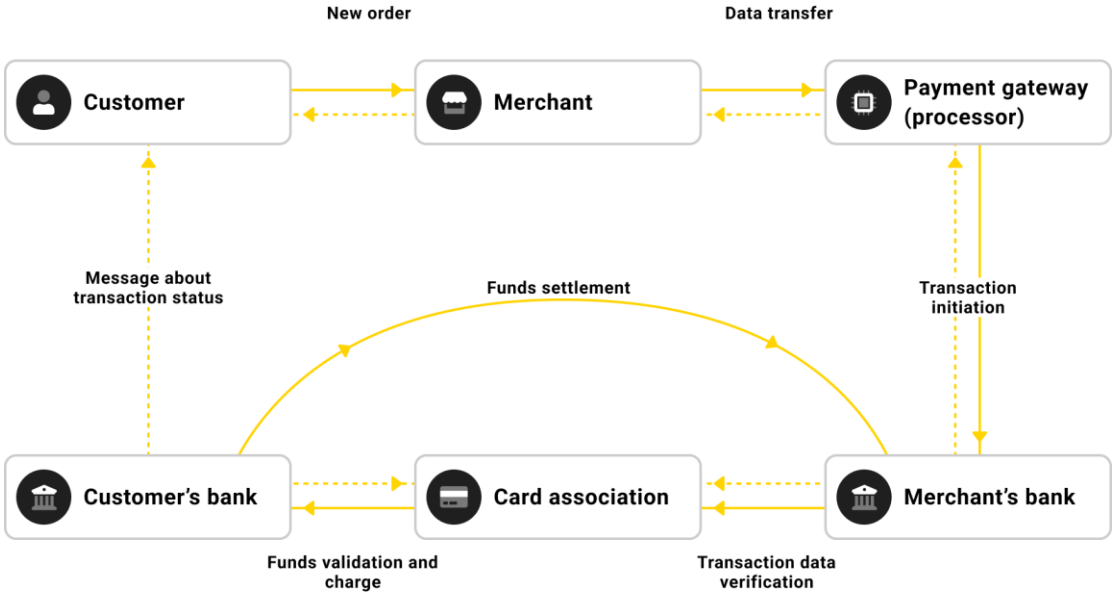


Figure 1.6: Image showing how electronic payment works.

6.Some solution of electronic payment:

6.1 paypal banking

Is American company providing a world wide online payment service system.the plateform serves as an alternaternative to payment by check or credit card**[Web3]**

Figure 1.7:logo paypal

6.2 Skrill Banking:

skrill is british company created in 2001 offerring an online banking and payment service , it allows you to send and receive money instantly as of octobre 12 2012it's claims 30 million registered users the service works with mutitude of currencies**[Web3.2]**

Figure1.8: skrill logo

6.3 neteller:

Neteller is electronic money transfer service used to transfer money to and from merchants such a forex trading companies social media companies**[Web3.3]**

Figure1.9: neteller logo

7.Electronic means of payment in Algeria:

According to article 06 of Law No. 18-05 relating to electronic commerce, a means of electronic payment is defined as: "any payment instrument, authorized in accordance with the legislation in force, allowing its holder to make payments proximity or remotely through an electronic system" (OFFICIAL JOURNAL OF THE ALGERIAN REPUBLIC N° 28, May 16, 2018, P 05).

To paraphrase what the previous law stipulates we can present the means of

electronic payment like any dematerialized medium (in the form of a CIB card) made available available to customers to allow them to pay with their purchases in storeslocally or remotely through an SPE in complete security without using cash. According to article 27 of the same previous law stipulates that "the payment of transactions electronic commerce is carried out, remotely or upon delivery of the product, by the means payment authorized in accordance with current legislation. When payment is electronic, it is carried out through dedicated payment platforms, set up and operated exclusively by banks approved by the Bank of Algeria and Algérie Poste and connected to any type of electronic payment terminal via the

public operator's network telecommunications. Payment for cross-border commercial transactions is carried out exclusively remotely by means of electronic communications" (JOURNAL OFFICIAL OF THE ALGERIAN REPUBLIC No. 28 of article 27, May 16, 2018, PP 07-08).

Furthermore, the electronic media available in Algeria are as follows:

7.1 CIB bank cards

The CIB card is an interbank card, it is recognizable by the CIB logo of interbank electronic banking. It also contains the logo and name of the issuing bank. from the menu. The card is equipped with a microprocessor commonly called a "chip" which manages and secures payment transactions, it allows its holder to pay for their purchases with various retail businesses and major billers. This instrument can provide a single service which is withdrawal or payment and withdrawal (simultaneously) interbank domestic which is accepted by merchants affiliated to the interbank electronic payment network and on all ATMs installed on the national territory (Refafa Brahim, 01/12/2020/ p 302). Her Validity is two years, automatically renewable. The CIB card distributed by the SATIM can be used in all distributors and retailers equipped with POS. He

There are two types of CIB cards (Figure N°1):

- **Classic CIB Card:** Is a payment and withdrawal card offered to customers according to the criteria established by each bank, with a comfortable ceiling which offers a large ability to pay and withdraw, in complete security, from merchants and different ATMs/ATMs and transaction speed.
- **CIB Gold card:** It has the same criteria as the classic CIB card, but with additional features and higher withdrawal and payment limits than the classic card. The same design, the color is different. Through the figure Next we can draw the first difference that jumps out at the eye is the color: the gold is generally presented in gold, on the other hand the classic can be blue or a another color and this differs from one bank to another. Other differences exist, the gold is intended for a certain income range (differs from one bank to another greater than or equal to that of the classic CIB. As there are business cards classic and gold intended for professionals and businesses.

Figure 1.10 :CIB classic Card and CIB golden Card

7.2 The EDAHABIA card

It is an electronic payment and withdrawal card under the EMV system (ensuring security transactions to its holders), allowing various withdrawal and transfer operations to be carried out. payment on the account (CCP), on bank automatic teller machines (ATM) and also on electronic payment terminals (TPE) (www.poste.dz). This card is distributed by Algeria posts for these customers free of charge following their requests. What you need to know is to obtain a CIB/EDAHABIA card you must first be holder of a bank account/CCP account with one of the banks members of the electronic banking network including Algeria Post. It should also be noted that the CIB bank cards and EDAHABIA card are credit cards. debit do not have credit, which means that payment will only take place if the customer has the money in their account, and expenses are deducted as they occur on your current account (bank or ccp).

7.3The Visa and Mastercard credit card in Algeria

The VISA card is a means of withdrawal and electronic payment, allowing you to make currency transactions on ATMs and POS as well as the Internet in all countries around the world. A Visa or Mastercard credit card is a card that allows you to make purchases in online and in supermarkets. Visa or Mastercard credit cards have the same use. The Visa credit card followed by the Mastercard card were introduced a few years ago. years, by several banks (BEA, BNP, BDL, etc.). There are several types of Visa and Mastercard credit cards: Gold card, prepaid card. The Prepaid card is the most used, the customer can load it with a fixed amount and he will not be able to not exceed the amount loaded on the card. For the Gold card and other cards, they

debit the bank account directly. In Algeria Visa and Mastercard are valid only during a trip abroad or for purchases on the Internet. During this year (2021), the total number of cards in circulation reached 10,712,133, an increase of +7.49% compared to the 1st quarter of 2021,

The distribution of cards by category is shown in the following pie chart:

Figure 1.11: distribution of payment cards in Algeria in 2021

We notice that the business card is practically non-existent on the diagram, this is interpreted by the number of the latter, which is put into circulation remains very low. In On the other hand, the interbank number (CIB) only represents 16% of the total number of cards for the year 2021, followed by Visa and MasterCard cards by 15% these proportions remain marginal compared to the share of the EDAHABIA card which represents more than half of all cards (69%) are equivalent to 7,363,219 cards.

7.4 The launch of e-payment in Algeria

Electronic payment or online payment is one of the services created by the GIE has been officially launched on October 4, 2016 and implemented by 11 banks and 9 companies and 9 web merchants offering this service to their customers. Concerning the banks there are six banks public; BADR, CPA, BDL, BNA, BEA, CNEP, and five other private companies; Trust Bank Algeria, Natixis Algeria, Société Générale Algeria, Gulf Bank Algeria and Al Baraka. And for the companies, These are: Air Algérie, Tassili Airlines, Djezzy, Algérie Télécom; Mobilis, SEDAL, CNAS, OOREDOO and the insurer Amana.

However, as for the platform of the Algerian electronic payment system, it is based mainly on the use of the CIB card in these two forms, one provided by the banks and the second by Algérie Poste, and this, through an entire network, as presented in the current time. Since October 2016, online payment by CIB card has been officially operational in Algeria. In a first phase, the service was opened for large billers, and Today, 105 web merchants are members of

the CIB internet payment system. To date, the overall number of transactions is 8,460,613 since the launch of the latter, A figure which is certainly not substantial given the importance and ease of this new payment system. A system that still remains unknown to the Algerian consumer who does not venture to type the code of their bank card on a small device in order to pay his purchases. Following the accession of Algérie Poste to the Monétique GIE and the realization of CIB/EDAHABIA interoperability, entered into production since January 5, 2020, the published statistics regarding card transactions will now include the electronic payment activity carried out by the CIB card and the EDAHABIA card.

In order to give an overview of the evolution of online e-payment since its launch until 2019, we will rely on the following figure:

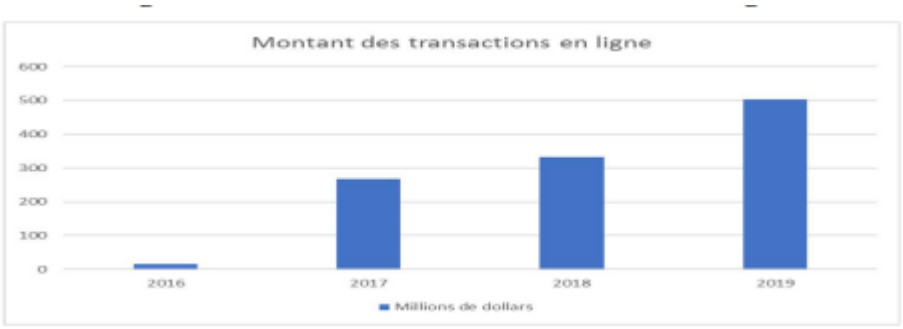


Figure 1.12: amount of online transactions.

We can notice that since the launch of e-payment the amount of transactions carried out follows an exponential trend over the years. Online payment in Algeria relies mainly on the telecommunications sector, which includes that of fixed and mobile telephony. On average, this sector represents 86% since the establishment of this payment method. The second sector is electricity and water where it has been permitted since 2016 citizens to pay their bills by EDHAHABIA card and the CIB card. The amount average transaction is a little more than 2,200 dinars over the period 2016-2019 and less of 1,500 dinars for the first four months of 2020.

7.5 The launch of mobile payment (M-payment) in algeria

Mobile payment, a service recently launched in Algeria at the beginning of 2020 by the GIE Monétique allowing consumers to pay for their purchases from a telephone mobile using an application designed by the GIE for smartphones, which will be connected to the network interbank to be able to carry out transactions which will be debited on the card interbank (CIB) or that of Algeria Post, Golden (EDAHABIA).

Algeria has moved towards the adoption of M-payment to further encourage the adoption of e-payment among Algerians. So, with M-payment, VSEs are no longer essential in order to carry out electronic payment transactions in spaces physical shopping mall, just a phone and a QR code displayed in the store or in a restaurant to make payments, and this can reduce the procedure and steps for economic operators in order to offer electronic payments - by mobile for their customers.

M-payment is a very fast and secure payment solution, but which remains at the stage primitive in Algeria. On the other hand, in a country like Algeria where the popularity of mobile telephony has reached extraordinary levels, the launch of Mpayment services will help boost the positive contribution of banks to the country's economy. Today, the mobility dimension has become a priority for providers of solutions e- payment.

The latter must admit that the number of mobile terminals has far exceeded the number of bank accounts. **[Web3.4]**

8 Conclusion:

electronic payment has transformed the way people conduct financial transactions, making it more convenient, secure, and efficient.

As technology advances, electronic payment systems are becoming more sophisticated and user-friendly, allowing more people to access these services.

However, as with any technology, there are risks associated with electronic payment, such as security vulnerabilities and data breaches.

As such, it is important for both consumers and businesses to exercise caution and take necessary steps to protect their financial information. Overall, electronic payment is here to stay and will .continue to play an important role in the way people buy and sell goods and services.

Chapter 02:

web technologies

Chapter 02: web technologies

1. Introduction

In this chapter, we will define the key concepts on which our web application is based. After a quick and simple introduction to web technologies, we start this chapter with the web (definition of the web, the evolution of the web and static and dynamic) then let's move on to the different Web technologies (Web browsers, Development languages, Frameworks, Database, Data formats and web protocols) then talks about Client / Server architecture and its types, and ends this chapter with a small conclusion

2. Web Technologies

Web technologies are the various tools and techniques used in the process of communication between different types of devices over the Internet. To better understand this term, let's break it down into two parts: 'web' and 'technology'.

2.1. The Web

The Web (referred to as the World Wide Web, more commonly known as WWW or W3) is a system of public web pages interconnected across the Internet. The web and the internet are not the same thing: the web is one of many applications built on top of the internet.

Tim Berners-Lee proposed the architecture of what was later known as the World Wide Web. He created the first web server, web browser, and web page on his computer at the CERN scientific research laboratory in 1990. In 1991, he announced his creation on the alt.hypertext newsgroup, indicating for the first time that the web was made public [web4].

What we know as "the web" consists of several elements:

The HTTP protocol (HyperText Transfer Protocol): governs the transmission of data between a server and a client. It defines two concepts:

Chapter 02: web technologies

URL (Uniform Resource Locator): is an address that specifies the location of an Internet resource by indicating the protocol to be adopted, the name of the machine, .the access path and the name of the file

HTML (hypertext markup language): is the starting point of the WWW. It is a client-side language used to code a website's interface. It is used to define how the .web page will be structured, using elements identified by tags

The evolution of the web .2.1.1

Since its launch in the early 1990s, the Internet has continued to reinvent itself. Driven both by technological developments and by new uses, the Internet has passed four main milestones, from Web 1.0 to Web 4.0. In the same way, digital marketing .has had to adapt to meet the expectations of Internet users as closely as possible

Web 1.0 (traditional web): Starts in the 1990s. It is above all a static web, centered on the distribution of information. Mostly associated with large corporations. It is characterized by product-oriented sites, which require little user intervention. It is .often the online transcription of paper catalogs

Web 2.0 (social web): It favors the dimension of sharing and exchanging information and content (texts, videos, images or others). He sees the emergence of social networks, smartphones and blogs. The web is becoming more democratic and dynamic. It’s the time of forums, wikis, etc. The opinion of the consumer is constantly .solicited and he takes a liking to this virtual socialization

Web 3.0 (semantic web): Aims to organize the mass of available information according to the context and needs of each user, taking into account their location, preferences, etc. It is a web that attempts to make sense of data. Algorithms reign supreme. It is also a more portable web that increasingly makes the link between the real world and the virtual world and meets the needs of mobile users, always .connected through a multitude of media and smart or fun applications

Web 4.0 (intelligent web): Is the logical evolution of the semantic web. It frightens as much as it fascinates, since it aims to immerse the individual in an environment

increasingly digital. Based on wireless communication connecting people and objects at any time and any place in the physical or virtual world in real time, web 4.0 pushes the path of personalization opened up by web 3.0 to its climax. But at the same time it raises many questions regarding the protection of privacy, data control, etc. [web5]

2.1.2. Static and dynamic websites

Websites come in different shapes and sizes, but they can be divided into static sites and dynamic sites. The difference isn't obvious when you just look at a website in your browser, but it can make a big difference in how your website functions.

2.1.2.1. Static websites

Static websites contain a fixed number of pages and the web page format is fixed and provides information to the customer. The content of a web page is modified during the execution of the page on the client's browser. This type of websites created from HTML and CSS coding on a simple text editor like notepad

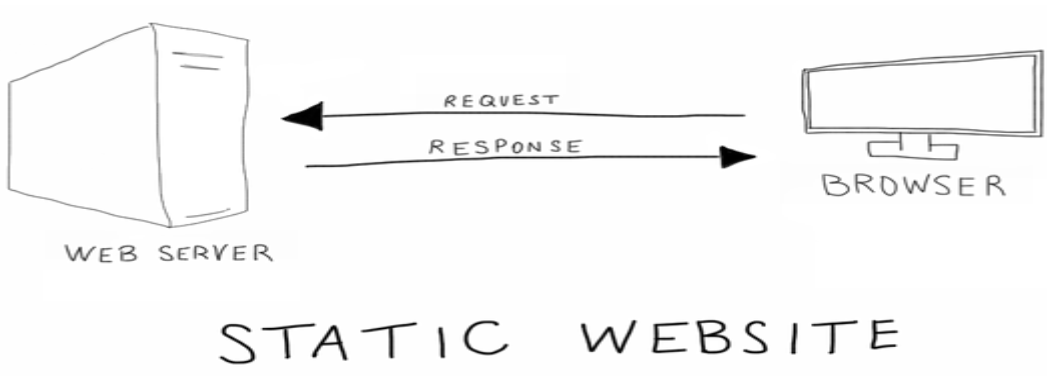


Figure 2.1 :transfer with static site

2.1.2.2Dynamic websites

Dynamic websites can change the content of a web page dynamically when the page is executed on the client's browser. This type of website uses server-side

programming such as PHP, ASP.NET.and JSP etc. to modify the page content at runtime. Dynamic websites use client-side scripting to prepare a dynamic design and server-side code to handle events, manage sessions and cookies, and store and retrieve data from the database [web6].

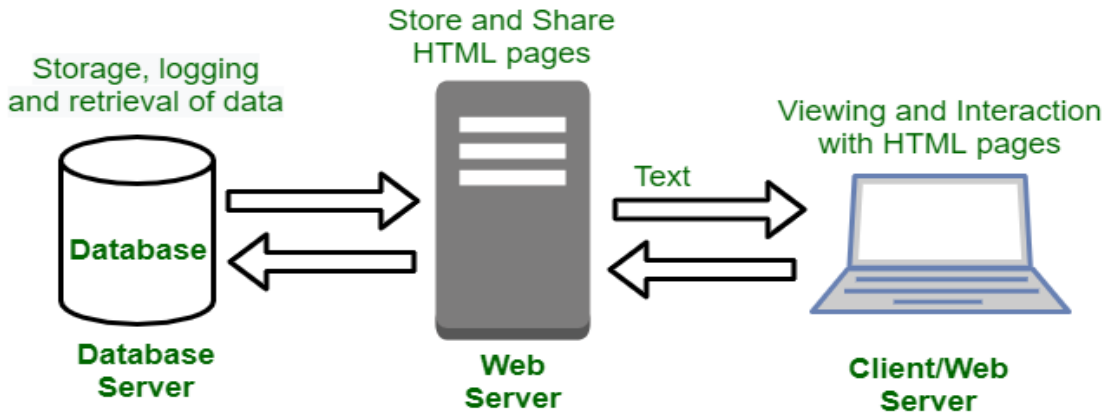


figure 2.2: transfer with dynamic site

2.1.3. Web applications

A web application refers to application software hosted on a server and accessible via a web browser.

Unlike traditional software, the user of a web application does not need to install it on his computer. All they have to do is connect to the application using their favorite browser. The current trend is to offer a user experience and functionality equivalent to software installed directly on computers. The technologies used to develop web applications are the same as those used in the creation of websites.

Cloud computing is therefore this rapidly expanding phenomenon which aims to evolve the traditional software model towards the Internet and mobile telephony is integrated as an extension of use to this model [web7].

Figure2.3: Web application

2.2. The different types of web technologies

Since we have limited space to review the wide range of web technologies available, we have selected a few that are most often used

2.2.1. Web browsers

Web browsers, often simply called browsers, allow us to view all the resources that are part of the World Wide Web. They are based on a client-server architecture. The client is the browser in this scenario. You can think of the server as a combination of software and hardware that receives requests from the client and then sends the requested resource to the client [web8].

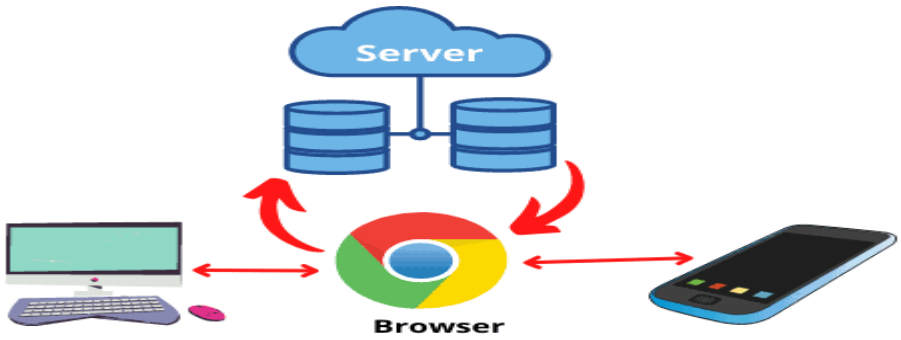


Figure 2.4:relation between browser and servers and devices

2.2.2. Frontend Vs Backend Development

Chapter 02: web technologies

Web development refers to the process of creating websites. This process is based on a number of steps, a website is made up of two distinct parts: the frontend and the backend.

Frontend: refers to all parts of a website that a user can see on their screen and interact with.

Backend: refers to the exact opposite of this. It involves the hidden mechanisms that make a web page work. A typical user is usually unaware of what is happening at the backend.

Different web technologies are used in the web design and development process, the two tables below represent a brief overview of the differences between these two types of web development

Frontend	Backend
Client side	Server side
Browser	Web server
Conception of website	Database
UI/UX	The servers
Languages:HTMl,CSS,JavaScript	Languages:PHP,JAVA,python,C#

Table2.1:diferent between frontend and backend

2.2.3. Web development languages

Programming languages are the most fundamental component of website building. To code a website, you need to be familiar with some programming concepts, some languages and their syntax, and a good IDE, below we locate some commonly used web development languages:

HTML: HyperText Markup Language (HTML) is the code used to structure a web page and its content. For example, the content of your page could be structured in a set of paragraphs, a bulleted list or with images and tables of data. As the title suggests, this article provides you with the basics of understanding HTML and its functions [web9].

Chapter 02: web technologies

CSS: CSS is one of the main languages of the open Web and has been standardized by the W3C. This standard evolves in the form of levels (Levels), CSS1 is now considered obsolete, CSS2.1 corresponds to the recommendation and CSS3, which is divided into smaller modules, is in the process of standardization [web10].

JavaScript: JavaScript is one of the most popular languages of the last decade. It allows developers to make the frontend and backend with the same syntax, which significantly reduces the workload.

Express.JS and Node.js help manage and run both ends of the application, while APIs help in faster and easier application development.

Python: Python was created in 1991, and it has established itself as an excellent general-purpose language ever since. It offers a very clean and easy backend development environment. According to a 2020 Stack Overflow survey, it was the 3rd most loved programming language by developers.

Ruby: This programming language was developed in 1990 by a Japanese programming expert. The advantage of this language is that it has a syntax similar to that of Python and Java. On top of that, it offers great possibilities for automation. This is why platforms like Airbnb and Esty use it for automation purposes.

PHP: It was developed in 1994, and since then it has become the best server-side development language in the world. According to a study by W3Tech, around 79% of websites in the world are created using PHP.

JAVA: Java was created in 1991, but it was officially released in 1995, and since then it has established itself as one of the best programming languages in the world. It is also ranked as the second best programming language in the TIOBE Index 2021 rankings. Additionally, it is the best option for creating mobile applications.

C#: C-Sharp is one of the most popular languages for creating the backend of a system. This is because of its amazing features like automation on Windows servers. Besides that, it is great because it executes the codes very fast. It can also be used for game development and CLI application creation.

Perl: Perl was developed about three decades ago. It is a language that still works exceptionally well where it is needed. The interesting thing about Perl is that it

was ranked as one of the highest paying languages by a Stack Overflow developer survey. Perl 5 is the version that is still widely used for prototyping and automation [web11]

2.2.4. Web development frameworks

In simple terms, a framework can be defined as a skeleton of code. Think of it as a rough outline without fine details. A framework defines the basic functions or tasks that code is supposed to perform. However, in order to tell this particular program how to perform these functions, you will need to write more detailed specific instructions. There are many web frameworks for frontend and backend development including the following:

2.2.4.1 Frontend Frameworks

Angular: Angular is a front-end framework specialized in building rich single-page applications. It is a dynamic framework capable of building complete client-side applications. Angular 1.x used Javascript, but later versions adopted TypeScript, which is a superset of JavaScript.

Vue: Vue.js started as an individual project and quickly grew to become one of the hottest JS frameworks. Vue has many advantages. First of all, it's a progressive framework, which means you can adopt Vue for part of an existing project, and everything will work just fine.

React: React is not a framework, it is a front-end library. React was the first to adopt the component-based architecture that Angular and Vue, along with many other frameworks, began to adopt later. React's virtual dom makes dom manipulation much faster, and it's quite easy to pick up, mainly thanks to its JSX [web12] syntax.

2.2.4.2 Backend Frameworks

Express: Thanks to the skyrocketing popularity of Node.js, Express has quickly become one of the best web development frameworks, and it's also compatible with other frameworks like Kraken, Sails, and Loopback.

Django: Django is a Model-View-Temple framework that uses Python for web development. Django has batteries-included functionality, which is a set of

features such as authentication and messaging. It follows the Convention Over Configuration model and the DRY model as well.

Laravel: Laravel is a Model-View-Controller framework that uses PHP, one of the most popular languages on the web.

Laravel comes with API support out of the box, and it also has a decent amount of packages that could extend its reach. Laracasts is a tutorial website with over a thousand videos on PHP, Laravel and frontend technologies of the Laravel ecosystem, which can be considered as a beginner's paradise.

Rails: Rails is a Model-View-Controller framework that uses Ruby, and it is a popular framework that is loved by many developers. There are many useful gems for Rails, which are library-like dependencies that extend the functionality of the application and help develop even faster and more efficiently.

Spring: Spring is a Model-View-Controller framework that uses Java, the most popular language. Spring has many sister projects that boost its performance and allow you to scale your business quickly. The fact that it uses Java, a strongly typed language, is a definite advantage for many web developers [web12].

2.2.5. Databases

All data exchanged over the web must be stored somewhere. For this most websites have their own databases associated with them. Below is a list of some databases - some relational and some non-relational - that are commonly used for web applications.

MySQL: is a relational database management system (RDBMS). It is distributed under a dual GPL and proprietary license. It is one of the most used database management software in the world3, both by the general public and by professionals [web13].

Microsoft SQL Server: is a database management system (DBMS) in SQL language incorporating, among other things, a RDBMS (relational DBMS)

developed and marketed by the Microsoft company. It works under Windows, Linux and Mac OS [web14].

PostgreSQL: is a relational and object database management system (SGBDRO). It is a free tool available under the terms of a BSD [web 5] type license.

Oracle: is a relational database management system (RDBMS) which since the introduction of object model support in version 8 can also be called an object-relational database management system (RDBMS). Provided by Oracle Corporation, it was developed by Larry Ellison, accompanied among others by Bob Miner and Ed Oates [web16].

MongoDB: is a document-oriented database management system that can be distributed over any number of computers and does not require a predefined data schema. It is written in C++. The server and tools are distributed under an SSPL license. It is part of the NoSQL movement [web17].

Redis: is an extensible, very high performance key-value database management system written in ANSI C and distributed under the BSD license. It is part of the NoSQL movement and aims to provide the highest performance possible [web18].

2.2.6. Data formats

Whenever there is a need to exchange data between two devices over the web, a proper procedure is followed. The data is properly packaged to be transmitted from source to destination. Special APIs are designed and integrated into websites for convenient data exchange. They organize the data in such a way that the receiver can easily decode and understand it. JSON and XML are two of the most widely used file formats on the web.

JSON: JavaScript Object Notation (JSON) is a standard format used to represent structured data similar to JavaScript objects. It is usually used to structure and transmit data on websites (for example, sending data from a server to a client in order to display it on a web page or vice versa). As this notation is extremely common [web19].

XML: for eXtensible Markup Language, is a general-purpose markup language recommended by the W3C like HTML. XML is a subset of the SGML language. This means that unlike other

markup languages, XML is not predefined, you must define your own tags. The main purpose of this language is the sharing of data between different systems, such as the Internet [web20]

2.2.7 Aplication programming interface(API):

2.2.7.1What is anAPI:

is a software-to-software interface, not a user interface.With API, applications talk to each other without any user knowledge or intervention.Note: Main goal of API is code/program reusable.An API resembles Software as a Service (SaaS), since software developers don’t have to start from scratch every time they write a program.Instead of building one core application that tries to do everything, the same application can contract out certain responsibilities to remote software that does it better.

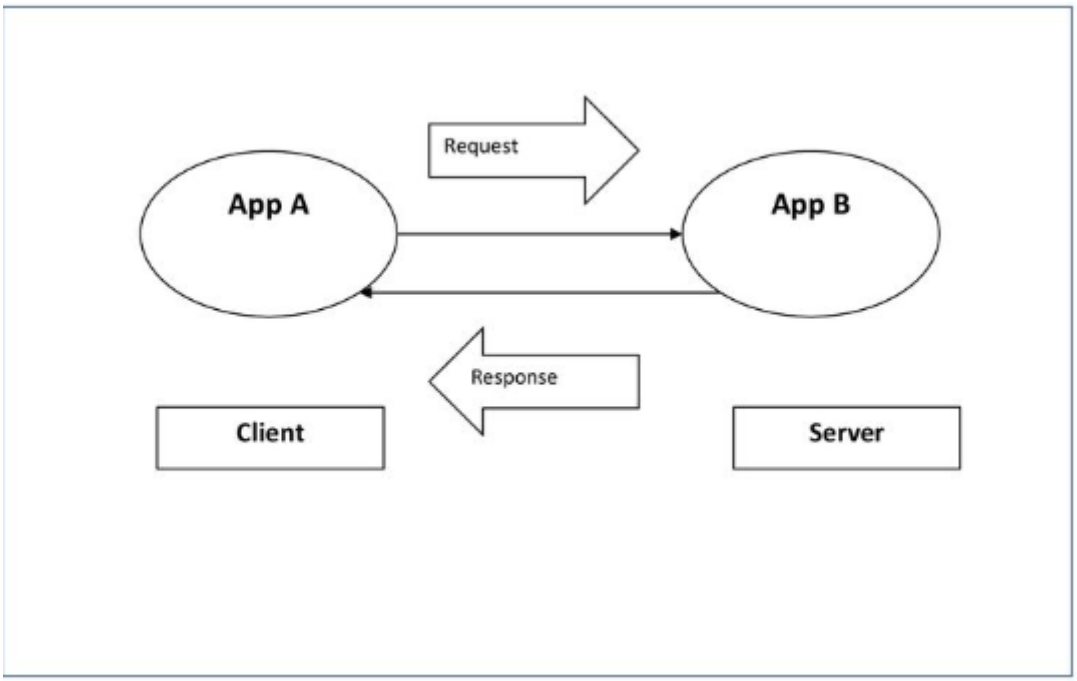


Figure 2.5: the structure of api

2.2.7.2 Types of API:

Local API: is the original form of API which is the OS APIs that provide services to application programs (Front-end/GUI) requesting services or data from the back-end such as voice service or data from DB.

Program API: is based on RPC (Remote Procedure Call) technology that making a remote program execution from another servers. SOA (Service Oriented Architecture) APIs are sample of Program API.

Web API: also known as Web Service, is application/device communicate to each others via World Wide Web (HTTP architecture).There are two kinds of Web

Service: SOAP (Simple Object Access Protocol) and RESTful (REpresentational State Transfer).

2.2.7.3What is SOAP Web Service?

SOAP Web Service: describes a standardized way of integrating Web-based applications using the XML, SOAP, WSDL and UDDI open standards over an Internet protocol backbone.

XML (EXtensible Markup Language) is used to tag the data.

SOAP (Simple Object Access Protocol) is used to transfer the data. (XML-based messaging protocol)WSDL (Web Services Description Language) is used for describing the services available. (Written in XML)

UDDI (Universal Description, Discovery, and Integration) is used for listing what services are available (XML-based registry).

Web services allow different web applications from different sources to communicate with each other without time-consuming custom coding and because all communication is in XML.Web services are not tied to any one operating system or programming language. For example, Java can talk with Perl, Windows applications can talk with UNIX applications.

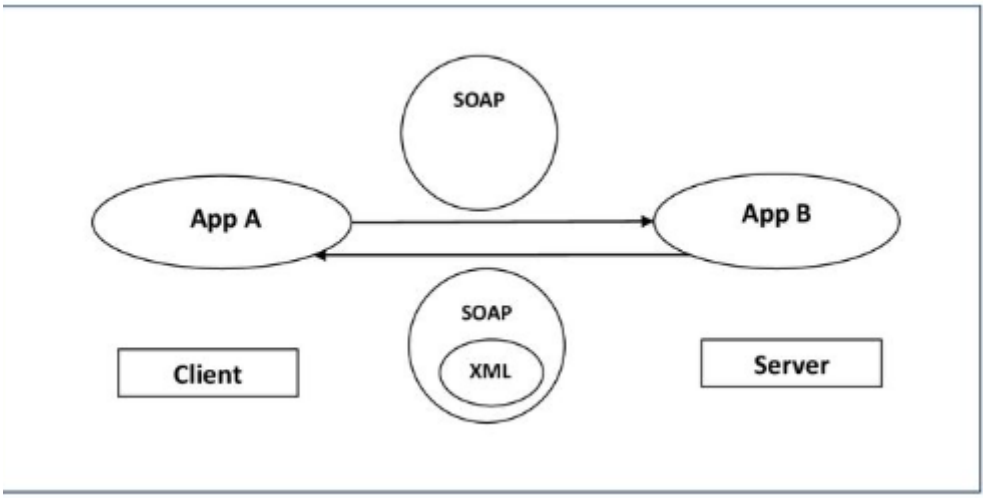


Figure 2.6: SOAP service web service

2.2.7.4 What is RESTful wweb service?

RESTful (REpresentational State Transfer) Web Service: also known as RESTful API, is based on REST technology which is an web application that uses HTTP requests to GET, PUT, POST and DELETE data.REST technology is generally preferred to the more robust Simple Object Access Protocol (SOAP) technology because REST leverages less bandwidth, making it more suitable for internet usage. With cloud use on the rise, APIs are emerging to expose web services. REST is

a logical choice for building APIs that allow users to connect and interact with cloud services.

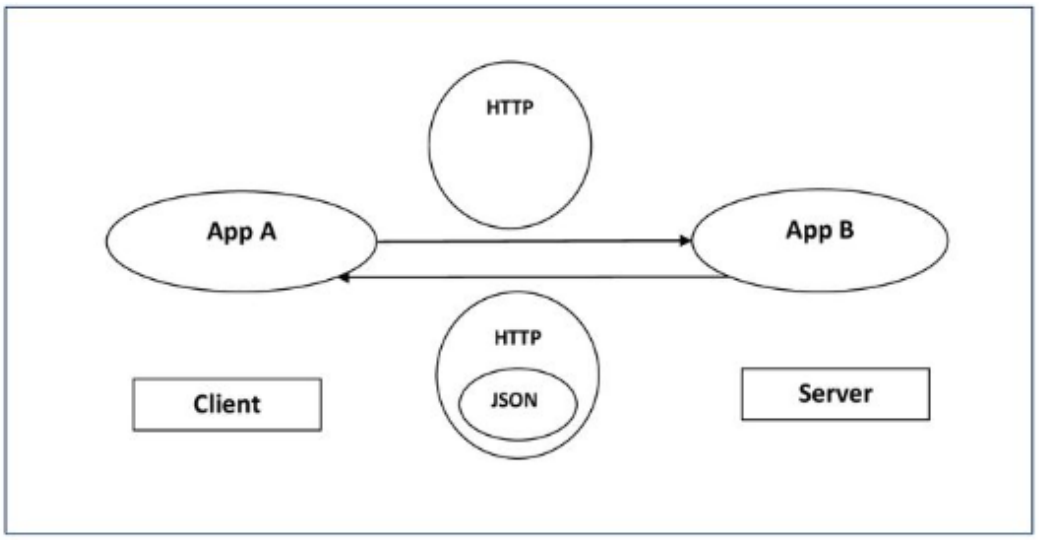


Figure 2.7: structure of Restful web Service

2.2.7.5 REST vs SOAP We Services

REST Web Services:

- **RESTful** web services are stateless. You can test this condition by restarting the server and checking if interactions survive.
- For most servers, RESTful web services provide a good caching infrastructure over an HTTP GET method. This can improve the performance if the information the service returns is not altered frequently and is not dynamic.
- Service producers and consumers must understand the context and content being passed along as there is no standard set of rules to describe the REST web services interface
- .REST is useful for restricted-profile devices, such as mobile, for which the overhead of additional parameters are less (e.g., headers).
- REST services are easy to integrate with existing websites and are exposed with XML so the HTML pages can consume the same with ease. There is little need to refactor the existing site architecture. As such, developers are more productive because they don't need to

rewrite everything from scratch; instead, they just need to add on the existing functionality.

- A REST-based implementation is simple compared to SOAP.
- REST does not enforce any message format such as XML or JSON. Whereas, SOAP is XML based messaging protocol

The Web Services Description Language (WSDL)

- describes a common set of rules to define the messages, bindings, operations and location of the service. WSDL is akin to a contract to define the interface that the service offers.
- SOAP requires less plumbing code than REST services design (e.g., transactions, security, coordination, addressing and trust). Most real-world applications are not simple and support complex operations, which require conversational state and contextual information to be maintained. With the SOAP approach, developers don't need to write plumbing code into the application layer.
- SOAP web services, such as JAX-WS, are useful for asynchronous processing and invocation.
- SOAP supports several protocols and technologies, including WSDL, XSDs and WS-Addressing

2.2.7.6 API Economy

API economy (application programming interface economy): is a general term that describes the way application programming interfaces (APIs) can positively affect an organization's profitability .An API is a customer interface for technology products that allows software components to communicate.There was a time when only software professionals knew about APIs. Today, business leaders are aware of the financial impact APIs can have and companies are generating revenue by exposing APIs as business building blocks for third party applications.

The emerging financial effects of APIs on businesses have gained steam thanks in part to mobile and social media technologies. Major companies that have gained revenue from APIs include SalesForce.com, Amazon, Facebook, Twitter, and Google.

2.2.7.7 Microsoft Azure – API Management

API Management consists of the following components:The API gateway is the endpoint that:Accepts API calls and routes them to your backends.Verifies API keys, JWT tokens, certificates, and other credentials.Enforces usage quotas and rate limits.Transforms your API on the fly without code modifications.Caches backend responses where set up.Logs call metadata for analytics purposes.

The publisher portal is the administrative interface where you set up your API program. Use it to:Define or import API schema.Package APIs into products.Set up policies like quotas or transformations on the APIs.Get insights from analytics.Manage users.

The developer portal serves as the main web presence for developers, where they can:Read API documentation.Try out an API via the interactive console.Create an account and subscribe to get API keys.Access analytics on their own usage.NOTE: For more information, see the Cloud-based API Management: Harnessing the Power of APIs PDF whitepaper.

https://slideplayer.com/slide/14475822/

2.2.8 Internet Protocols:

Web protocols are predefined rules that must be followed by all machines communicating on the Web. Several types of dialogs are used, below is a list of common web protocols and their uses

Protocol	Usage
TCP	It ensures that the sending of information is accompanied by an automatic .response from the recipient in order to ensure the arrival of the data sent
UDP	It just sends data over the network. Here, there is no verification regarding the .receipt of information
ICMP	Internet Control Message Protocol). It is used to retrieve information about) the state of your connection. By sending a “ping” or a “traceroute” to your machine, you can very easily find out if you are connected. If your extension responds to the request, this means that it is present and therefore accessible
SMTP	Electronic message transfer protocol. An SMTP server allows the sending of

	.emails
POP	Electronic message transfer protocol. The POP protocol allows users with .client software to retrieve and read their email from a server
FTP	File transfer protocol. The FTP protocol allows files to be saved between .computers on the “client-server” model
HTTP	.The HTTP protocol allows the transfer of Hypertext documents
HTTPS	With S for secured, i.e. “secured”) is the variant of HTTP secured by the use) of SSL or TLS protocols
TELNET	It is a terminal emulation protocol, which allows users to communicate remotely with computers hosts on the Internet. There are TELNET .applications for all operating systems

Table 2.2:web protocols and their use

3. Client/server architecture

The client/server architecture designates a mode of communication between several computers on a network which distinguishes one or more client workstations from the server: each client software can send requests to a server. A server can be specialized as an application, file, terminal, or even electronic messaging server [web21].

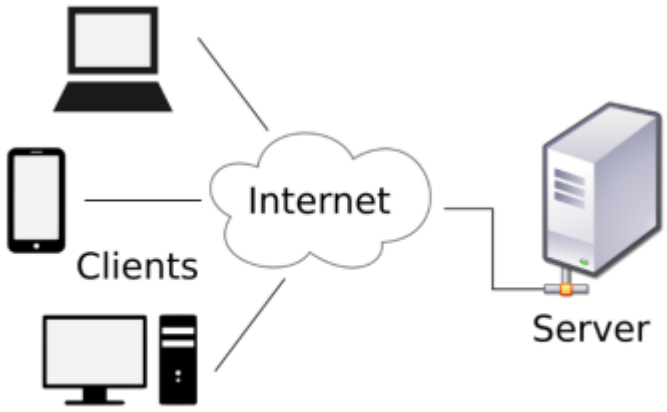


Figure 2.8: Client-Server Architecture

- **Web client:** A web client is software capable of sending HTTP requests to a web server and displaying the results. Web browsers are the most common web clients.
- **Web server:** A web server is software capable of responding to HTTP requests, i.e. returning data (for example an HTML page), in response to requests written in HTTP (for example a GET request) .

3.1. Types of client-server architecture

3.1.1. level 2 architecture

The two-tier architecture (also called 2-tier architecture, tier meaning row in English) characterizes client/server systems for which the client requests a resource and the server provides it directly, using its own resources. This means that the server does not call on another application to provide part of the service.

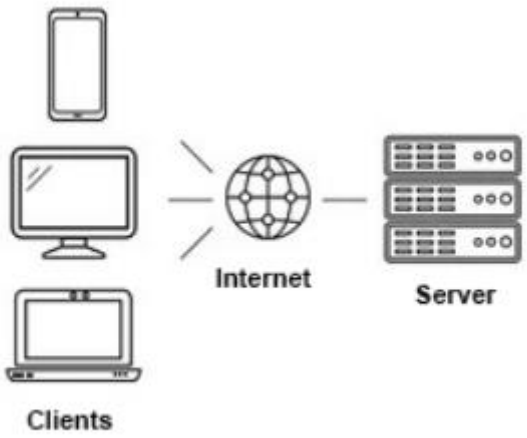


Figure 2.9: 2-tier architecture.

3.1.2. level 3 architecture

In this architecture (3-tier) there is an additional level:

1. A client (the resource requesting computer) equipped with a user interface (usually a web browser) responsible for the presentation.

2. An application server (called middleware) which provides the resource, but calls on another server.

3. A data server that provides the application server with the data required to respond to the client

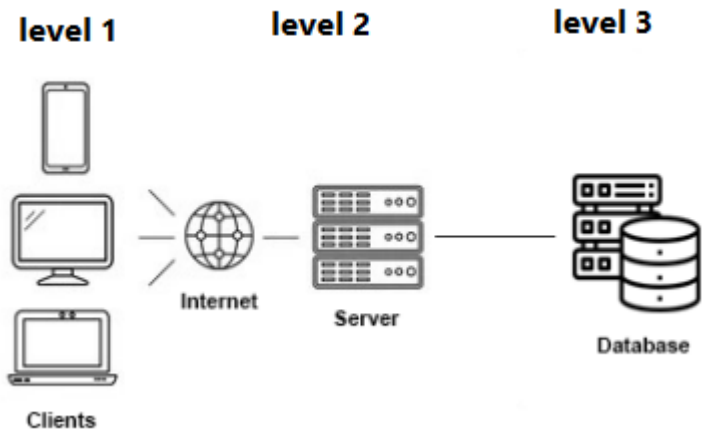


Figure 2.10:3-tier architecture

3.1.3. Multi-level architecture

In the 3-tier architecture, each server (tiers 2 and 3) performs a specialized task (service). A server can therefore use the services of one or more other servers in order to provide its own service. Therefore, the three-tier architecture is potentially an N-tier architecture...! [web22].

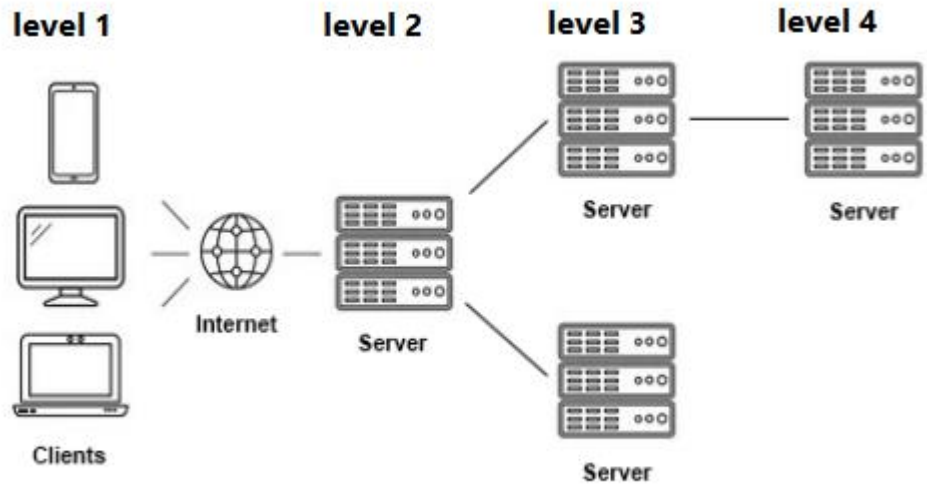


Figure 2.11: architecture n-tier

4. conclusion

The World Wide Web will expand at an astonishing rate to include more people, devices and resources available online. Therefore, web technologies are expected to evolve, and technology professionals need to do more to provide accessible web content to users around the world**.**

Chapter 03:

Analysis and Design

Chapter 03: Analysis and Design

1. Introduction:

In this chapter, we will present the needs of the users from the use case diagrams, the interactions between the actors and the system (within the use cases) will be explained in textual form (textual description) in graphical or average form of sequence diagram and activity diagrams.

Design represents the essential stage for the development of an IT project. The main objective of this phase is to analyze all the needs, then to imagine the contexts of use, to fix the choices of the data and the processing of manipulation. To arrive at a good design it is necessary to follow a design method, which serves to analyze the problem and design an effective and simple solution to implement. For this we presented the design of our project using the unified UML method.

2. Modeling methods and tools

2.1. UML modeling language

For the design of our system we adopted an object-oriented method. Indeed, the latter is an essential approach in the context of the development of applications.

To better present the architecture of our system, we will choose the most adopted modeling language UML:

2.1.1Definition

UML (Unified Modeling Language) was designed to be a common, semantically and syntactically rich visual modeling language. It is intended for the architecture, the design and the implementation of complex software systems by their structure as well as their behavior. UML has applications beyond software development, especially for process flows in industry

It is similar to plans used in other fields and consists of different types of diagrams. Overall, UML diagrams describe the boundary, structure and behavior of the system and the objects within it.

Chapter 03: Analysis and Design

UML is not a programming language, but there are tools that can be used to generate code in multiple languages from UML diagrams. UML has a direct relationship with object-oriented analysis and design [web23].

Figure 3.1: UML Logo

2.1.2. Usefulness of UML

Nowadays, business process modeling tools (ex BOUML) are expanding each year and the software suites are more and more numerous. The use and the functionalities of UML differ from one perimeter to another, according to the needs of the customers and application providers.

As part of an IT project for the IS, the use of UML modelling provides numerous benefits that act on:

- Modularity.
- Abstraction.
- Concealment.
- Consistent structuring of functionalities and data.

It also allows you to first clearly define customer needs, and thus avoid additional costs related to the delivery of software that does not satisfy the customer (According to a study of the Standish Group in 1994, for 53% of the software created, the delivery time rate. not respected is 120%, we had 90% of budgets not kept, and 60% of unavailability of certain functionalities).

In addition, UML modeling makes it possible to popularize aspects related to design and architecture, specific to the software, to the client. Also, it provides a quick understanding of the program to other external developers in case of recovery of the software and facilitates its maintenance.

However, there are also drawbacks to using UML. For the most part, mainly related to exceeding the delivery time of the software. To overcome this problem, the use of a spiral project cycle is recommended because it provides more agility and better risk management.

The use of UML also requires prior training to know its standard norms, this can be a disadvantage for the customer who does not have skills in this area. Note also that there may be a poor correspondence between the UML and the finalized project (Different development scheme at the design level: the analyst and the developer being two separate people) [web24].

2.1.3. UML formalism

UML is a modeling language. The current version, UML 2.5, offers 14 types of diagrams including 7 structural and 7 behavioral. For comparison, UML 1.3 had 25 types of diagrams.

UML not being a method, the use of the diagrams is left to the appreciation of each one. The class diagram is generally considered the central element of UML. Methods, such as the unified process proposed by the original creators of UML, use all the diagrams more systematically and focus the analysis on the use cases to develop by successive iterations a model of analysis, a design model, and other models. Other approaches are content to model only partially a system, for example certain critical parts which are difficult to deduce from the code.

UML is broken down into several parts:

- **Views:** these are the observables of the system. They describe the system from a given point of view, which can be organizational, dynamic, temporal architectural, geographical, logical, etc. By combining all these views, it is possible to define (or find) the complete system.

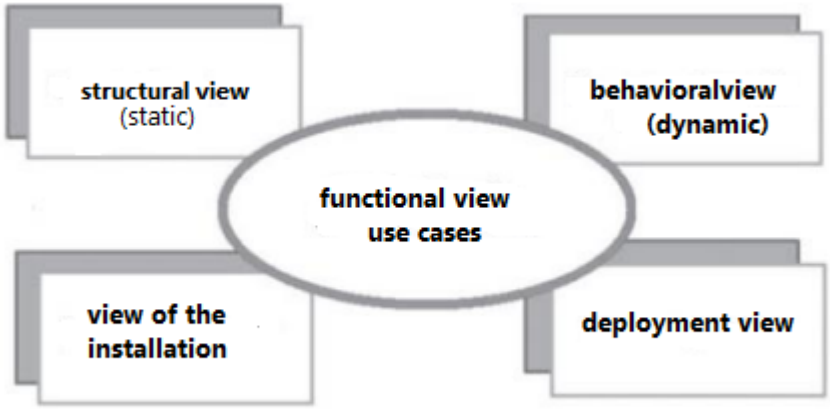


Figure 3.2: The 4 + 1 views of UML.

- **Diagrams:** these are sets of graphic elements. They describe the content of views, which are abstract notions. They can be part of several views.

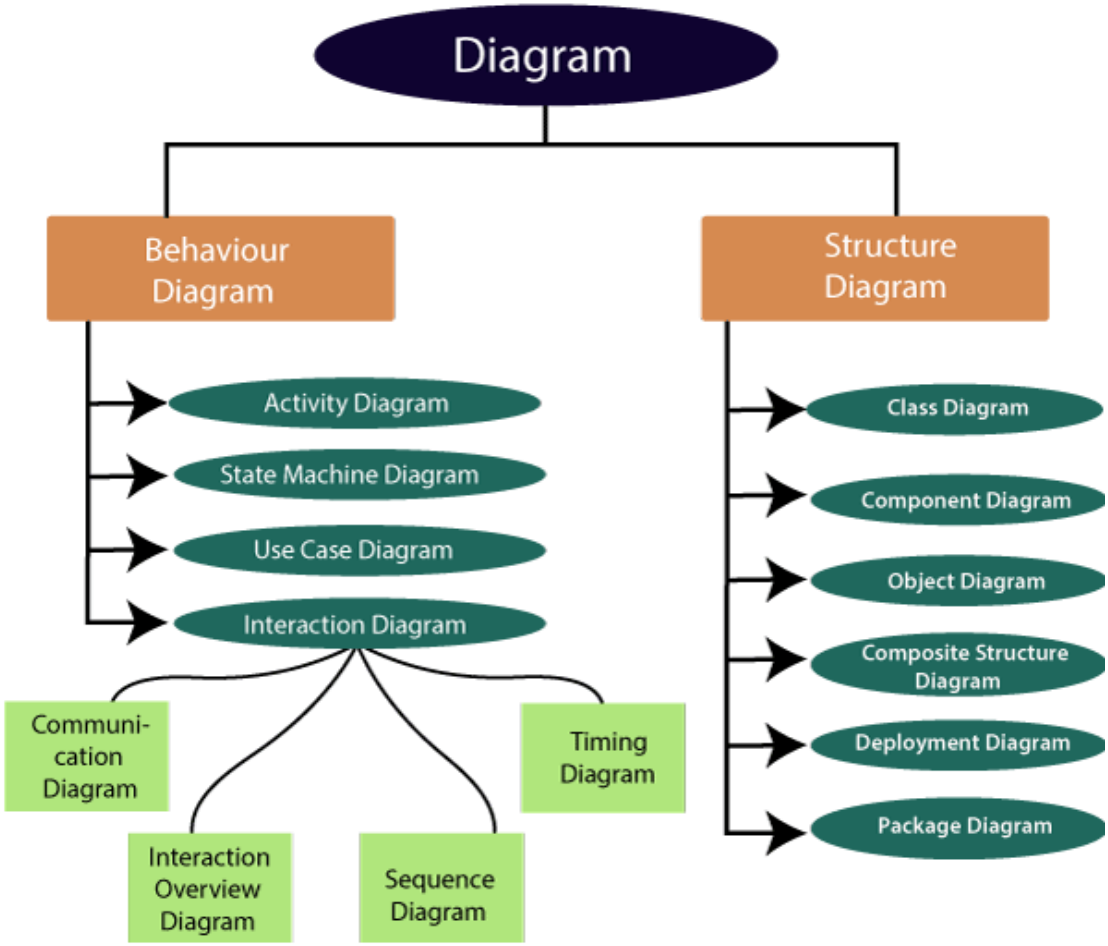


Figure 3.3: different type of UML diagram

- **Element models:** these are the graphic elements of the diagrams.

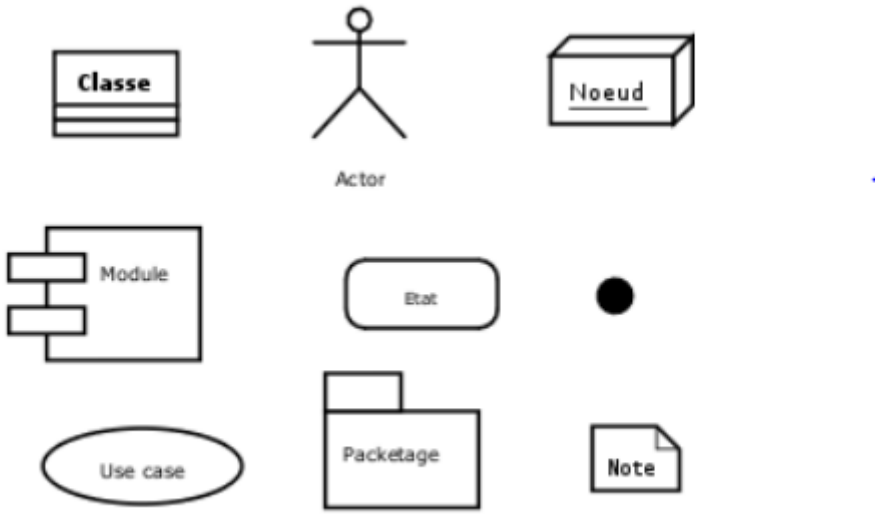


Figure 3.4: UML element model

2.1.4 Advantages and disadvantages of UML

2.1.4.1 Strengths of UML

- UML is a formal and standardized language allowing great precision and facilitating the use of tools.
- UML is a tool that facilitates communication, offers an analysis framework, allows the representation of abstract and complex elements and offers great expressiveness thanks to its versatility and flexibility.
- Gain of precision.
- Guarantee of stability
- Encourages the use of tools.
- It facilitates the understanding of complex abstract presentations.
- Its versatile character and flexibility make it a universal language.

2.1.4.2 Weak points of UML

- Putting UML into practice requires learning and goes through a period of adaptation
- UML is not at the origin of the object approach but specifies the underlying concepts.

- UML is not a method (and does not offer one): This will probably change later

2.2. Modeling tools

2.2.1 Visual Paradigm Online.

Visual Paradigm Online (VP Online) offers an easy-to-use online drawing tool with many useful features and applications, simplifying online diagramming with a powerful diagram editor and a central workspace to access your work and share it

Figure 3.5: Visual Paradigm Online Logo.

2.2.2 Draw.io Desktop

Draw.io Desktop is an Electron-based desktop diagramming and whiteboarding application that wraps the core draw.io editor.

You can use it as a flowchart maker, network diagram software, to create UML, as an ER diagram tool, to design a database diagram or to create BPMN.

Figure 3.6: Draw.io desktop logo

3. UML design:

This phase will make it possible to define in an unambiguous manner, using a modeling language, the future operation of the system, in order to facilitate its implementation. For the modeling of each application, we will distinguish:

• a use case diagram as a behavior diagram

• sequence diagrams as interaction diagrams

We will end with an activity diagram as a static diagram

3.1. usecase diagram:

The use case diagram represents the structure of the major functionalities necessary for users of the system. This is the first diagram of the UML model, the one where the relationship between the user and the objects that the system implements is ensured [AUD 09].

Use case diagrams illustrate and define the context and requirements of an entire system, or essential parts of a system.

You can model a complex system with a single use case diagram, or create many use case diagrams to model system components [web25].

This part of our modeling will consist of presenting the different actors in the system as well as the different interactions that they can carry out with it.

Use case diagram:

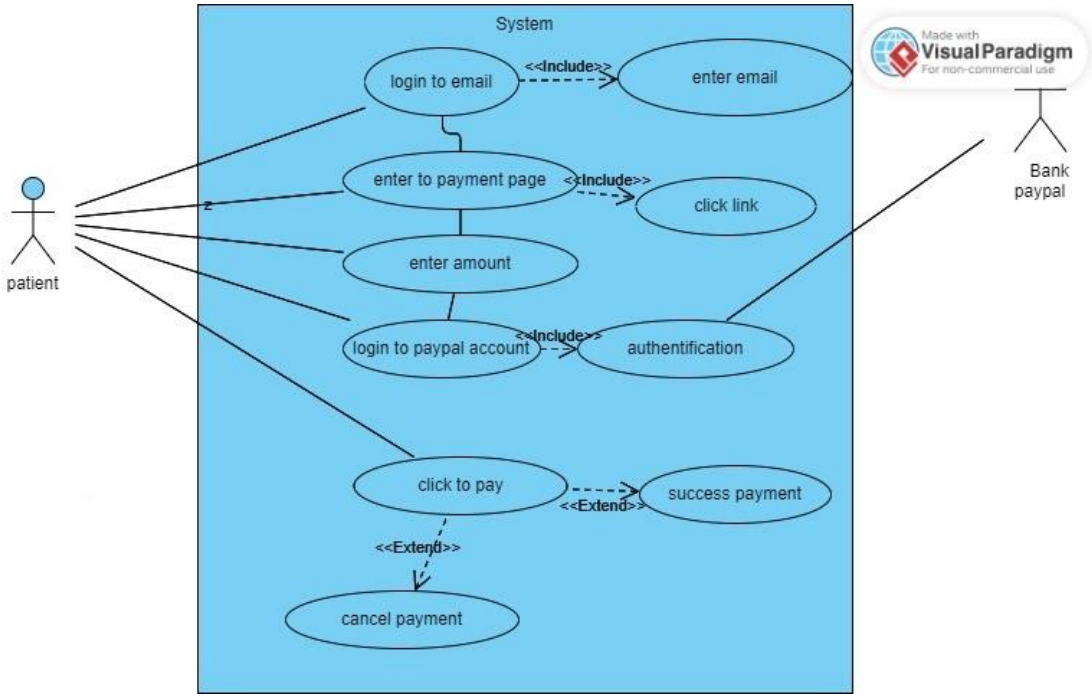


Figure3.7:use case diagram

3.2. Sequence diagrams

The sequence diagram represents the chronological succession of operations carried out by an actor. It indicates the objects that the actor will manipulate and the operations that move from one object to another.

Sequence diagram:

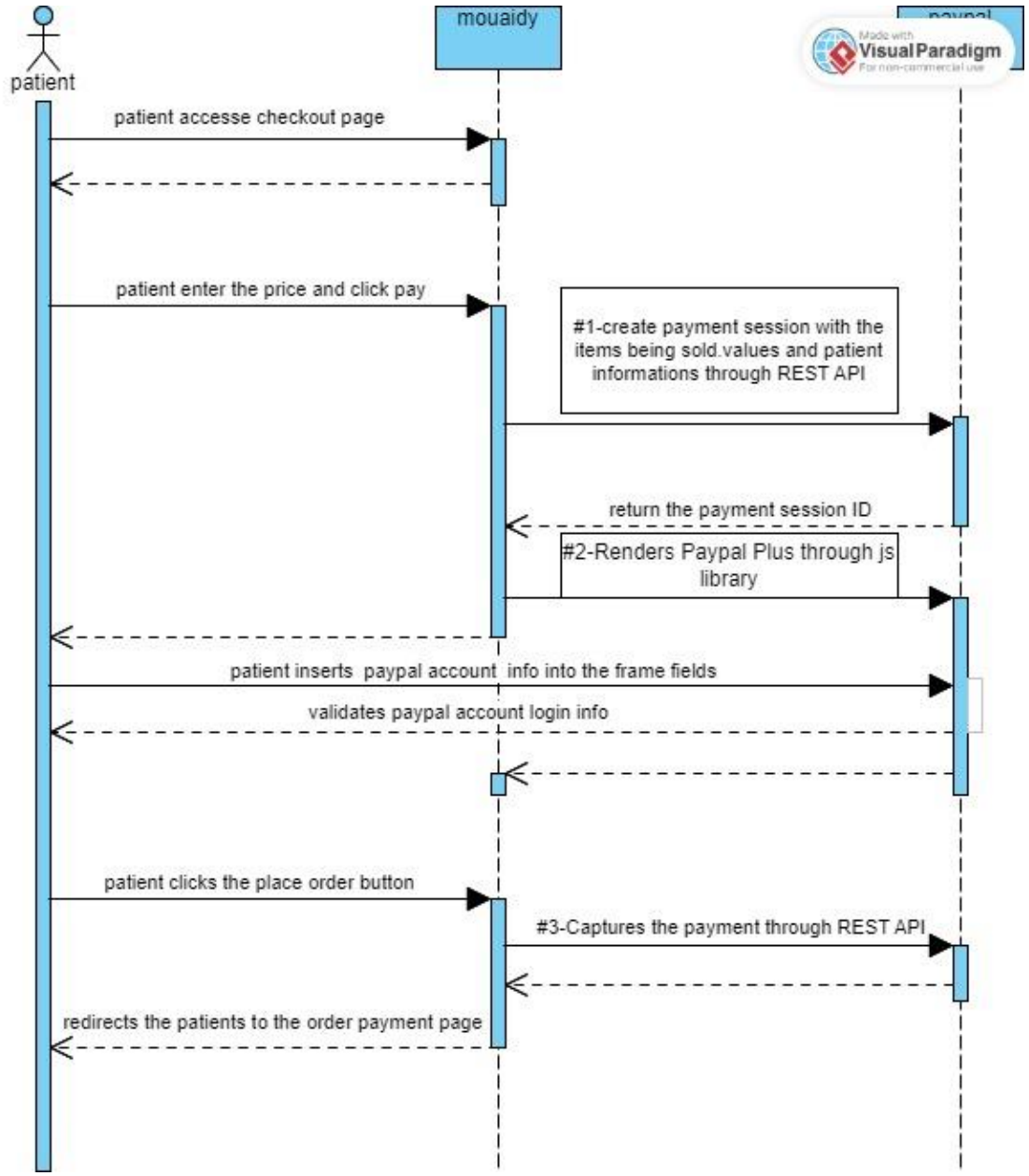


Figure3.8:sequence diagram

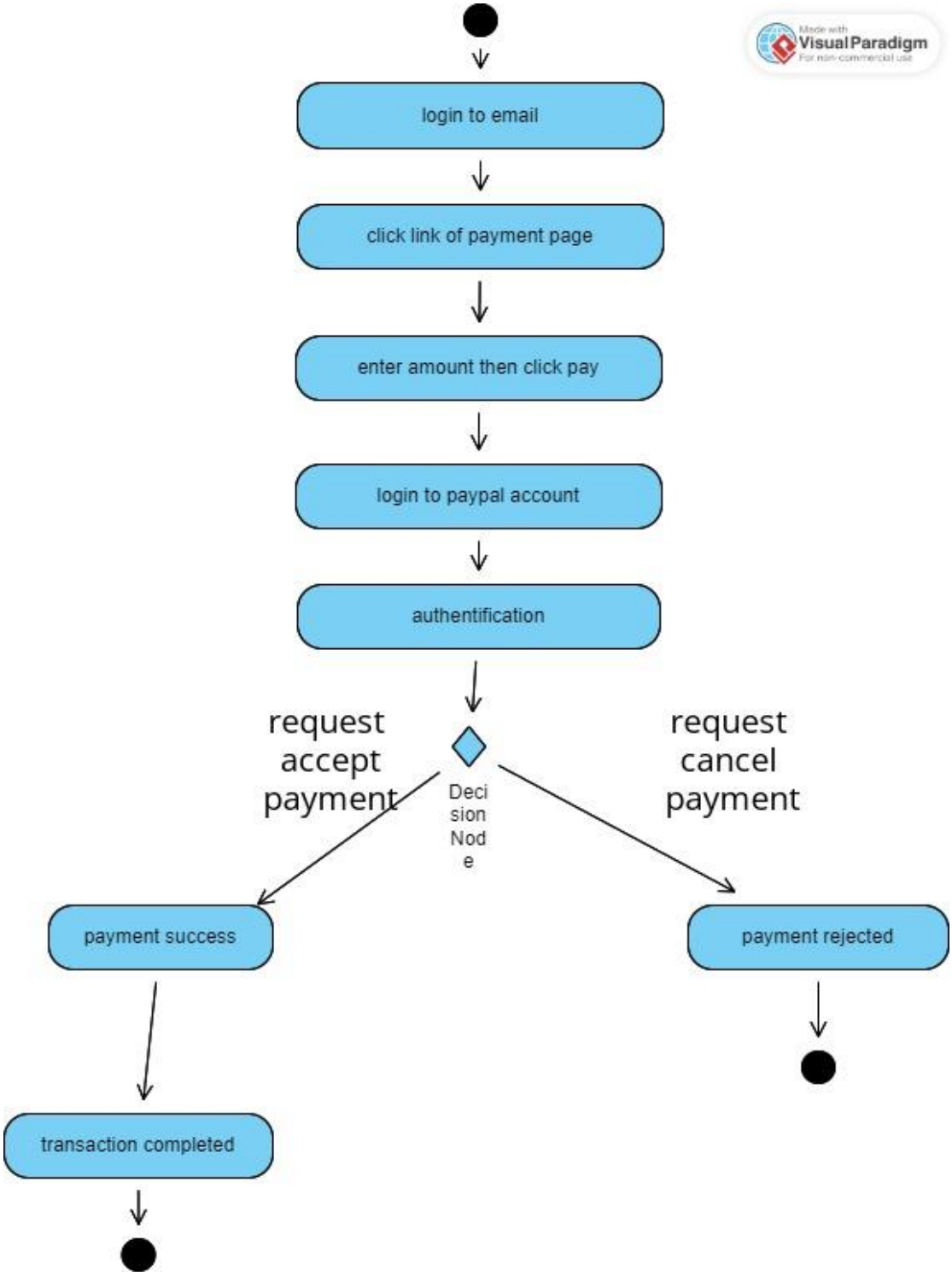
3.3. Activity diagrams

In UML, an activity diagram provides a view of a system's behavior by describing the sequence of actions of a process. Activity diagrams are similar to information processing flowcharts because they show the flows between actions in

an activity. Activity diagrams can, however, also show simultaneous parallel flows and replacement flows.

In activity diagrams, you use activity nodes and activity edges to model the flow of control and data between actions [web26].

Activity diagram:



Figures3.9:activity diagram

4. Conclusion

In this chapter we focused on the analytical and conceptual aspects of our system. The conceptual phase is a fundamental step for the realization of any project, on the one hand it allows to model the system before its realization, on the other hand it allows to help and guide the future user of the system .

This chapter allowed us to build the functional aspects corresponding to the needs expressed. This construction could not be easy and comprehensive without the use of diagrams and descriptive tables, and thanks to these, things are clearer and the implementation of our web application has become easier.

After having taken the trouble to design our system in a healthy way, it is now time to move on to its implementation. This is what we will see in the next chapter.

Chapter 04: implementation

Chapter 04: Implementation

1. Introduction

This chapter represents the last part of our dissertation, it will be devoted to the implementation of our online appointment booking system “Mouaidy”. We start with the Laravel web framework used to create this web application. Moving on to the presentation of the hardware and software resources as well as the languages and web technologies used in the development of our web application, then we will present overviews of the interfaces corresponding to our web application and we end this chapter with a conclusion.

2. Web development frameworks

Frameworks provide developers with the necessary tools and features and dictate the rules for building the architecture of websites, applications, APIs, services and other solutions. A framework may also include utility programs, code libraries, scripting languages, and other software intended to facilitate the development and integration of various components of a large software project.

2.1. Why use a framework?

The advantages of frameworks are numerous, these advantages include the following:

- A Framework is portable, due to its database abstraction and generic cache management.
- Development times with a Framework are really shorter.
- All essential tools are already written.
- The development of secure applications is easy thanks to authentication systems, SQL injection management as well as CSRF (Cross-Site Request Forgery) protection which is managed by most Frameworks.
- Frameworks are community tools and therefore have forums, mailing lists and IRC channels to support them.
- Furthermore, since the frameworks are widely deployed, the chance of finding fixes to the problems encountered is greater.

2.3. Argument for the choice of Laravel Framework

- For the realization of our project we chose Laravel for several reasons:
- Laravel is suitable for application development with complex needs, whether small or large.
- It is a complete PHP frameworks with features that will help us to customize complex applications.
- Laravel is very expressive. Its speed and security meet the expectations of a modern web application.
- Laravel is the way to go for developers who want to create enterprise applications that will evolve according to web trends.

3.The Laravel PHP Framework:

Developing web applications and sites has become increasingly simple in recent years. Even the most tech-newbies have become quite adept with products like WordPress and Wix.

For more advanced developers, there are plenty of tools to simplify the development process. One of these most useful tools is Laravel.

3.1. Definition:

Laravel is a free, open-source web framework. Created by Taylor Orwell and intended for developing web applications following the model-view-controller (MVC). Some of the features of Laravel are a modular packaging system with a dedicated dependency manager. To access different ways of relational databases, utilities that help with application deployment and maintenance and its orientation towards syntactic sugar [Fah 17]

Chapter 04: Implementation

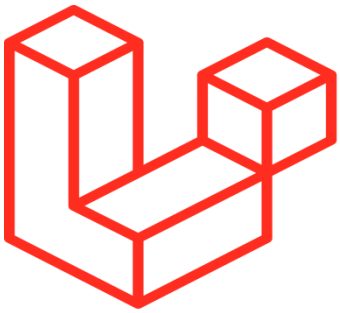


figure 4.1: logo of the the laravel framework

3.2. Structure of a Laravel application

A Laravel application contains several folders at the root. Knowing their content allows you to better understand subsequent development and improve your understanding of how the framework works.

- **The app directory:** contains the base code of your application. We'll explore this directory in more detail soon, however, almost every class in your application will be in this directory.
- **The bootstrap directory:** contains the app.php file which bootstraps the framework. This directory also contains a cache directory which contains files generated by the framework for performance optimization such as route and service cache files.
- The config directory, as its name suggests, contains all of your application's configuration files. It's a great idea to read through all of these files and familiarize yourself with all of the options available to you.
- **The database directory:** contains your database migrations, model factories and seeds. If desired, you can also use this directory to hold an SQLite database.
- **The public directory:** contains the index.php file, which is the entry point for all requests entering your application and configures automatic loading. This directory also hosts your resources such as images, JavaScript, and CSS.
- **The resources directory:** contains your views as well as your uncompiled raw resources such as LESS, SASS or JavaScript. This directory also contains all your language files.

- **The routes directory:** contains all route definitions for your application. By default, several route files are included with Laravel: web.php, api.php, console.php and channels.php. The web.php file contains the routes that the RouteServiceProvider places in the web middleware group, which provides session state, CSRF protection, and cookie encryption. If your app doesn't offer a stateless RESTful API, all of your routes will likely be defined in the web.php file.

  The api.php file contains the routes that the RouteServiceProvider places in the api middleware group, which provides rate limiting. These routes are intended to be stateless, so requests entering the application through these routes are intended to be authenticated via tokens and will not have access to session state.

  The console.php file is where you can define all your Closure-based console commands. Each closure is linked to a command instance allowing a simple approach to interact with the IO methods of each command. Although this file does not define HTTP routes, it defines console entry points (routes) into your application.

  The channels.php file is where you can register all event broadcast channels supported by your application.

- **The storage directory:** Contains your compiled Blade models, file-based sessions, file caches, and other files generated by the framework. This directory is separated into app, framework and logs directories.

  ➢ The app directory can be used to store all the files generated by your application.

  ➢ The framework directory is used to store files and caches generated by the framework.

  ➢ The logs directory contains your application's log files

- **The tests directory:** contains your automated tests. A sample PHPUnit test is provided out of the box. Each test class must be suffixed with the word Test.

You can run your tests using phpunit commands or php vendor/bin/phpunit [web27].

3.3. Why use Laravel?

Laravel provides a wide range of robust tools that make the web application development process easier and faster. The end application codebases are well structured and easy to maintain, here are some of the specific advantages of Laravel

Laravel is easy to learn: Laravel is relatively easy to learn, given the proper training. Laravel requires a general understanding of PHP and Object Oriented Programming (OOP) concepts for use
effective. It is also useful to know at least a few notions of HTML. And for any MVC architecture, it's also helpful to understand relational database management systems, such as MySQL or PostgreSQL.

Laravel simplifies the development process: From the ground up, Laravel was designed to simplify tasks common to many web development projects, like routing, authentication, migration, caching, etc. Laravel makes it easy to integrate pre-made modules into an application, using intuitive and expressive command line and Composer interfaces.

Laravel has tools for developers of all levels: Laravel describes itself as a progressive framework, which means it includes a variety of features that users of all levels will find useful. For example, beginners have access to starter kits for modules such as basic authentication functions.

Laravel scales easily: Laravel is highly scalable. With built-in support for fast, distributed caching systems, Laravel applications are capable of processing millions of requests per month. Laravel also offers a serverless deployment platform, Vapor, which is based on AWS and offers a high degree of scalability.

Laravel has a massive ecosystem and community: Laravel has a tremendous ecosystem supported by a large community of developers. Since Laravel is one of the most widely used PHP frameworks, the library of Laravel applications and packages

available is significant. Official Laravel packages and third-party packages are readily available.

Laravel is widely used: Many companies use Laravel to build highly functional websites [web28].

3.4. The advantages of Laravel

Laravel provides a range of benefits covering the following points:

Security and Performance: One of the most important benefits of choosing Laravel for your web application development is its capabilities to provide high-level security. If you choose Laravel, your web application will not presents no risk of unintentional and hidden SQL injections. Additionally, Laravel is capable of providing excellent performance of web applications. Sometimes certain features and functionalities affect the performance of the site. But Laravel offers various tools that help developers improve their performance.

Object-Oriented Libraries: Laravel is one of the best PHP frameworks because it has object-oriented libraries and other pre-installed, which are not found in any other PHP framework. One of the pre-installed libraries is the authentication library. These libraries are packed with great features that are easy to use and implement for every developer.

Documentation and Community: Laravel has a powerful developer community that constantly provides support to make it more flexible and scalable. Thus, if you wish to provide complex functionalities, numerous documentations are at your disposal.

Unit testing: With Laravel unit testing, each module of your web application is tested before the site goes live. These tests ensure a high-performance, bug-free, and ultimately hassle-free application for your end users. This is another exception from the Laravel framework.

Artisan: Laravel provides a built-in command line tool called Artisan. This tool helps to create a "skeleton" code and database architecture, as well as their migrations. Database management becomes easier. The Artisan tool can perform almost any repetitive and tedious programming task.

Chapter 04: Implementation

MVC architecture support: If you search on Google you will find that Laravel follows a Model-View-Controller architecture. And that's what makes Laravel a "great" framework to use for your web application development. It improves performance, provides clarity and allows for better documentation.

Security and Performance: One of the most important benefits of choosing Laravel for your web application development is its capabilities to provide high-level security. If you choose Laravel, your web application has no risk of unintended and hidden SQL injections

Generating URLS: Laravel also helps generate URLs, which is very useful for creating links in your models. When a user clicks or taps a link, he wants to see the desired content, such as an article, a product description, etc. This which is not possible without the help of URL routing. The Laravel framework provides a very simple route description strategy by just accepting a URI and a closing.

Integration with messaging services: Integration of the messaging service is another advantage provided by Laravel. It is used to send notifications to users to inform them of different events that occur. He gives also drivers for Mailgun, SMTP, Mandrill, SparkPost, mail function electronics of PHP and Amazon SES, which allow an application to start quickly.

Multilingual app creator: So this is the right option for businesses looking to expand across different countries with different languages. The Laravel framework therefore helps you create your application easily and quickly Web in different languages.

Laracasts tutorials: Laravel offers Laracasts features, a mix of free and paid video tutorials that show you how to use Laravel For the development. The videos are all created by Jeffery Way, an instructor expert. The video quality is excellent and the lessons are well thought out and useful [web29].

3.5. Laravel features

Laravel's feature set is far too vast to cover in this brief overview. Here are some of the most important features

Route processing: Laravel offers simple and intuitive route management, using simple names to identify routes rather than long route names path. The use of route identifiers also facilitates the maintenance of applications, because the road name can be

changed in one place rather than having to change it everywhere. All routes in the web interface of a Laravel application are saved in the routes/web.php file.

Security: Laravel includes a number of security features, including user authentication, user role authorization, email verification, encryption services, password hashing, and reset functions passwords.

Migration: Laravel provides version control for application databases using migrations. Migrations help track how a database has changed over time, making it easier to destroy or recreate the database if necessary.

Templating: Blade is Laravel's PHP templating engine. PHP templating engines help separate business logic from HTML templates, resulting in a more maintainable code base. Many Laravel features rely on Blade templates.
Blade offers more features than other templating engines because it allows you to use simple PHP code, which others do not.

Sessions: Laravel uses sessions to store information about the user across multiple requests. Cookies are an example of a session driver built into Laravel. Data validation. Laravel makes it easy to validate incoming user data. Laravel includes a number of data validation rules, with customizable error messages.

Cache processing: Laravel supports data caching to minimize application task processing times. Laravel's caching API supports a variety of third-party caching backends such as Memcached and Redis.

Error handling: Error handling is automatically configured when starting a new Laravel project. Laravel applications can be run in debug mode, generating detailed error messages for any errors that occur.

Testing: Laravel offers great testing functionality right out of the box. Laravel supports unit testing, which tests small, isolated sections of application code, as well as feature testing, which tests larger sections of code and higher-level functionality.

File storage and management: Laravel uses the Flysystem PHP package to provide drivers for working with a variety of file systems, from local file systems to cloud storage like Amazon S3. Laravel also allows file transfer with the SSH File Transfer Protocol (SFTP).

Email: Laravel includes an email API based on the SwiftMailer library, which allows you to send emails via the service of your choice. Laravel supports attachments and email queuing.

Notifications: Laravel supports sending notifications through a number of channels, whether known channels like SMS or Slack, or channels developed by the Laravel community [web28].

3.6 Laravel MVC architecture Laravel,

like many other PHP frameworks, has a so-called MVC (Model – View – Controller) architecture. Here is a simple illustration to make you quickly understand the logic of this architecture.

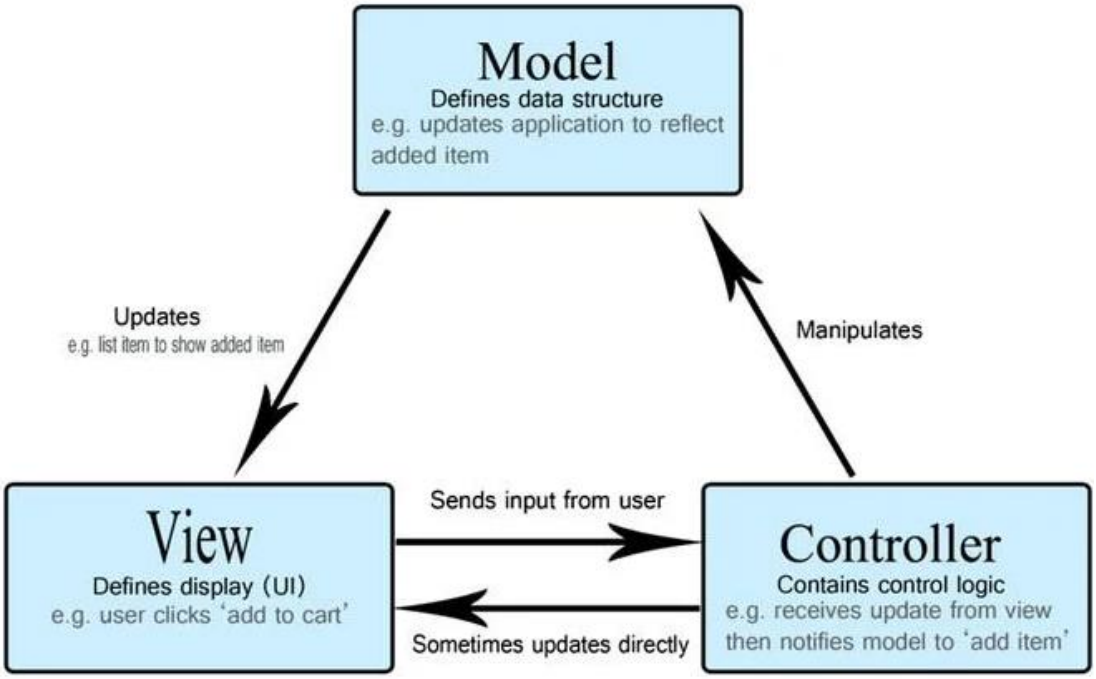


Figure 4.2: The MVC architecture.

Each user action goes through the Controller which sends requests to manipulate an object to the model. The model (model) performs the
changes the object and sends it back to the controller which in turn passes it to the View. So these three poles have different responsibilities:

- **The Model:** consists of the application data, as well as all the associated logic. A subscriber table, for example, can be a template.
- **The View:** actually provides the user interface element of the application. It will render the model data in a form suitable for the user interface.
- **The Controller:** receives user input and makes calls to model objects and the view to perform the appropriate actions

The controller therefore has a central place in this architecture. It is the bridge between user interactions and data processing. Everything goes through him in order to be able to control everything... hence his name.

4. Development environment

To set up our system, we used a development environment which ensured the smooth running of the implementation phase. This environment includes hardware and software tools.

3.1. Hardware environment

For the development of our application we used a “DELL Inspiron 3521” laptop PC whose configuration is as follows:

- ✓ Processor: Intel(R) Core(TM) i3-4005U CPU @ 1.70GHz 1.70GHz.
- ✓ Installed memory (RAM): 8.00GB.
- ✓ System Type: 64-bit operating system, x64 processor.
- ✓ Windows Edition: Windows 8.1 Professional

4.2. Software environment

During this part we will list the different software used for the application implementation. The choice of these software is based on their simplicity of use and their ways of providing results corresponding to our needs.

4.2.1 Visual Studio Code

Visual Studio Code is a standalone source code editor that runs on Windows, macOS and Linux. The best choice for JavaScript and web developers,with tons of extensions to support just about any language programming. It comes with built-in support for JavaScript, TypeScript, and Node.js and has a rich ecosystem of extensions for other languages (such as C++, C#, Java, Python, PHP, Go) and runtimes (such as .NET an Unity).

Figure 4.3:visual studio logo

4.2.2 XAMPP

XAMPP is a set of software for setting up a serverlocal Web, an FTP server and an email server. It's about adistribution of free software (X (cross) Apache MariaDB Perl PHP) offering a good flexibility of use, renowned for its quick and easy installation. Thus, he is at the reach of a large number of people since it does not require knowledge

features and also works on the most popular operating systems

Figure 4.4: xampp logo

4.2.3 Composer:

Composer is a free dependency manager software written in PHP. It allows its users to declare and install the libraries that the main project needs

Figure 4.5: composer logo

4.3. Developement languages and tools

In this part, we present the different languages, libraries and back-end and front-end development frameworks used in the realization of our Web application.

4.3.1. HTML (HyperText Markup Language)

HyperText Markup Language (HTML) is the code used to structure a page

website and its content. For example, the content of your page could be structured in a set of paragraphs, a bulleted list or with images and data tables.
As the title suggests, this article provides you with the basics of understanding HTML and its functions. [web30].

Figure 4.6: html logo

4.3.2. CSS (Cascading Style Sheets)

Cascading style sheets, generally called CSS Cascading Style Sheets, form a computer language that describes the presentation of HTML and XML documents. The standards defining CSS are published by the World Wide Web Consortium (W3C). Introduced in the mid-1990s, CSS became commonly used in web design and well supported by web browsers in the 2000s [web31].

Figure 4.7:css logo

4.3.3. JavaScript

JavaScript is a scripting programming language primarily used in interactive web pages and as such is an essential part of web applications. Along with HTML and CSS, JavaScript is at the heart of the languages used by web3 developers. A large majority of websites use it4, and the majority of web browsers have a JavaScript5 engine to interpret it [web32].

Figure 4.8: javascript logo

4.3.4. jQuery

jQuery is a fast, small, and feature-rich JavaScript library. He makes things like traversing and manipulating HTML documents, managing much simpler events, animation and Ajax with an easy to use API which works on a multitude of browsers. With a combination of versatility and extensibility [web33].

Figure 4.9: jquery logo

4.3.5. Bootstrap

Bootstrap is a collection of tools useful for creating design (graphics, animation and interactions with the page in the browser, etc.) of sites and applications web. It is a set that contains HTML and CSS codes, forms, buttons, navigation tools and other interactive elements, as well as JavaScript extensions in option. It is one of the most popular projects on the management platform GitHub development [web34].

Figure 4.10: boostrap logo

4.3.6. PHP (Hypertext Preprocessor)

PHP: Hypertext Preprocessor, better known by its acronym PHP (self-referential acronym), is a free programming language, mainly used to produce dynamic web pages via an HTTP server, but can also run like any locally interpreted language. PHP is an imperative language object oriented.

PHP has made it possible to create a large number of famous websites, such as Facebookand Wikipedia30. It is considered one of the bases for the creation of so-called websites. dynamic but also web applications [web35].

Figure 4.10: php logo

4.3.7. SQL (Structured Query Language)

SQL (Structured Query Language) is a language for communicating with a database. This computer language is particularly widely used by web developers to communicate with data on a website. SQL.sh lists

SQL lessons and explanations of the main commands for reading, inserting, modify and delete data in a database [web36].

figure4.11: sql logo

Chapter 04: Implementation

4.4. Web libraries used

In this part of the chapter, we will present the web libraries used in our web application implementation, these libraries help us to simplify certain tasks such as sending emails and SMS.

4.4.1. PHPMailer

PHPMailer is the most popular open source PHP library for sending emails. It was first published in 2001, and since then it has become the preferred way for PHP developers to send email programmatically, to shares a few other favorites like Swiftmailer. It provides a list of features progress:

- ✓ SMTP Authentication
- ✓ Secure encryption/MIME
- ✓ Support for TLS and SSL protocols
- ✓ HTML content with plain text
- ✓ Multiple fs, string and binary attachments
- ✓ Support for embedded images

4.4.2. ChartJS

Chart.js is a community-maintained, open-source library that lets you helps to easily visualize data using JavaScript. It is similar to Chartist and Google Charts. It supports 8 different chart types (including bars, lines and sectors), and they are all responsive. In other words, you configure your chart once, and Chart.js will do the heavy lifting for you and make sure it's always.

4.4.3. Laravel Livewire

Laravel Livewire is a library that simplifies the creation of interfaces modern, responsive and dynamic using Laravel Blade as the programming language modelization. This is a great library to choose if you want to create a dynamic and responsive app, but you don't feel comfortable jumping in a full JavaScript framework like Vue.js.

3.4.4. Twilio SMS

Twilio's programmable SMS API helps you add robust messaging to your applications. Using this REST API you can send and receive SMS messages, track delivery of sent messages, schedule SMS messages to send later and retrieve and modify the history of messages.

5. Database Implementation

In this part of our chapter, we present the tools and languages

which allowed us to create and manipulate the database of our web application

5.1. Choice of MySQL SGDB

MySQL is a relational database server (RDBMS). Open Source allows data to be stored in separate tables rather than putting everything together in only one table. This improves the speed and flexibility of the whole thing. The tables are linked by defined relationships, which make possible the combination of data between multiple tables during a query. The SQL in "MySQL" stands for "Structured Query Language": the standard language for database processing [web37].

MySQL is one of the most widely used database management software in the world. world, both by the general public and by professionals, in competition with Oracle, PostgreSQL and Microsoft SQL Server.

Figure 4.12: mysql logo

5.2. Database management tool “phpMyAdmin”

For users, phpMyAdmin is a web tool often available to create, populate and use MySQL databases.

phpMyAdmin is a free software tool written in PHP, intended to manage web-based MySQL administration. phpMyAdmin supports a wide range operations on MySQL and MariaDB. Frequently used operations (management of databases, tables, columns, relationships, indexes, users, permissions, etc.) can be done through the UI, while you have always the ability to directly execute any SQL statement.

This is one of the most famous interfaces for managing a database MySQL on a PHP server. This convenient interface makes it very easy to run and without great knowledge of databases, queries such as the creation of data table, insertions, updates, deletions and structure modifications of the database, as well as granting and revocation of rights and import/export. This system allows you to conveniently save a database as a file sql and transfer data there, even without knowing SQL [web38].

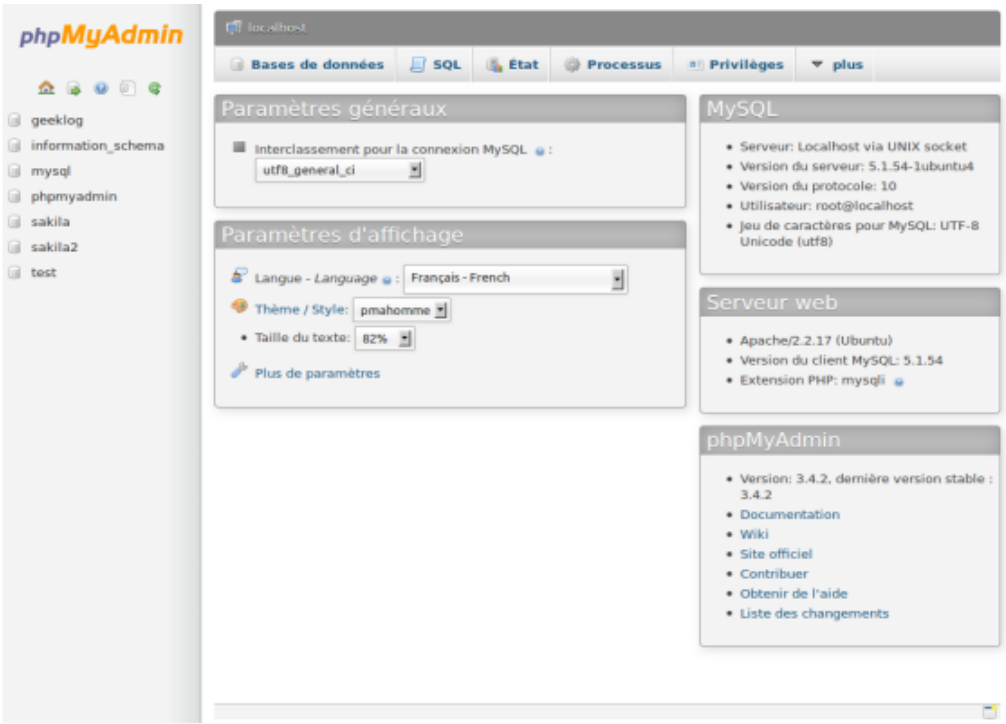


Figure 4.13: interface of phpmyadmin

6. The interfaces of our web application

In this part of the chapter we will present some screenshots of the different parts of our web application after improvement by showing the different changes to our site.

Our web application contains a lot of parts and interfaces, we don't We can't show them all because there aren't enough spaces, so we choose some of their essential interfaces

The “Home” page: this is the home page or index page of our application web, it is made up of several sections, among these sections you can find:

- ✓ The Page Header.
- ✓ Page of payment
- ✓ 3 steps paypal payment
- ✓ The Footer of the page

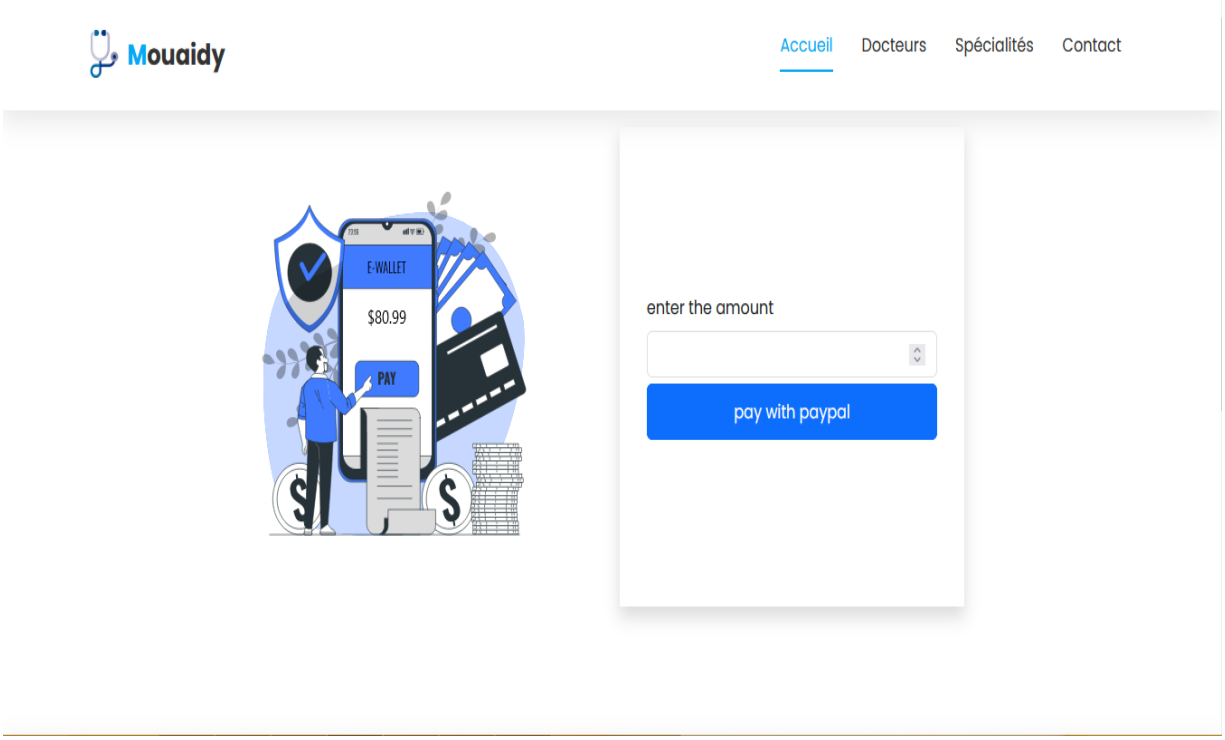


Figure 4.14 .:page of payment

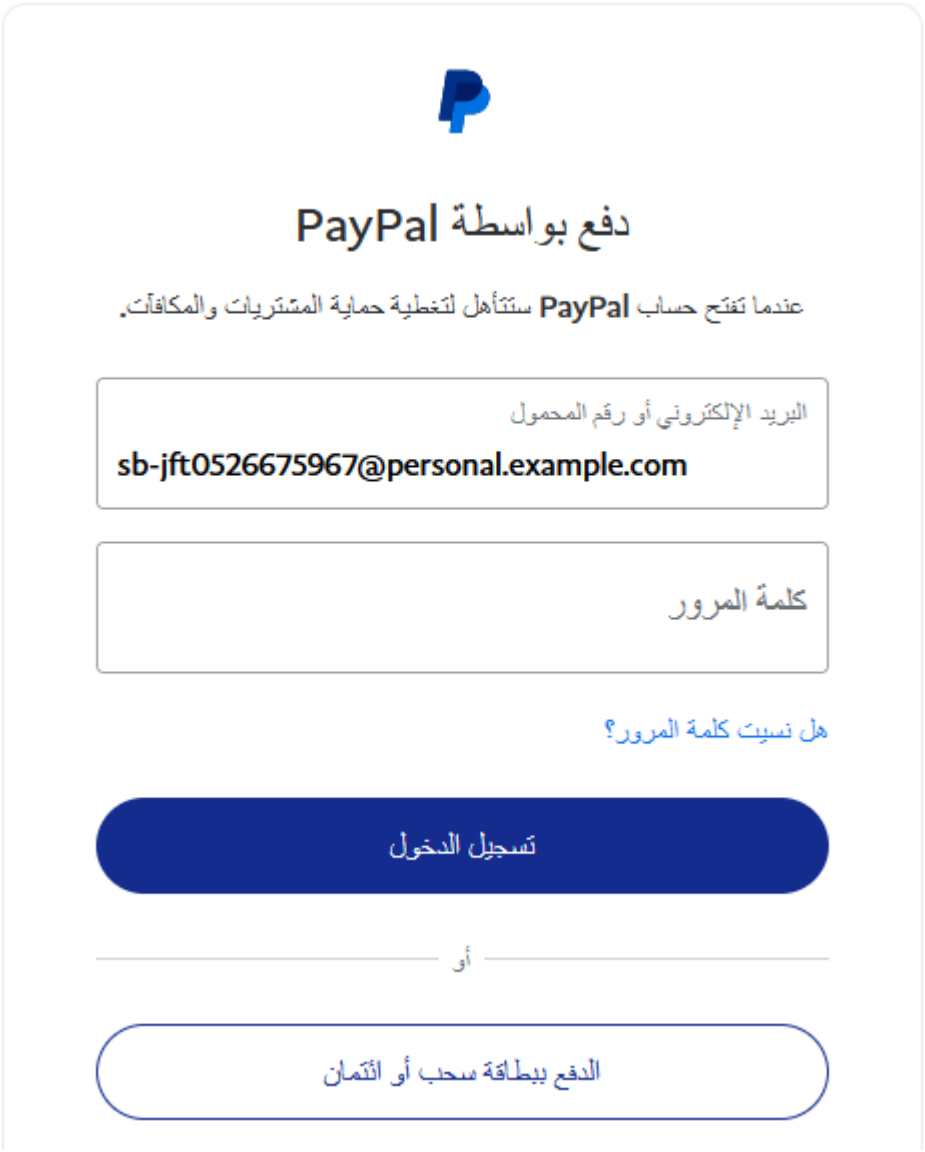


Figure 4.15:steps entered information of login

Figure 4.16:last steps to payment

7.Conclusion

In this chapter, we described the process of creating our web application by outlining the hardware and software resources used to implement the project, and the different tools, frameworks, and development languages we used to implement our web application. Through Next, we presented some of the interfaces of our web application. Optimization. In other words, we maintain the final version of the site that offers all the useful features needed to run a medical appointment booking site with a proper online payment system.

General Conclusion

In conclusion, electronic payment has transformed the way we handle financial transactions, offering a myriad of benefits and reshaping the landscape of finance and commerce. This digital evolution has allowed individuals, businesses, and organizations to transition away from traditional payment methods towards faster, more convenient, and secure electronic alternatives.

In summary, electronic payment has become an integral part of our daily lives, facilitating financial transactions with unprecedented speed, convenience, and security. As technology continues to advance, the world of electronic payments will undoubtedly see further innovation, shaping the future of finance and commerce in ways that are more efficient, inclusive, and adaptable to the evolving needs of society

electronic payment systems have become an integral part of modern finance, offering an array of benefits while also posing challenges that require ongoing attention. As society continues to embrace digital transformation, it is imperative to strike a balance between innovation and regulation to harness the full potential of electronic payments while safeguarding the interests of users and the stability of financial systems. The future of electronic payments holds exciting possibilities, and researchers, policymakers, and industry stakeholders must collaborate to navigate this dynamic landscape effectively.

In our project, we have linked our website to the PayPal electronic payment system, and this will make it easier for our website users to conduct financial transactions more easily.

.

Bibliography

[Web1]: https://wikimemoires.net/2011/04/e-paiement-electronique-e-commerce

[Web1.2] :https://www.ayoconnect.com/blog/brief-history-of-online-payments-where-electronic-payments-began

[Web1.3] : https://tfig.unece.org/FR/contents/e-payments.html

[Web1.4]: https://www.dbs.com/digibank/in/articles/pay/what-is-electronic-payment

[Web2]: https://home.bluesnap.com/snap-center/blog/types-of-electronic-payment/

[Web2.2]: https://etactics.com/blog/how-do-electronic-payments-work.

[Web3] :https://en.wikipedia.org/wiki/paypal.

[Web3.2]: https://en.wikipedia.org/wiki/skrill.

[Web3.3]: https://en.wikipedia.org/wiki/neteller.

[Web3.4]: https://www.asjp.cerist.dz/en/downArticle/417/16/1/184453.

[Web4] :https://developer.mozilla.org/fr/docs/Glossary/World_Wide_Web

[Web5] : https://c-marketing.eu/du-web-1-0-au-web-4-0/

[Web6]:https://waytolearnx.com/2018/09/difference-entre-les-pages-web-statiques-et-dynamiques.html

[Web7] : https://www.ideematic.com/dictionnaire-digital/application-web/

[Web8] ; https://www.goodcore.co.uk/blog/web-technologies/

[Web9] : https://developer.mozilla.org/fr/docs/Web/HTML

[Web10] : https://developer.mozilla.org/fr/docs/Web/CSS

[Web11] : https://blog.back4app.com/fr/les-dix-meilleurs-langages-backend/

[Web12] : https://mobiskill.fr/blog/conseils-emploi-tech/quel-framework-utiliser-pour-du-developpement-web/

[Web13] : https://fr.wikipedia.org/wiki/MySQL

[Web14] : https://fr.wikipedia.org/wiki/Microsoft_SQL_Server

[Web15] : https://fr.wikipedia.org/wiki/PostgreSQL

[Web16] : https://fr.wikipedia.org/wiki/Oracle_Database

[Web17] : https://fr.wikipedia.org/wiki/MongoDB

[Web18] : https://fr.wikipedia.org/wiki/Redis

[Web19] : https://developer.mozilla.org/fr/docs/Learn/JavaScript/Objects/JSON

[Web20] : https://developer.mozilla.org/fr/docs/Web/XML/XML_introduction

[**Web20.1**]:https://slideplayer.com/slide/14475822/

[Web21] : https://www.techno-science.net/definition/3743.html

[Web22] : https://web.maths.unsw.edu.au/~lafaye/CCM/cs/cs3tier.htm

[Web23] : https://www.lucidchart.com/pages/fr/langage-uml

[Web24] : https://manurenaux.wp.imt.fr/2013/09/27/interet-de-luml-dans-un-projet-informatique/

[Web25] : https://www.ibm.com/docs/fr/rsar/9.5?topic=diagrams-use-case

[Web26] : https://www.ibm.com/docs/fr/rsar/9.5?topic=diagrams-activity

[Web27] : https://www.oulub.com/fr-FR/Laravel/structure

[Web 28] : https://kinsta.com/fr/base-de-connaissances/qu-est-ce-que-laravel/

[Web 29] : https://easypartner.fr/blog/pourquoi-choisir-le-framework-laravel/

[Web30] : https://developer.mozilla.org/fr/docs/Learn/Getting_started_with_the_web/HTML_basics

[Web31] : https://fr.wikipedia.org/wiki/Feuilles_de_style_en_cascade

[Web32] : https://fr.wikipedia.org/wiki/JavaScript

[Web33] : https://jquery.com/

[Web34] : https://fr.wikipedia.org/wiki/Bootstrap_(framework)

[Web35] : https://fr.wikipedia.org/wiki/PHP

[Web36] : https://sql.sh/

[Web37] : https://www.futura-sciences.com/tech/definitions/internet-mysql-4640/

[Web38] : https://fr.wikipedia.org/wiki/PhpMyAdmin

Table of content

General introduction	01
Chapter 01 : electronic payment	
1. introduction	02
2. what is electronic payment	03
2.1. definition 01	03
2.2. definition 02	03
2.3. definition 03	03
2.4. definition 04	03
3. advantages of electronic payment	04
3.1.time saving	04
3.2.efficient	04
3.3.cashless economy	04
3.4.security	04
3.5.cerrtainty	04
4.types of electronic payment	05
4.1.card payment	05
4.2.bank transfer payments	05
4.3.ewallet payment	06
5.the electronic payment how it's work	07
6.some solution of electronic payment	07
6.1.paypal banking	07
6.2 skrill banking	08
6.3 neteller banking	08
7.electronic means payment in Algeria	09
7.1.CIB bank cards	10
7.2.EDAHABIA card	11
7.3.the visa and mastercard credit card in Algeria	11
7.4.the launch of e-payment in Algeria	12
7.5.the launch of mobile payment in algeria	14
8.conclusion	15

Chapter 02 : web technologies

1. Introduction	16
2. web tecknologies	16
2.1. the web	16
2.1.1. the evolution of the web	17
2.1.2. static and dynamic websites	18
2.1.2.1. static websites	18
2.1.2.1. dynamic websites	18
2.1.3. web application	19
2.2. the different types of web technologies	20
2.2.1. web browsers	20
2.2.2. fontend vs backend developement	20
2.2.3. web developement languages	21
2.2.4. web developement frameworks	22
2.2.4.1 frontend frameworks	23
2.2.4.2 backend frameworks	23
2.2.5. database	24
2.2.6. data formats	25
2.2.7. application programming interface(api)	26
2.2.7.1.what is an api	26
2.2.7.2.types of api	26
2.2.7.3.what is soap web service	27
2.2.7.4.what is RESTful web service	28
2.2.7.5.REST vs SOAP web service	28
2.2.7.6.api economy	29
2.2.7.7.Microsoft Azure—API Management	30
2.2.8.internet protocole	30
3.Client/Server architecture	31
3.1.types of client-server architecture	32
3.1.1.level 2 architecture	32
3.1.2.level 3 architecture	33
3.1.3. Multi-level architecture	34

4. Conclusion	34
Chapter 03 : Analysis and Design	
1. Introduction	35
2. Modeling methods and tools	35
2.1. UML modeling language	35
2.1.1.Définition	35
2.1.2.Usefulness of UML	36
2.1.3.UML formalism	37
2.1.4.Advantages and disadvantages of UML	39
2.2.4.1. Strengths of UML	39
2.2.4.2.Weak points of UML	39
2.2.Modeling tools	40
2.2.1 Visual Paradigm Online	40
2.2.2 Drawio.Desktop	40
3. UML design	41
3.1 usecase diagram	41
3.2. sequence diagram	42
3.3.acitivity diagram	42
4. Conclusion	43
Chapitre 04 : Implémentation	
1. Introduction	44
2. web developement framework	44
2.1. why use a framework	44
2.3. argument for the choice of laravel framework	45
3. the laravel php framework	45
3.1. Definition	45
3.2. Structure of laravel application	46
3.3. why use laravel?	48
3.4. the advantages of laravel	49
3.5. laravel features	50
3.6. laravel MVC architecture laravel	52
4.developement ennvironment	53

4.1. hardware environment	53
4.2. Software environment	53
4.2.1 Visual Studio Code	53
4.2.2 XAMPP	54
4.2.3 Composer	54
4.3.developement languages and tools	54
4.3.1. HTML (HyperText Markup Language)	54
4.3.2. CSS (Cascading Style Sheets)	55
4.3.3. JavaScript	55
4.3.4. jQuery	56
4.3.5. Bootstrap	56
4.3.6. PHP (Hypertext Preprocessor)	57
4.3.7. SQL (Structured Query Language)	57
4.4.web libraries used	58
4.4.1. PHPMailer	58
4.4.2. ChartJS	58
4.4.3. Laravel Livewire	58
4.4.4. Twilio SMS	59
5.database implementation	59
5.1. Choice of MYSQL SGDB	59
5.2. database management tool « phpMyAdmin »	59
6. the interfaces of our web application	60
7. Conclusion	64
Conclusion générale	65

Figures table

Chapter 1 : electronic payment	
Figure 1.1 : an image showing different types of electronic payment	03
Figure 1.2 : mastercard for bank bnp Paribas	05
Figure 1.3 : visacard for bank paysera.	05
Figure 1.4 : An image showing a method of transferring money from one bank to another	05
Figure 1.5 : An image showing the types of ewallet used in electronic payment.	06
Figure 1.6 : Image showing how electronic payment works.	07
Figure 1.7 : paypal logo	08
Figure 1.8 : skrill logo	08
Figure 1.9 : neteller logo	09
Figure 1.10 : CIB classic card.and CIB golden card	11
Figure 1.11 : distribution of payments cards in Algeria 2021	12
Figure 1.12 : amount of online transactions	13
Chapter 2 : web technologies	
Figure 2.1 : Transfert with static site	18
Figure 2.2 : Transfert with dynamic site.	19
Figure 2.3 : web Application.	19
Figure 2.4 : relation between browser and servers and devices	20
Figure 2.5 : the structure of api	26
Figure 2.6 : SOAP service web service	27
Figure 2.7 : Structure of RESTful web service	28
Figure 2.8 : Client/server architecture	32
Figure 2.9 : 2-tier architecture	33
Figure 2.10 : 3-tier architecture .	33
Figure 2.11 : architecture n-tier .	34
Chapter 3 : Analysis and design	
Figure 3.1 : UML Logo.	36
Figure 3.2 : the 4 + 1 views of UML	38
Figure 3.3 : Different type of UMl diagram .	38

Figure 3.4 : Uml element model	39
Figure 3.5: Visual Paradigm Online Logo.	40
Figure 3.6 : Drawio.desktop Logo.	40
Figure3.7 : use case diagram	42
Figure 3.8: sequence diagram	43
Figure3. 9 : acitivity diagram	44
Chapter 4 : Implementation	
Figure 4.1 : Logo of Laravel framework.	46
Figure 4.2 : MVC architecture.	52
Figure 4.3 : Logo of Visual Studio Code.	53
Figure 4.4 : XAMPP Logo.	54
Figure 4.5 : Logo of Composer.	54
Figure 4.6: Logo of HTML.	55
Figure 4.7: Logo of CSS.	55
Figure 4.8: Logo of JavaScript.	56
Figure 4.9: jQuery Logo.	56
Figure 4.10: Bootstrap Logo.	56
Figure 4.11: PHP Logo.	57
Figure 4.12: SQL Logo.	57
Figure 4.13: MySQL Logo.	59
Figure 4.14: Interface of phpMyAdmin.	60
Figure 4.15 :page of payment	61
Figure 4.16 : step enterd information of login	62
Figure 4.17 :last step of payment	63

Table lists

Chapter 2 : Web technologies	
Table 1.1 : Difference between frontend and backend	21
Table 1.2 : Web protocole and their use	31