



**POLITECNICO**  
**MILANO 1863**

Department of Electronics, Information and Bioengineering  
Doctoral Programme In Information Technology

# Doctoral Dissertation: a Latex Template

Doctoral Dissertation of:  
**Name Surname**

Advisor: Prof. Name Surname

Co-advisors: Name Surname, Name Surname

Tutor: Prof. Name Surname

Year 2023 - XXXV Cycle



# Abstract

Here goes the Abstract in English of your thesis followed by a list of keywords. The Abstract is a concise summary of the content of the thesis (single page of text) and a guide to the most important contributions included in your thesis. The Abstract is the very last thing you write. It should be a self-contained text and should be clear to someone who hasn't (yet) read the whole manuscript. The Abstract should contain the answers to the main scientific questions that have been addressed in your thesis. It needs to summarize the adopted motivations and the adopted methodological approach as well as the findings of your work and their relevance and impact. The Abstract is the part appearing in the record of your thesis inside POLITesi, the Digital Archive of PhD and Master Theses (Laurea Magistrale) of Politecnico di Milano. The Abstract will be followed by a list of four to six keywords. Keywords are a tool to help indexers and search engines to find relevant documents. To be relevant and effective, keywords must be chosen carefully. They should represent the content of your work and be specific to your field or sub-field. Keywords may be a single word or two to four words.

**Keywords:** here, the keywords, of your thesis



# Contents

<b>Abstract</b>	<b>i</b>
<b>Contents</b>	<b>iii</b>
 <b>Introduction</b>	 <b>1</b>
<b>1 Context and Problem Set</b>	<b>3</b>
1.1 Context of the project . . . . .	3
1.1.1 Project framework . . . . .	3
1.1.2 Overview of the Host Comapany . . . . .	3
1.2 Core idea of the project . . . . .	4
1.3 State of the art . . . . .	5
1.3.1 Existing solutions . . . . .	5
1.3.2 Critique of Existing solutions . . . . .	9
1.4 Suggested solution . . . . .	9
1.4.1 Text Description . . . . .	9
1.5 Project management framework . . . . .	10
1.5.1 Scrum . . . . .	10
1.5.2 XP . . . . .	12
1.5.3 Assesment and Comparison . . . . .	14
1.5.4 Selection of Method . . . . .	15
 <b>2 Requirements Analysis and Specification</b>	 <b>17</b>
2.1 Gathering and analyzing the needs . . . . .	17
2.1.1 Identification of actors . . . . .	17
2.1.2 Functional needs . . . . .	17
2.1.3 Non-Functional needs . . . . .	19
2.2 requirements specification . . . . .	20
2.2.1 general use case diagram . . . . .	20

2.2.2	Detailed use case diagrams . . . . .	20
2.2.3	System sequence diagrams . . . . .	22
2.3	Product backlog and project management with Agile Scrum . . . . .	26
2.3.1	The SCRUM roles . . . . .	26
2.3.2	Product Backlog . . . . .	27
2.3.3	Project management with Agile Scrum . . . . .	29
<b>3</b>	<b>PROJECT INITIALIZATION</b>	<b>31</b>
3.1	Programming Language . . . . .	31
3.1.1	Comaparative Study . . . . .	31
3.1.2	Choosing The Promgramming Language . . . . .	32
3.2	Exploring Development Frameworks . . . . .	33
3.2.1	Front-End . . . . .	33
3.2.2	Back-end . . . . .	37
3.2.3	API'S . . . . .	39
3.2.4	Database . . . . .	40
3.2.5	Scraping Tools . . . . .	42
3.3	Architectural Model . . . . .	44
3.4	Software Environment . . . . .	45
<b>4</b>	<b>Release 1:Administrator</b>	<b>47</b>
4.1	Figures, Tables and Algorithms . . . . .	48
4.1.1	Figures . . . . .	48
4.1.2	Tables . . . . .	49
4.1.3	Algorithms . . . . .	50
4.2	Theorems, propositions and lists . . . . .	51
4.2.1	Theorems . . . . .	51
4.2.2	Propositions . . . . .	51
4.2.3	Lists . . . . .	51
4.3	Use of copyrighted material . . . . .	51
4.4	Plagiarism . . . . .	51
4.5	Bibliography and citations . . . . .	52
<b>5</b>	<b>Conclusions and future developments</b>	<b>53</b>
	<b>Bibliography</b>	<b>55</b>

A Appendix A	57
B Appendix B	59
List of Figures	61
List of Tables	63
List of Symbols	65
Acknowledgements	67





# Introduction

This document is intended to be both an example of the Polimi L<sup>A</sup>T<sub>E</sub>X template for Master Theses, as well as a short introduction to its use. It is not intended to be a general introduction to L<sup>A</sup>T<sub>E</sub>X itself, and the reader is assumed to be familiar with the basics of creating and compiling L<sup>A</sup>T<sub>E</sub>X documents (see [7, 9]).

The cover page of the thesis must contain all the relevant information: title of the thesis, name of the Study Programme and School, name of the author, student ID number, name of the supervisor, name(s) of the co-supervisor(s) (if any), academic year. The above information are provided by filling all the entries in the command `\puttitle{}` in the title page section of this template.

Be sure to select a title that is meaningful. It should contain important keywords to be identified by indexer. Keep the title as concise as possible and comprehensible even to people who are not experts in your field. The title has to be chosen at the end of your work so that it accurately captures the main subject of the manuscript.

Since a thesis might be a substantial document, it is convenient to break it into chapters. You can create a new chapter as done in this template by simply using the following command

```
\chapter{Title of the chapter}
```

followed by the body text.

Especially for long manuscripts, it is recommended to give each chapter its own file. In this case, you write your chapter in a separated `chapter_n.tex` file and then include it in the main file with the following command

```
\input{chapter_n.tex}
```

It is recommended to give a label to each chapter by using the command

```
\label{ch:chapter_name}%
```

where the argument is just a text string that you'll use to reference that part as follows:

*Chapter ?? contains AN INTRODUCTION TO ....*

If necessary, an unnumbered chapter can be created by



# 1 | Context and Problem Set

## Introduction

In this introductory chapter, We begin by discussing the overall context of my project. Next, We present the general idea behind my project. We then examine existing solutions and their limitations before introducing my proposed solution. Finally, We wrap up this chapter by analyzing various development methodologies and comparing them to determine the most suitable approach for developing our web application.

### 1.1. Context of the project

Within this section , We will provide information about the framework of the project and we'll present some details of the host company.

#### 1.1.1. Project framework

As a student of the Innovation and Project Management Master's degree program at the Higher Institute of Computer Science and Multimedia of Sfax , I am required to work on an end-of-studies project to obtain my degree. This project is being completed under the mentorship of professionals at Wimobi, the company that has hosted me throughout the duration of the project.

#### 1.1.2. Overview of the Host Comapany

Created in 2017, Wimobi is an international digital agency that brings its expertise and know-how in three areas: Web, mobile, and design. With a culture of innovation at the heart of its DNA, Wimobi has a track record of over 30 challenging projects and more than 30 talents and experts.



Figure 1.1: Wimobi logo

**contacts of Wimobi**

<b>Adress</b>	Prestige building , Avenue Taieb Mhiri , 5111 , Mahdia city
<b>Phone</b>	+216 73 655 822
<b>Email</b>	contact@wi-mobi.com
<b>Website</b>	www.wi-mobi.com

Table 1.1: information about the Wimobi.

## 1.2. Core idea of the project

The rapid growth of e-commerce has transformed the way consumers shop and interact with businesses. In Tunisia, this trend has created new opportunities for consumers to save time and money by taking advantage of online promotions. Our project aims to capitalize on this trend by developing a website that scrapes and aggregates promotions from existing e-commerce websites in Tunisia. By constantly updating the scraped data, our website will provide consumers with a one-stop-shop for finding the best deals and special offers. This innovative approach has the potential to revolutionize the e-commerce landscape in Tunisia and empower consumers to make more informed purchasing decisions.

## 1.3. State of the art

The successful completion of any project requires a thorough analysis of existing applications. This analysis involves identifying the strengths and weaknesses of current solutions and assessing the needs of users. The insights gained from this analysis can inform the design and implementation of the project by highlighting key features and considerations. To accurately capture user needs, it is important to consider a diverse range of existing applications and evaluate their content and quality.

### 1.3.1. Existing solutions

We suppose that we are an internet user seeking to find best promotions in tunisia so as a francophone country we will automatically write 'promotions en tunisie' as a keyword in the search engine. And here are the results of search

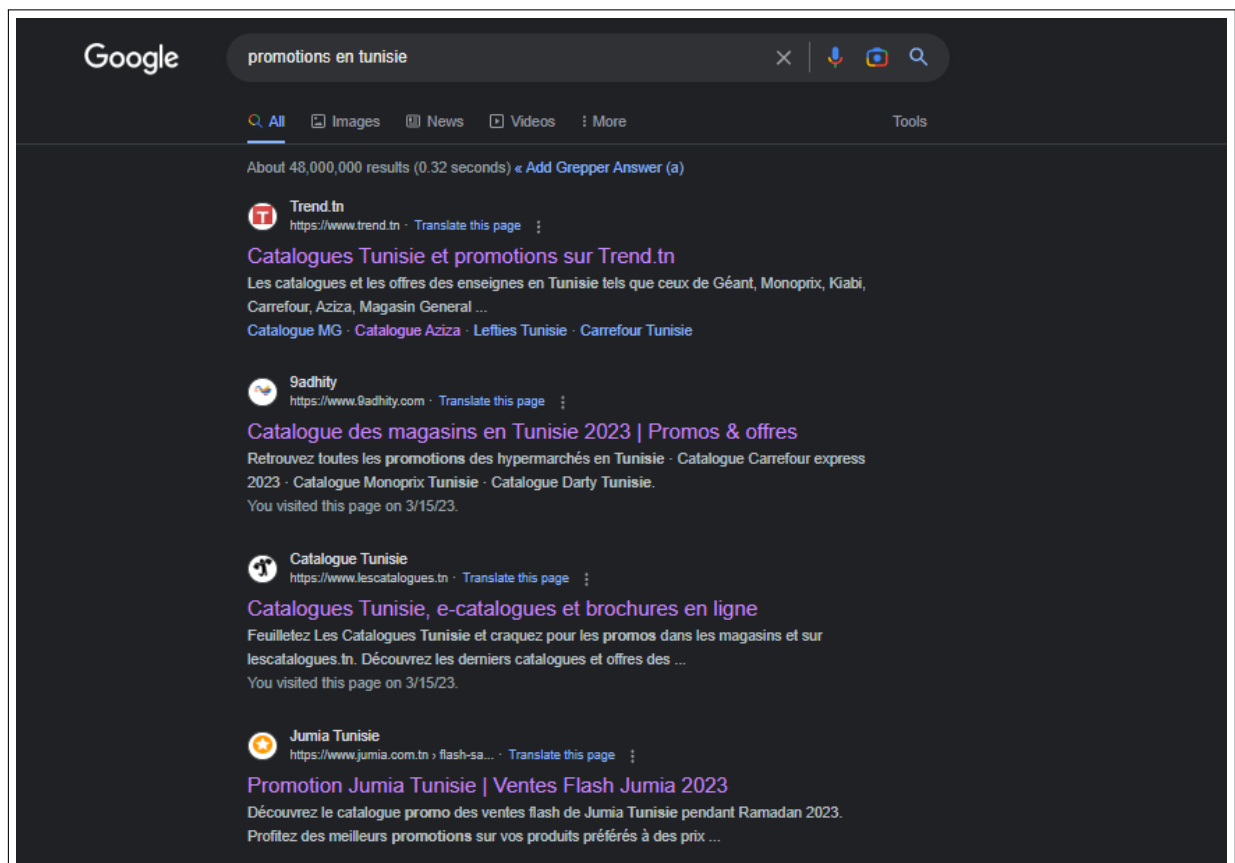


Figure 1.2: standing of the websites of promotions in tunisia

## Website 1: Trend.tn

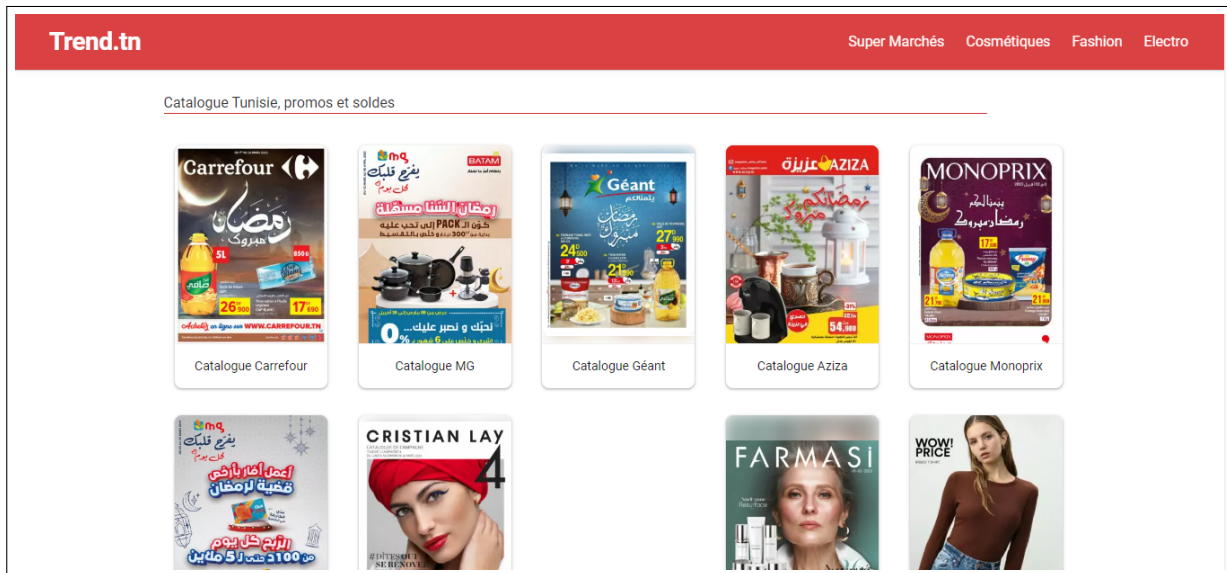


Figure 1.3: home page of trend.tn

The main goal of the website is to provide users with a comprehensive collection of catalogs, discounts, and promotions from a wide range of retailers across various sectors and categories. In addition to this, the platform also offers information about the location and other relevant data about the stores. This is all part of the range of services and information that the platform provides to its users.

## Website 2: 9adhity.com



Figure 1.4: home page of 9adhity.com

"9adhity" is a website that provides users with a comprehensive collection of offers and promotions from different stores in Tunisia such as Géant, Carrefour, Monoprix, Magasin Général, and Aziza. The website presents all up-to-date catalogs and allows users to create alerts based on the store or product family. With "9adhity", users can make informed choices for smart shopping.

lescatalogues.tn



Figure 1.5: home page of lescatalogues.tn

LesCatalogues.tn is a website that provides users with a comprehensive collection of catalogs and promotions from various stores and brands in Tunisia. It is the first free web-to-store portal available to Tunisian internet users. The site allows users to easily identify updated promotions to make their shopping more affordable



### 1.3.2. Critique of Existing solutions

Feature	Trade	9adhity	LesCatalogues.tn
Show catalog of promotions	Yes	Yes	Yes
Link to products under discount	No	No	No
Online purchase	No	No	No
Show ads of products under discount	No	Yes	No
Number of supermarket brands supported	More than 6	6	Not specified
use scraping	No	No	No
Admin space	Not sure	Not sure	Not sure
Support e-commerce	No	No	No
User feedback on products and promotions	No	No	No
User authentication and register	No	No	No
support alerts /notification system	No	Yes	No
save promotions	No	No	No

## 1.4. Suggested solution

Considering the criticisms of the current situation and the previously mentioned results, it becomes clear that none of the existing solutions adequately address our issue. As a result, we propose a reliable solution that fulfills our requirements and goals.

### 1.4.1. Text Description

The proposed solution must address the shortcomings identified in existing solutions. With this in mind, we have decided to create and develop our own website that will offer the following features:

- The website should have a feedback system, allowing users to rate and review promotions
- The website should redirect users to the links of the products on discount or special offers in general
- The website should have a search bar to filter results
- The website should have a backoffice to manage the website

- The website should have authentication and registration
- The website should have notification system for authenticated users.

## 1.5. Project management framework

To successfully complete a project, it is essential to follow a well-defined methodology that guides the team through each stage of the project, from planning to execution. Choosing the right methodology is crucial for the success of the project. In this section, we will introduce several popular project management methodologies and compare their strengths and weaknesses to help you make an informed decision.

### 1.5.1. Scrum

Scrum is a process framework used to manage product development and other knowledge work. Scrum is empirical in that it provides a means for teams to establish a hypothesis of how they think something works, try it out, reflect on the experience, and make the appropriate adjustments. That is, when the framework is used properly. Scrum is structured in a way that allows teams to incorporate practices from other frameworks where they make sense for the team's context. [1]

In Fact, Scrum is an agile method for managing projects that is based on dividing the project into short iterations called sprints. Each sprint typically lasts between two and four weeks and begins with a planning meeting where the team defines the sprint goal and breaks down the work into smaller tasks. At the end of each sprint, the team demonstrates what they have accomplished and evaluates their progress. Before starting the next sprint, the team conducts a retrospective to reflect on their process and identify areas for improvement. Scrum includes several key concepts and roles that are important to understand in order to effectively use the framework.

### Scrum Team

**Product Owner :** The Product Owner is responsible for maximizing the value of the product and managing the Product Backlog. They represent the needs of stakeholders and make decisions about the content and ordering of the Product Backlog.

**Scrum Master :** The Scrum Master is responsible for promoting and supporting Scrum by helping everyone understand Scrum theory and practice. They serve the Scrum Team, Product Owner, and organization by enabling the team to improve its practices and re-

moving impediments (obstacles/roadblocks) to progress.

**Development Team :** The development team consists of the people who deliver the product increment inside a Sprint. The Development Team is responsible for delivering a valuable product increment during each Sprint. They self-organize to determine how to divide the work.

The members of the scrum team has to be familiar with the following key concepts we can divide those concepts into two big points artifacts and events:

## Events

**Sprint :** A Sprint is a timebox of one month or less during which the team produces a potentially shippable product Increment. Sprints have consistent duration and start immediately after the previous one ends. The start and end dates are fixed.[1]

**Sprint planning :** A team starts out a Sprint with a discussion to determine which items from the product backlog they will work on during the Sprint. The end result of Sprint Planning is the Sprint Backlog. [1]

**Daily Scrum :** The Daily Scrum is a brief meeting where the team plans their activities for the next day. It is not for status reporting or problem-solving.

**Sprint Review :** At the end of the Sprint, the team reviews the results with stakeholders to get feedback. The Sprint Review is not a status report. Feedback is placed into the Product Backlog for future consideration.

**Sprint Retrospective :**After the Sprint Review, the team reflects on the previous sprint and identifies adjustments.

## Artifact

**Backlog Product :** It is used to break down the project into a set of “User Stories” classified and sorted by priority.

**Sprint Backlog :** The Sprint Backlog is the collection of product backlog items and tasks necessary to deliver them and achieve the Sprint Goal.

**Increment :** The increment is the collection of the Product Backlog Items that meet the team’s Definition of Done by the end of the Sprint. The Product Owner may decide to release the increment or build upon it in future Sprints. [1]

**Definition of Done :** The definition of done is a team’s shared agreement on the criteria that a Product Backlog Item must meet before it is considered done.[1]

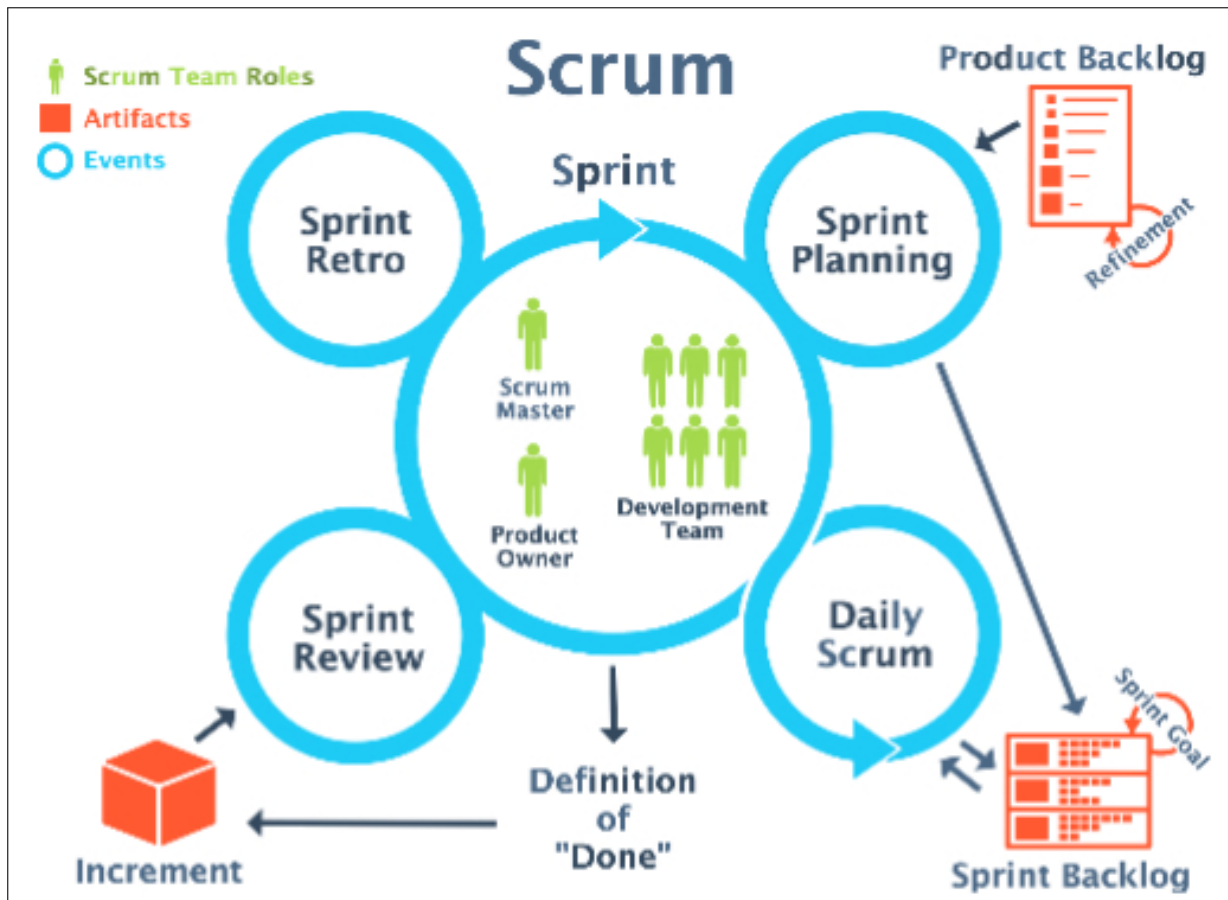


Figure 1.6: scrum framework process

### 1.5.2. XP

#### Overview

Extreme Programming (XP) is an agile software development methodology that prioritizes close collaboration, adaptability, and the delivery of high-quality software solutions. It addresses the challenges of rapidly changing requirements and promotes continuous improvement throughout the development process.

#### Core Principles

XP is built on a set of core principles that emphasize communication, simplicity, feedback, and courage. These principles guide the practices and behaviors of XP teams, fostering a customer-centric and adaptable approach to software development.

### Customer Collaboration:

XP places a strong focus on customer satisfaction and responsiveness to market demands. It emphasizes frequent and open communication with customers to understand their needs, gather feedback, and deliver value through short development iterations.

### Testing and Continuous Integration:

XP advocates for frequent testing and continuous integration. Developers write automated tests to validate software functionality, enabling rapid feedback and early identification of issues. Continuous integration ensures regular integration of code changes, reducing conflicts and allowing for immediate bug fixes.

### Sustainable Pace:

XP promotes a sustainable pace of work, encouraging a balanced workload and avoiding burnout. It emphasizes team member ownership and responsibility for tasks, fostering a sense of accountability and empowerment.

### Simplicity in Design:

XP encourages simplicity in software design and implementation. By keeping the design straightforward and reducing complexity, teams can facilitate easier maintenance and future enhancements.

### Benefits of XP:

By embracing XP, development teams can respond quickly to changing requirements, deliver valuable software incrementally, and continuously learn and improve their processes. XP enhances the quality of software solutions, improves productivity, and provides a framework for effective teamwork.

By following the principles and practices of Extreme Programming, software development teams can cultivate a collaborative and adaptive environment, leading to the development of high-quality software solutions that meet the evolving needs and expectations of customers. the following figure gives us a quick overview about the XP process

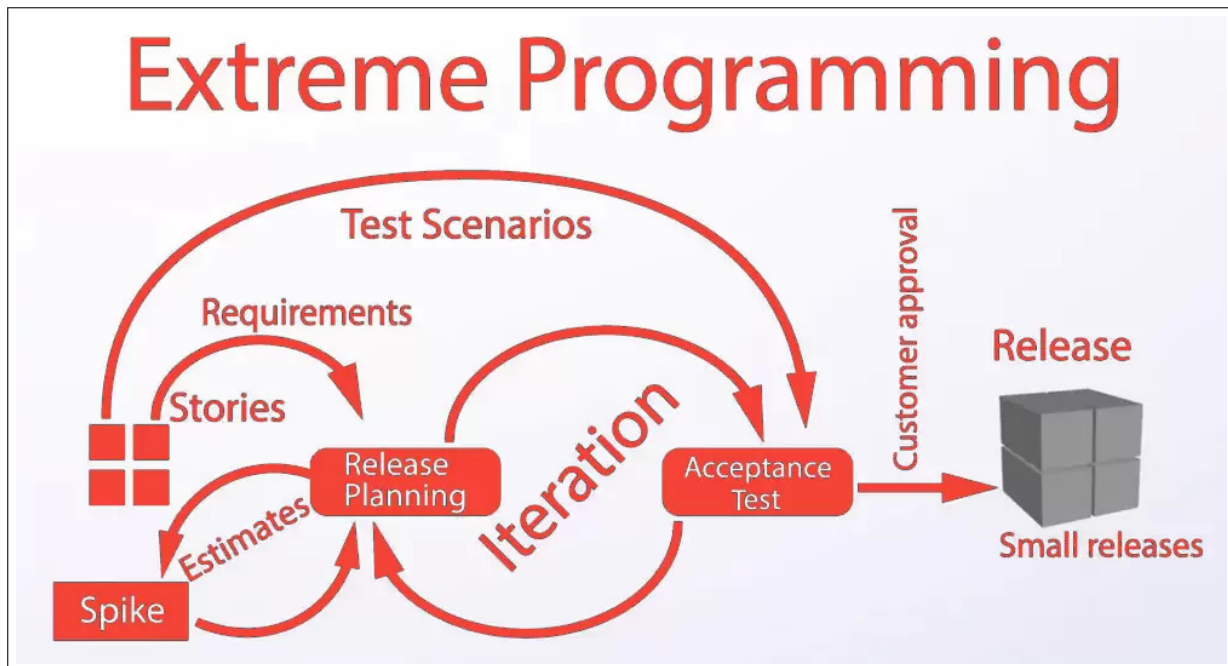


Figure 1.7: XP process

### 1.5.3. Assessment and Comparison

Methodology	Principles	Pros	Cons
<b>RUP</b>	It's an iterative software development process framework. It consists of four phases: inception, elaboration, construction, and transition.	- Clear rules for the team to follow. Prevents wasting costs and other resources.	- Heavily relies on the team's expertise.
<b>Spiral</b>	The spiral development methodology is an iterative software development process framework that puts a big emphasis on risk handling.	- Perfectly suits projects in which risk evaluation is crucial.	- Requires a lot of management effort.

Methodology	Principles	Pros	Cons
<b>XP</b>	It is an agile software development methodology that aims to achieve the advanced quality of software solutions and improve their responsiveness to ever-changing market conditions.	<ul style="list-style-type: none"> <li>- Documentation leads to cost-saving.</li> <li>- Short iterations allow a team to release frequently and test code thoroughly.</li> </ul>	<ul style="list-style-type: none"> <li>- Heavy reliance on communication.</li> </ul>
<b>Scrum</b>	The six principles of Scrum are empirical process control, self-organization, collaboration, value-based prioritization, timeboxing, and iterative development.	<ul style="list-style-type: none"> <li>- Result-oriented.</li> <li>- Effective use of a budget.</li> <li>- Better collaboration within a team and between a team and a client.</li> </ul>	<ul style="list-style-type: none"> <li>- Challenging to adopt in a short period and in large teams.</li> </ul>
<b>Waterfall</b>	The Waterfall model is based on three main principles: low customer involvement, strong documentation, and sequential structure.	<ul style="list-style-type: none"> <li>- Fits simple short-term projects.</li> <li>- Makes a budget, a timeline, and results predictable for the client.</li> </ul>	<ul style="list-style-type: none"> <li>- Lack of flexibility.</li> </ul>

Table 1.2: Software Development Methodologies

#### 1.5.4. Selection of Method

Based on the comparative study between the development methodologies, we have chosen the Scrum methodology for the following reasons:

Firstly, Scrum has been thoroughly developed and tested for short iterations. This allows us to adapt the software creation process according to the evolving needs of the project, leading to increased productivity of the team.

Moreover, Scrum is currently the most widely used agile method because of its efficiency and flexibility compared to conventional methods. It places the needs of the client at the center of project priorities, ensuring their satisfaction throughout the development process. By incorporating feedback from the client and stakeholders, Scrum enables the team to be more responsive and adaptive to customer expectations.

Additionally, Scrum provides greater visibility and transparency in project management. It allows for regular communication and collaboration within the team, ensuring everyone is aligned and aware of project progress. This visibility enhances the team's ability to identify and address issues promptly, resulting in improved project outcomes.

By adopting the Scrum methodology, we aim to achieve a more efficient and effective software development process, delivering high-quality solutions that meet the evolving needs and expectations of our clients.

## Conclusion

In this opening chapter, we began by introducing the general context of the project. We then conducted a review of existing solutions and outlined the work to be done, taking into account their limitations. Finally, we explored various development methodologies and ultimately chose the Scrum methodology for our project. The next chapter will focus on analyzing and specifying requirements.



# 2 | Requirements Analysis and Specification

## Introduction

### i may change

This chapter outlines the initial phase of our project, beginning with the identification of actors involved in the system. After that we proceed to examine both functional and non-functional requirements. These requirements are then represented through use case diagrams, which are further elaborated upon through the exploration of potential scenarios that can be executed by different actors. Finally, we conclude by organizing the project's sprints according to the principles of the SCRUM methodology that we have chosen to adopt.

## 2.1. Gathering and analyzing the needs

In this part of our report, we will first focus on identifying the actors and their roles. Subsequently, we will define both the functional and non-functional requirements that our application seeks to fulfill.

### 2.1.1. Identification of actors

An actor represents a user or another system that interacts with the system being developed in order to accomplish a specific task or achieve a particular goal.

- **Customer** : it's the actor who will take on advantage the promotions.
- **Administrator** : it's the actor responsible for managing the website.

### 2.1.2. Functional needs

Functional requirements describe the features and capabilities that an application must have in order to meet the needs of its users. These requirements specify what the system

should do and are essential for ensuring that the system meets the expectations of its users.

the Website should enables the **Customers** to:

- **Register (signup) :** The register functionality enables new users to create an account on the platform, allowing them to access its services and features.
- **Authenticate (sign in) :** The “authenticate” functionality allows users to verify their identity by providing their login credentials. This enables them to access the platform’s services and features that are only available to authenticated users.
- **Redirect customers to the promotions :** This functionality allows authenticated users to view and access promotions offered by the original platform. This can be achieved by providing a link or button that redirects users to the original page of the offer. By doing so, users can easily view and take advantage of the promotions available to them.
- **Rate a promotion :** This allows users to provide feedback on promotions offered by the platform. This can be achieved by allowing users to rate the promotion using a star rating system or by leaving a comment. By doing so, users can share their opinions and experiences with the promotion, providing valuable feedback to the platform and other users.
- **Search and filter promotions or products :** This enables users to easily find and view promotions and products that are relevant to their interests. This can be achieved by providing a search bar and filter options that enable users to narrow down the list of promotions based on specific criteria, such as category or price. By doing so, users can quickly find promotions that meet their needs and preferences, improving their overall experience on the platform.
- **Social Media Integration (share) :** Social media integration gives the users the option to share promotions from the website directly to their social media accounts. This can help increase the visibility and reach of the promotion, potentially attracting new customers .
- **Manage profile :** All users can manage their profiles by updating some personal data.

Moreover the website should enables the **Administrator** to :

- **Manage promotions(products) :** This allows website administrators to create, edit, delete and track promotional campaigns, such as special offers or discounts.
- **Manage customers :** This allows website administrators to view,delete,update and create customers information.
- **Manage scraping :** This gives the website administrator to manually scrap or schedule a scraping date to scrape new promotions.
- **Manage categories :** This empowers the website administrator to create ,read,update

and delete categories .

- Alert customers on new promotions : This option gives the website administrator the possibility to alert customers about new promotions by sending notifications or emails.

### 2.1.3. Non-Functional needs

Non-functional requirements are the conditions that must be met to ensure the system functions properly and to improve the quality of services for users. For our website, The next paragraph identifies the non-functional requirements.

**Usability:** The application must have a user-friendly interface that is easy to navigate and use.

**Convivability :** The success of a project often depends on user satisfaction. The usability of the user interface is crucial in achieving this goal. Our platform should have a user-friendly interface that is clear, simple, and easy to navigate. This will help users to easily access and use the platform's services, improving their overall experience and satisfaction .

**Performance :** Efficiency is an important non-functional requirement for any system. This means that the system must perform its tasks quickly and effectively, ensuring a minimum level of performance. By guaranteeing a minimum execution rate, the system can provide a consistent and reliable user experience. This can help to improve user satisfaction and increase the overall effectiveness of the system.

**Maintenance :** The code must be clear and flexible to allow future developments or improvements. It is also important to ensure that the platform is portable, expandable, reusable and reliable

**Security :** Security is a crucial non-functional requirement for any platform. Our platform must ensure the integrity and confidentiality of user data by implementing measures such as authentication and encryption. Authentication helps to verify the identity of users and control access to the platform's services. Encryption, such as hashing, can be used to protect sensitive data stored in the database by making it unreadable to unauthorized users. By implementing these security measures, our platform can provide a secure and reliable environment for users to access its services.

**Reliability :** The application must function correctly and consistently, with minimal downtime or errors.

**Scalability :** The application must be able to handle increasing amounts of work and accommodate growth in the number of users or data.

## 2.2. requirements specification

In this section, we provide an overview of the various features available on our platform, using both use cases and user stories. A user story describes a basic functional need from the user's perspective, while use cases offer a precise and efficient way to communicate how the system should be used. We begin by presenting a general use case diagram, followed by detailed use cases for each actor in the application.

### 2.2.1. general use case diagram

the following use case diagram, represents a general view of the functional behavior of the platform.

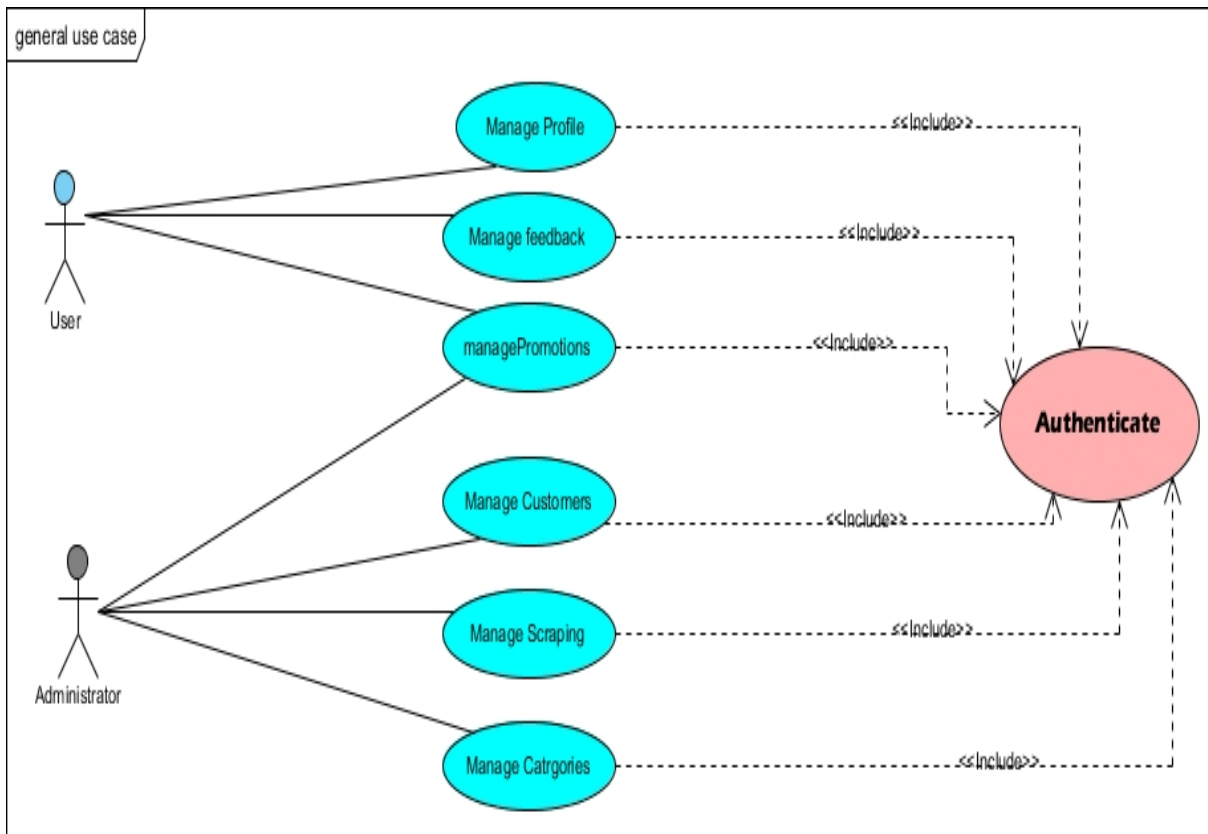


Figure 2.1: general use case

### 2.2.2. Detailed use case diagrams

In this section, we will present a detailed description of the use cases.

### Detailed use case diagram of actor "User"

The Figure 2.2 illustrates the features that can be done by the user after authentication. He can manage his profile, manage promotions, and manage feedback.

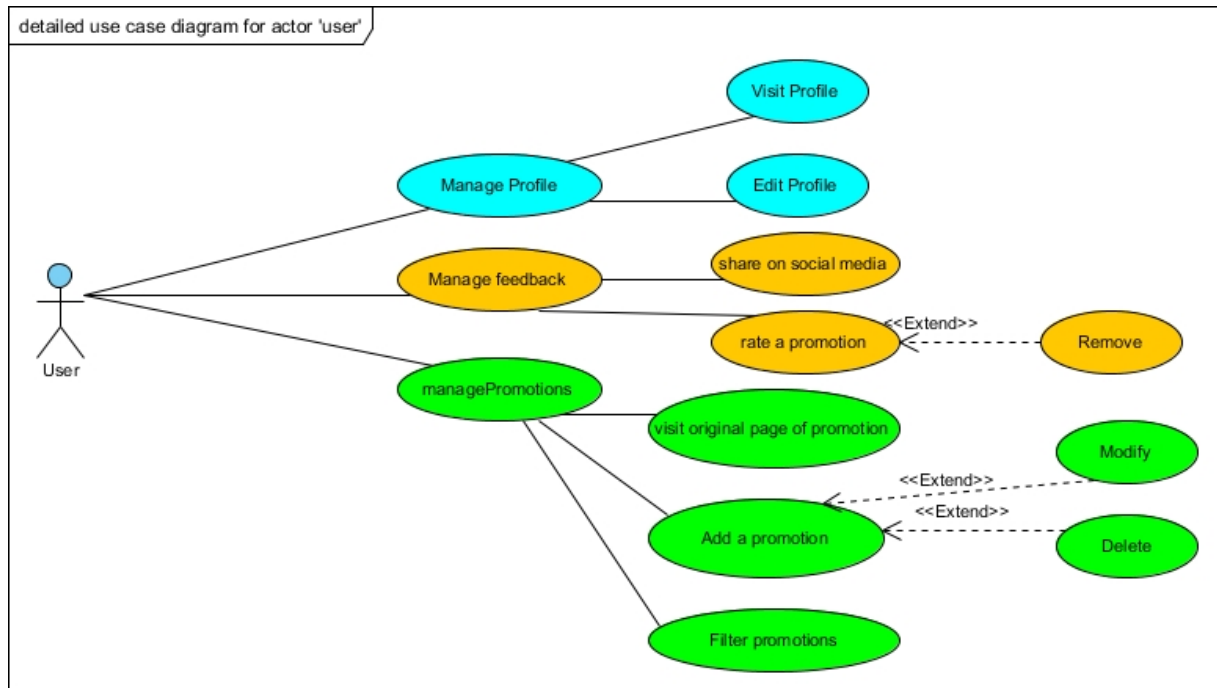


Figure 2.2: Detailed use case diagram for the actor "User"

### Detailed use case diagram of actor "Administrator"

After authentication, the Administrator is authorized to perform the features shown in Figure 2.3. We list: Manage Scraping, Manage Customers, Manage Categories, Manage Promotions.

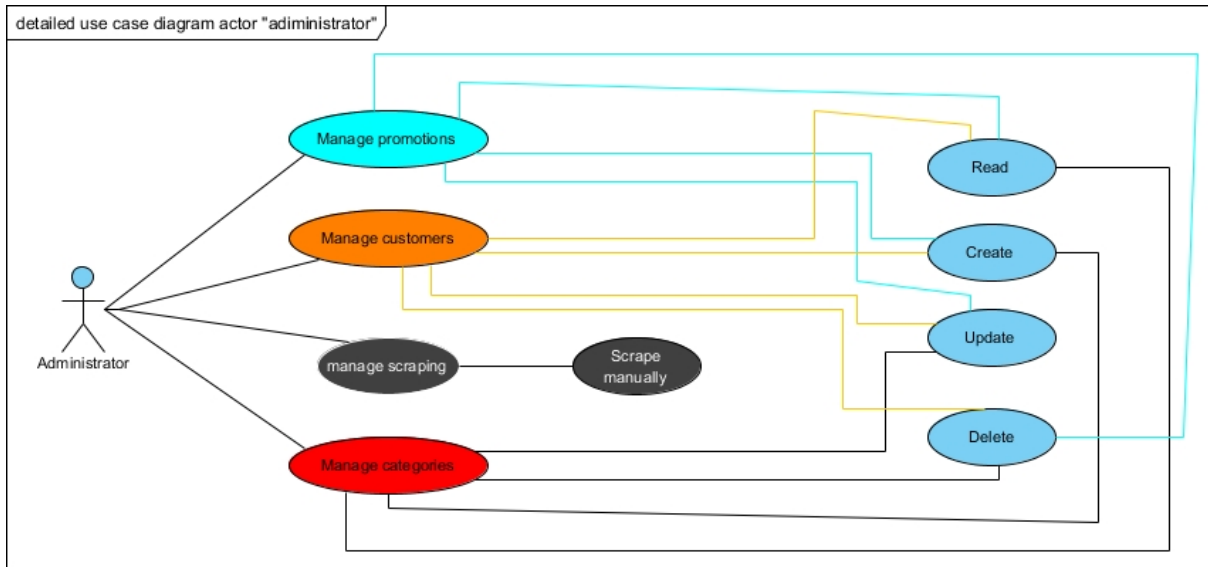


Figure 2.3: Detailed use case diagram for the actor "Administrator"

### 2.2.3. System sequence diagrams

A sequence diagram simply represents the interaction between objects in a sequential order, i.e. the order in which these interactions take place. We can also use the terms event diagrams or event scenarios to refer to a sequence diagram. Sequence diagrams describe how and in what chronological order objects works in a system.

Sequence Diagram Authenticate

The following table and figure explains the sequence diagram of authentication.

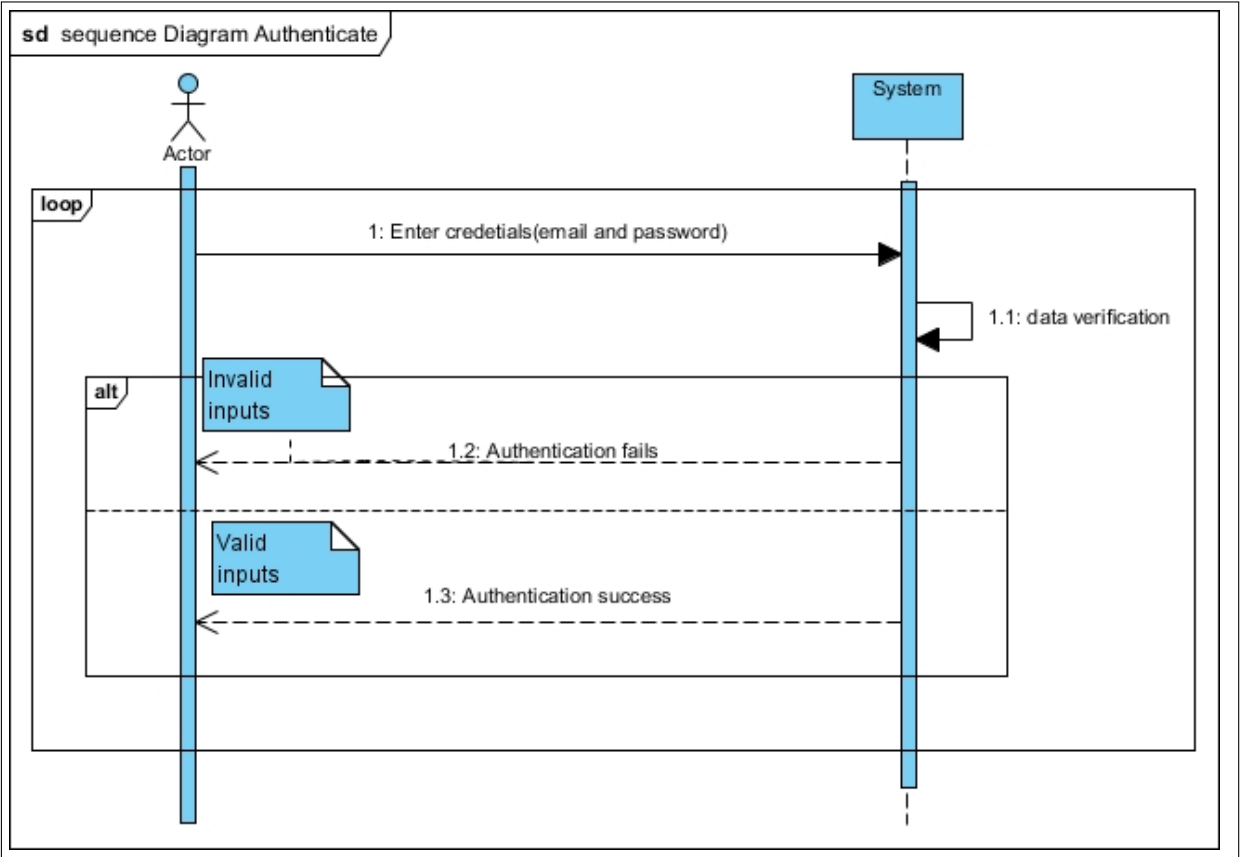


Figure 2.4: Sequence Diagram "Authenticate"

Table 2.1: Authenticate Scenario

<b>Title</b>	<b>Authenticate</b>
<b>Actors</b>	User
<b>Pre-Condition</b>	Unauthenticated user
<b>Post-Condition</b>	Authenticated user
<b>Main Scenario</b>	
<b>Actor Action</b>	<b>System Reactions</b>
User visits the authentication page.	The system displays the authentication page.
User enters their credentials and submits the form.	The system verifies the user's credentials. If the credentials are valid, the system authenticates the user. If the credentials are invalid, the system displays an error message.
<b>Exception</b>	
The data entered by the user is missing or invalid. The system displays an error message and the scenario is repeated.	

## Sequence Diagram Add Promotion

The table .. explains the sequence diagram "Add Promotion" displayed in the figure 2.4.



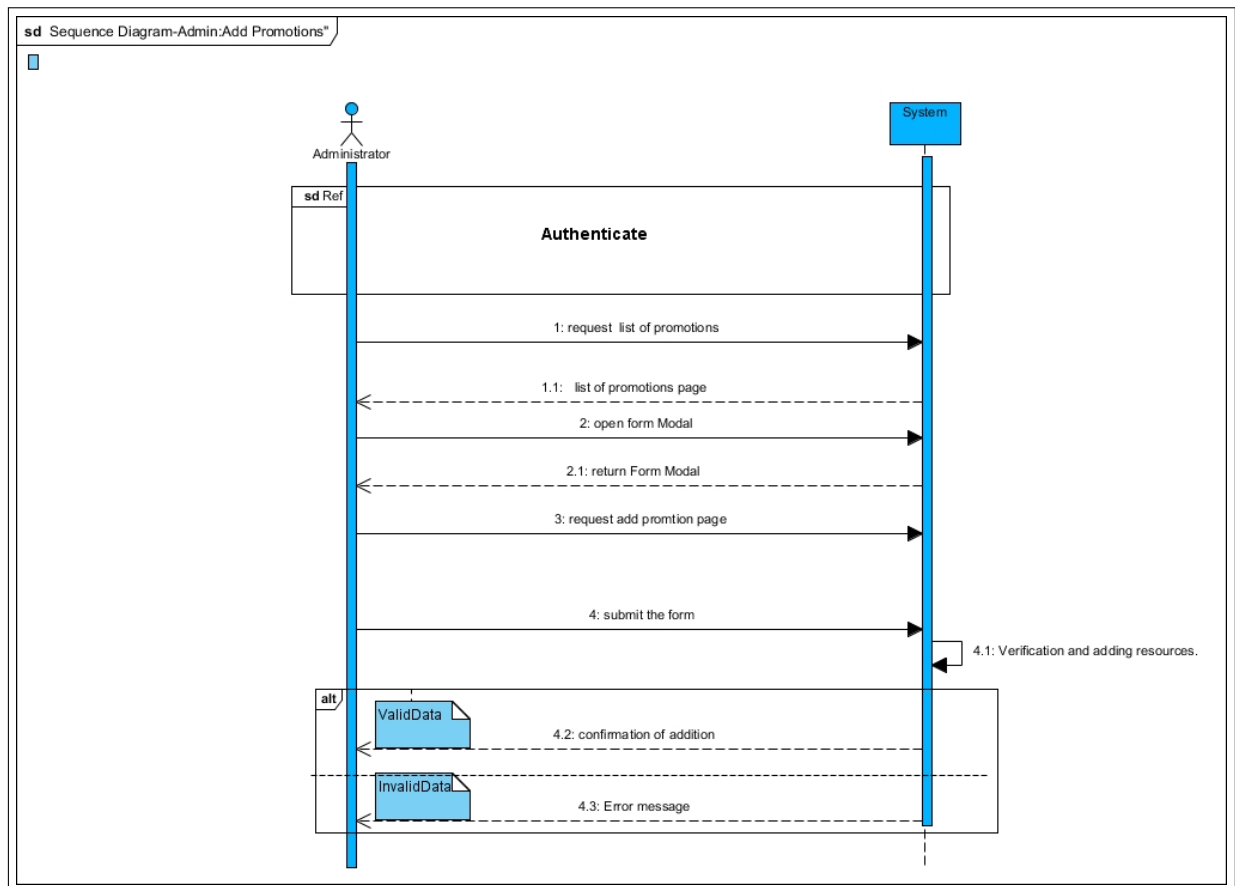


Figure 2.5: Sequence Diagrams "Add Promotion"

Table 2.2: Add Promotions Scenario

<b>Title</b>	Add Promotions
<b>Actors</b>	Administrator
<b>Pre-Condition</b>	Authenticated Administrator
<b>Post-Condition</b>	Promotion added
<b>Main Scenarios</b>	
<b>Actor Action</b>	<b>System Reactions</b>
1. The Administrator asks for the scrapped promotions list page. 3. The Administrator opens the form modal. 5. The Administrator submits the form.	2. The system returns the promotions list page. 4. The system returns the form modal. 6. The system returns the main page of promotions and add the new promotion to the list
<b>Exception</b>	
The data entered by the user is missing or incorrect. The system displays an error message and the scenario is repeated.	

## 2.3. Product backlog and project management with Agile Scrum

In this third section, we will represent the Scrum roles of the project, then the product backlog, and finally the project management using Agile Scrum methodology.

### 2.3.1. The SCRUM roles

In this section, we introduce all the participants involved in the different stages of the project and in the creation of the internship report. Our development team is made up of a single developer who is in charge of executing the project from design to development. The Product Owner, who represents the customer and user, identifies the requirements, sets priorities, and guides the development team in creating functionalities. The Scrum Master acts as a facilitator, ensuring that the project team operates smoothly and maintains a positive working environment.

The following Table lists the roles, associated actors, and their responsibilities.

Role	Mission	Actor
Scrum Team	Design. Development. Testing and validation. Deployment.	Med Noor Haj Ammar
Product owner	Definition of the main system needs and functionalities.	Fedi Dayeg
Scrum Master	Project approval.	Nedia Bouacida

Table 2.3: Project Team

### 2.3.2. Product Backlog

The Product Backlog is a list of customer requirements, known as “User Stories,” organized by priority.

Each User Story is a brief description of a functional requirement, written as “As an [actor], I want [an objective] for [a reason].”

The Product Backlog includes fields for Rank, Theme, User Story, and Priority. Priority is determined using the “MoSCoW” method, where each User Story is classified as Must (must be done), Should (should be done if possible), or Could (could be done if it does not impact other tasks).

ID	Theme	User Story	Priority
1	Register	As a new user, I want to be able to create an account on the platform, so I can access its services and features.	Must
2	Authentication	As a registered user, I want to be able to login , so that I can access its services and features that are only available for authenticated users..	Must

3	Profile Management	As a registered user, I want to be able to manage my profile by updating some personal data, so that I can keep my information up-to-date.	Should
4	view Promotions	As an authenticated user, I want to be able to view and access promotions offered by the platform, so that I can take advantage of them.	Must
5	Feedback Management	As an authenticated user, I want to be able to rate promotions offered by the platform, so that I can provide feedback and share my opinions and experiences with other users.	Should
6	Users management	As an administrator, I want to create, edit or delete a user.	Must
7	Filtering management	As an authenticated user, I want to be able to search and filter promotions or products based on specific criteria, so that I can quickly find promotions that meet my needs and preferences	Must
8	FeedBack Management	As an authenticated user, I want to be able to share promotions from the website directly to my social media accounts, so that I can increase the visibility and reach of the promotion.	Could
9	Promotions Management	As an administrator, I want to be able to create, edit, delete and track promotional campaigns, such as special offers or discounts, so that I can manage promotions on the platform.	Must

10	User Management	As an administrator, I want to be able to view, delete, update and create customer information, so that I can manage customers on the platform.	Must
11	Scraping Management	As an administrator, I want to be able to manually scrape to scrape new promotions, so that I can keep the platform up-to-date with new promotions.	Must
12	Scraping Management	As an administrator, I want to be able to manually schedule a scraping date to scrape new promotions, so that I can keep the platform up-to-date with new promotions...	Could
13	Manage Categories	As an administrator, I want to be able to create, read, update and delete categories, so that I can manage the categories on the platform.	Should
14	Alert management	As an administrator, I want to be able to alert customers about new promotions by sending notifications or emails, so that I can increase the visibility and reach of the promotions.	Could

### 2.3.3. Project management with Agile Scrum

#### Release:

A release refers to the delivery of a specific version of an application. It encompasses the entire period from the start of work on that version to its delivery, and includes a series of successive sprints.

#### Sprint:

In Scrum methodology, a sprint refers to an iteration, or a block of time dedicated to creating an increment of the potentially deliverable product. The goal of each sprint should be defined in business terms, rather than technical terms, so that it is easily understandable by all team members.

For our project, we have chosen to develop three releases illustrated by the next table. The first release is called “Administrator and Scraping”, the second is called “User” .

RELEASE	SPRINT	TASKS
Release 1:Administrator	Sprint 1	<ul style="list-style-type: none"> <li>• Manage categories</li> <li>• Second Manage customers</li> <li>• Manage scraping</li> <li>• Manage alert</li> <li>• manage promotions</li> </ul>
Release 2:User	Sprint 2	<ul style="list-style-type: none"> <li>• Feedback management</li> <li>• Profile Management</li> <li>• Filtering Management</li> </ul>

Table 2.5: Planning of Releases and Sprints

## Conclusion

In this chapter, we have determined the actors and outlined the functional and non-functional requirements of our application. Moreover We have presented diagrams for use cases and system sequences. Additionally, we have provided the product backlog and planned out the sprints and releases. The following chapter will focus on sprint 0, which involves initializing the project.

# 3 | PROJECT INITIALIZATION

## Introduction

### 3.1. Programming Language

#### 3.1.1. Comaparative Study

When building an apartment, the first step is to choose the stone that will be used in the construction. Similarly, when building a web application, the first step is to choose a programming language. Thus, we will present a comparative analysis between three popular programming languages.

**JavaScript** is a versatile language that can be used for both front-end and back-end development. It has a large and active community and many libraries available for web scraping. Its performance in front-end development is excellent, while its performance in back-end development is good.

**Python** is another versatile language that can be used for a wide range of tasks. It has an excellent selection of libraries for web scraping and its performance in back-end development is excellent. Python also has a large and active community.

**Java** is known for its strong performance in back-end development and its strong security features. It has a large and active community and many libraries available for web scraping. Java is less versatile than JavaScript and Python, but it is still a popular choice for many projects.

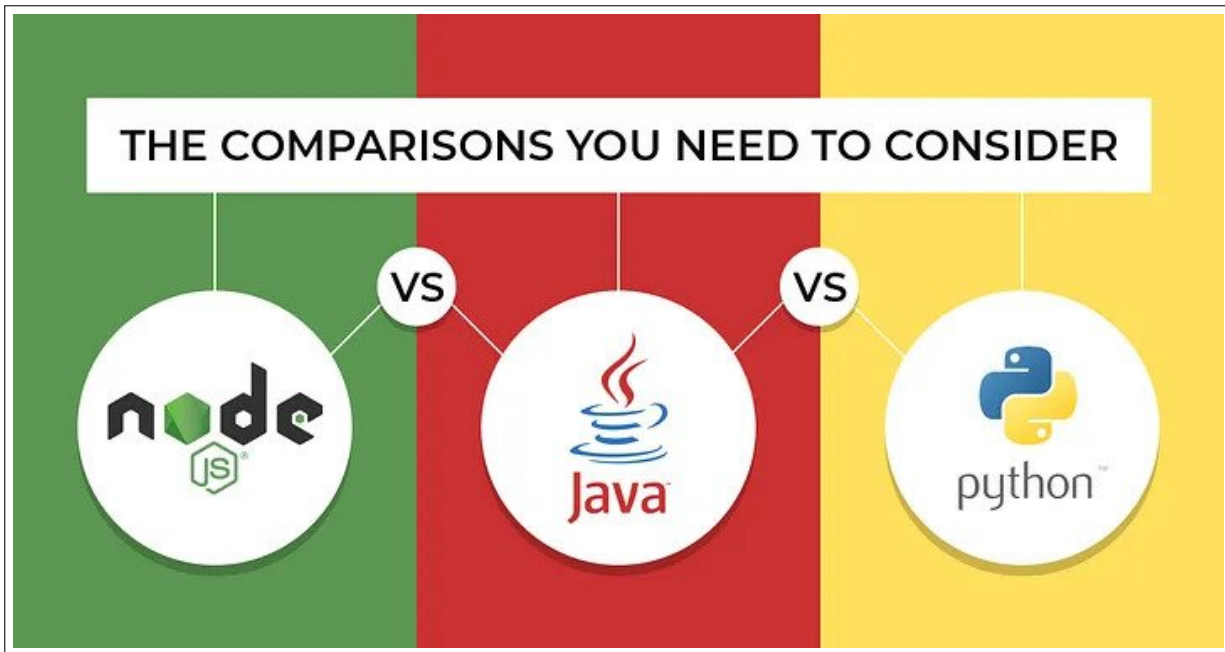


Figure 3.1: Most popular programming languages

Criteria	TypeScript (Node.js)	Python	Java	JavaScript
Versatility	5/5	4/5	3/5	4/5
Scraping Libraries	5/5	5/5	4/5	3/5
Front-End Performance	4/5	3/5	3/5	5/5
Back-End Performance	4/5	4/5	5/5	3/5
Performance with REST APIs	5/5	4/5	5/5	4/5
Performance with GraphQL	4/5	4/5	4/5	4/5
Community	5/5	5/5	4/5	5/5
Security	4/5	4/5	4/5	4/5

### 3.1.2. Choosing The Promgramming Language

The main reason for choosing Javascript is it's verstality and front-end efficiency ,Note that Node Js is JavaScript runtime environment for web applications where a rutime en-vironment is the execution environment for running software applications.Thus, instead of using native JS, and according to the table NodeJs with typescript is the most versatile technology and with the highest rate in the table if we calculate the average it's 4.5/5 .



## 3.2. Exploring Development Frameworks

In this section, we will discuss the Front-End and Back-End frameworks that we have chosen for the development of our website, and provide a comparison between them.

### 3.2.1. Front-End

The number of client-side development technologies is continuously increasing day by day. Three of the most popular languages are shown in Figure 3.1.

- ReactJs
- AngularJs
- VueJs

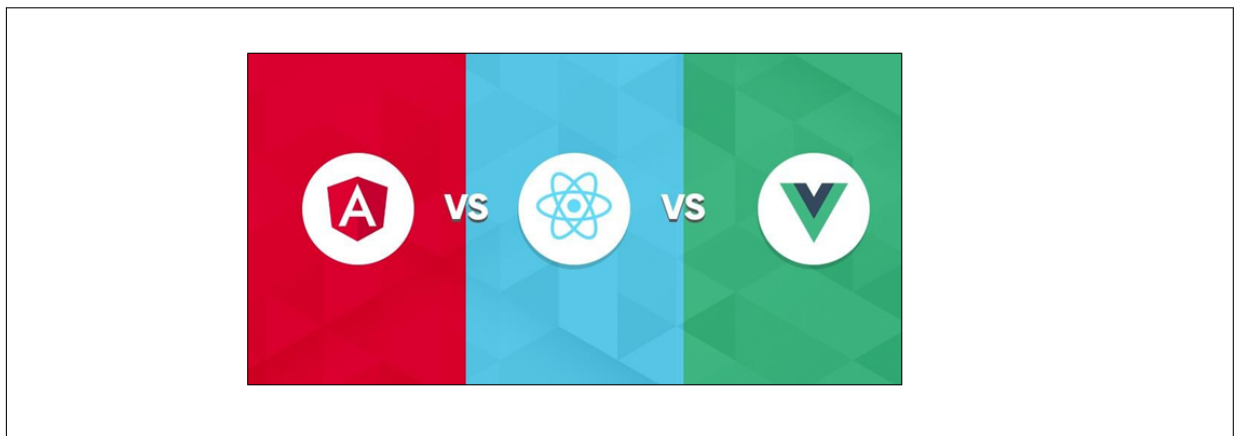


Figure 3.2: Most popular Front-End technologies

### React Js

React JS is an open-source JavaScript library developed by Facebook in 2013. It enables the development of web applications where each component can have its own independent state, separate from other components. The architecture of an application using React JS is shown below .

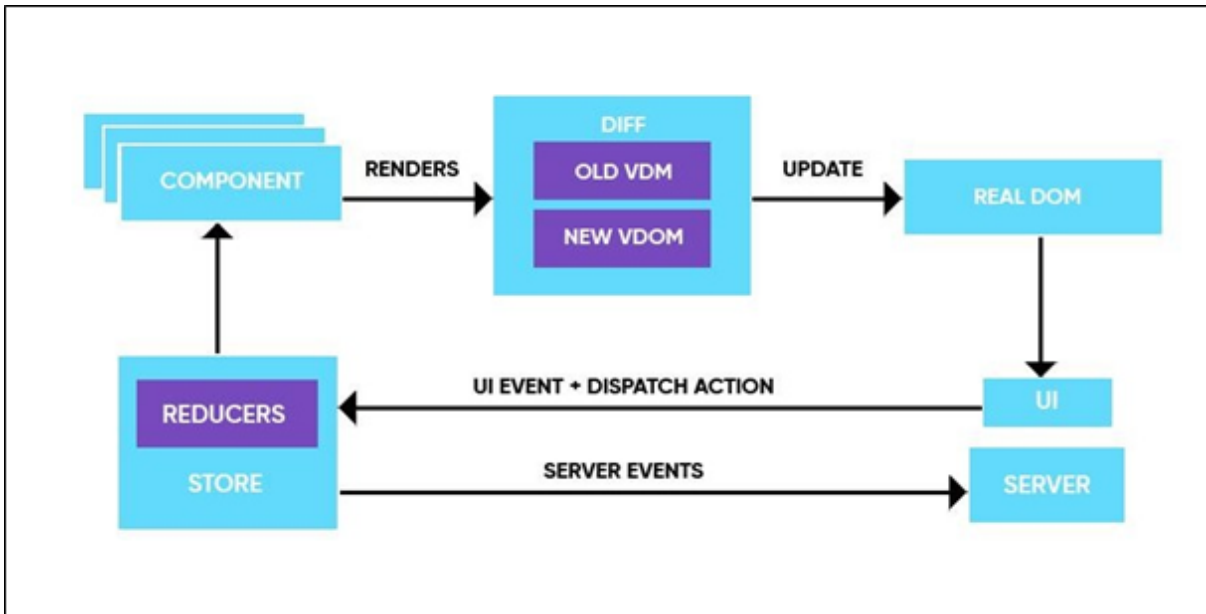


Figure 3.3: React JS architecture

- **Advantages:**

- React JS is used by several companies such as Facebook, Instagram, Netflix, Airbnb, Dropbox, etc.
- React JS uses a virtual DOM, which makes the application efficient in terms of speed and performance.
- JSX format is very similar to HTML.

- **Drawbacks:**

- Lack of documentation.
- React is just a library, unlike other technologies like Angular.
- The initial learning curve can be challenging.

## Angular Js

Angular is an open-source JavaScript Framework developed by Google in 2016, it allows the creation of single-page web applications (SPA). Its architecture, illustrated by Figure 3.4, provides to structure the code and separates the view (Interfaces) from the models.

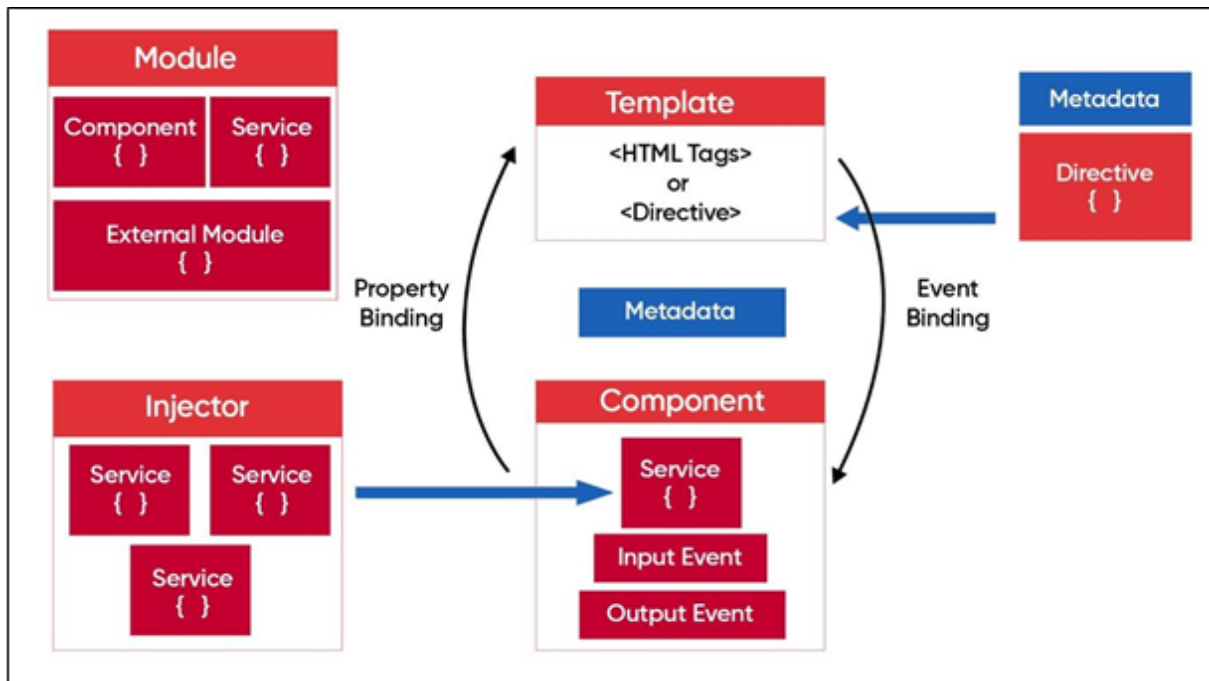


Figure 3.4: Angular architecture

- **Advantages:**

- Angular is one of the most powerful and robust JavaScript frameworks. It is very popular and handy since it supports HTML and CSS. It is used by several popular companies such as UpWork, PayPal, etc.
- Angular provides a command line interface (CLI) that offers several services, the most important of which is its ability to automatically create and generate code.

- **Drawbacks:**

- The learning curve for Angular can be long and difficult, which may be a barrier for some beginners.
- There have been some migration issues between old and new versions of Angular.

## Vue Js

Vue JS is a relatively new framework, having been created in 2014. Despite not being backed by a large company, it has gained significant popularity in recent years. The architecture of an application using Vue JS is shown in Figure 3.5.

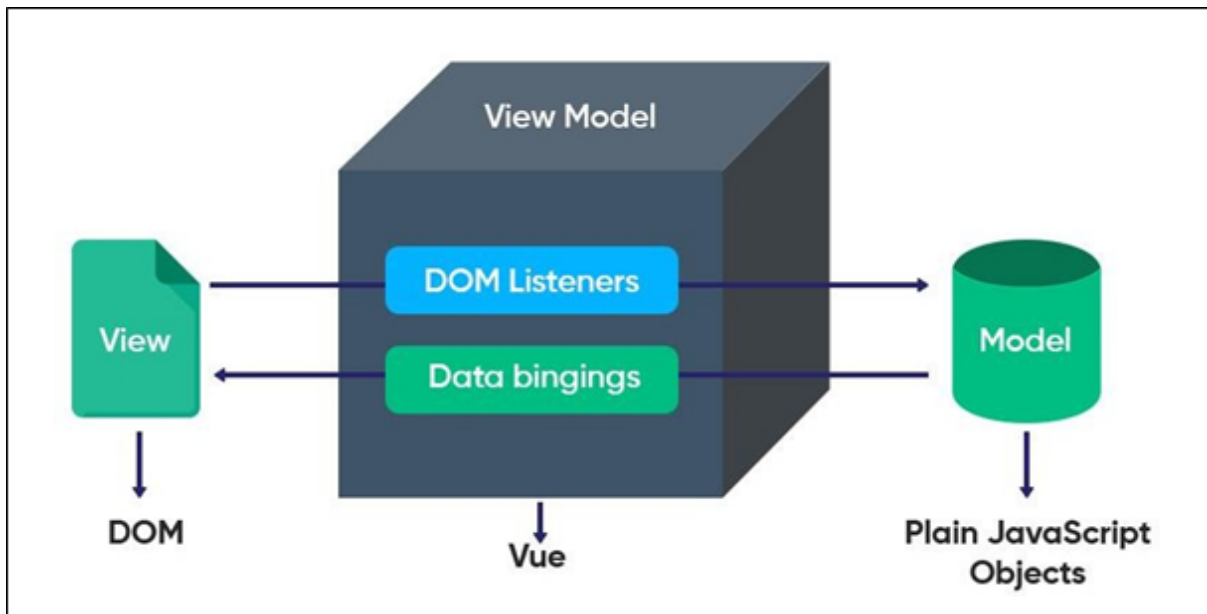


Figure 3.5: Vue Js architecture

- **Advantages:**

- Vue JS is lightweight in terms of framework size while preserving its flexibility and speed.
- Vue JS has a limited number of choices, which can be an advantage as it minimizes confusion during development.

- **Drawbacks:**

- Vue JS has a smaller community compared to Angular and React.
- There are limited resources available for Vue JS development.

### Comparative study and summary

The Figure 3.6 shows a comparative study between Angular JS, React JS, and Vue JS in 2022. [10]

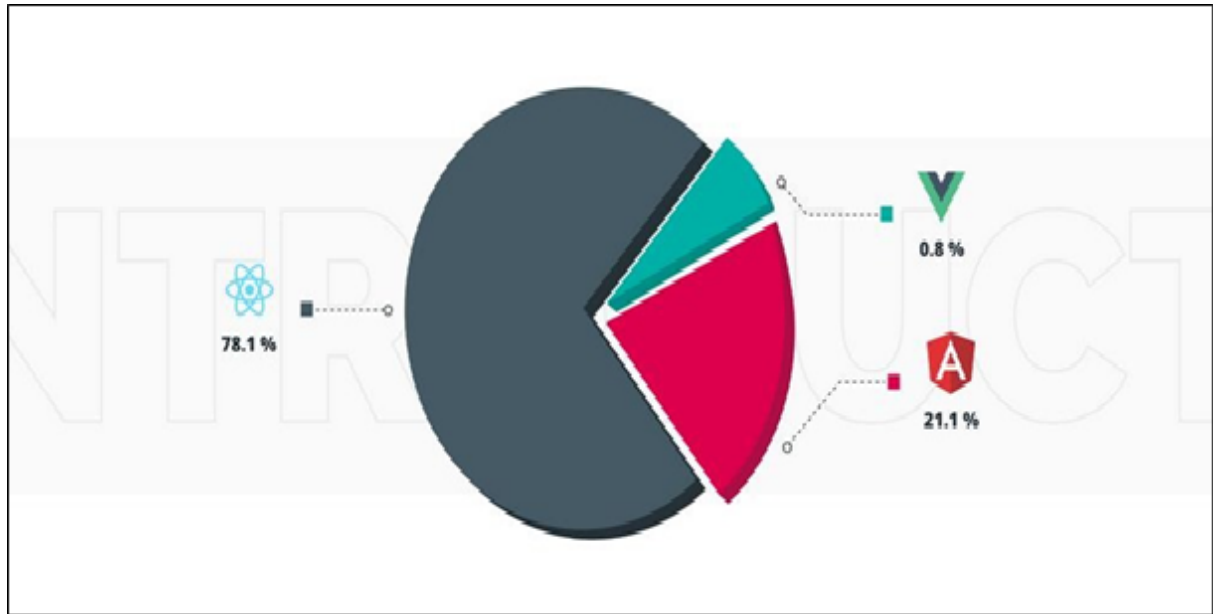


Figure 3.6: Comparative study between Angular JS, React JS, and Vue JS in 2022

After comparing these three major frameworks, we have decided to choose React JS. In recent times, React JS has become increasingly popular and has gained a significant presence on the web compared to other frameworks.

#### 3.2.2. Back-end

There are several frameworks based on Node js we can mention MentorJs, ExpressJs, Koajs and NestJS. Among those there's two popular frameworks which are NestJs and ExpressJs. And we chose NestJs because it's built in top of ExpressJs.

#### NestJs

Nest.js is a powerful framework for building efficient and scalable server-side applications using Node.js. It supports both TypeScript and JavaScript, and combines elements of Object-Oriented Programming, Functional Programming, and Functional Reactive Programming. Nest.js provides an abstraction layer over common Node.js frameworks like Express and Fastify, while also giving developers direct access to their APIs. This allows for the use of a wide range of third-party modules.

## NestJs Pros and architecture

The mains reasons behind the success of NestJs in the market of web Technologies :

- Built on Typescript
- Powerful Command Line Interface (CLI)
- Modularization
- Database agonistic
- Support various libraries

Nest.js is known for its modular architecture, where every application is composed of at least one module, known as the 'root' module. In a typical Nest.js application, each logical part of the app is contained within its own module, allowing for easy organization of components. This means that an application can have multiple modules, each with a specific set of related capabilities. The below figure[4] clarifies the NestJS architecture

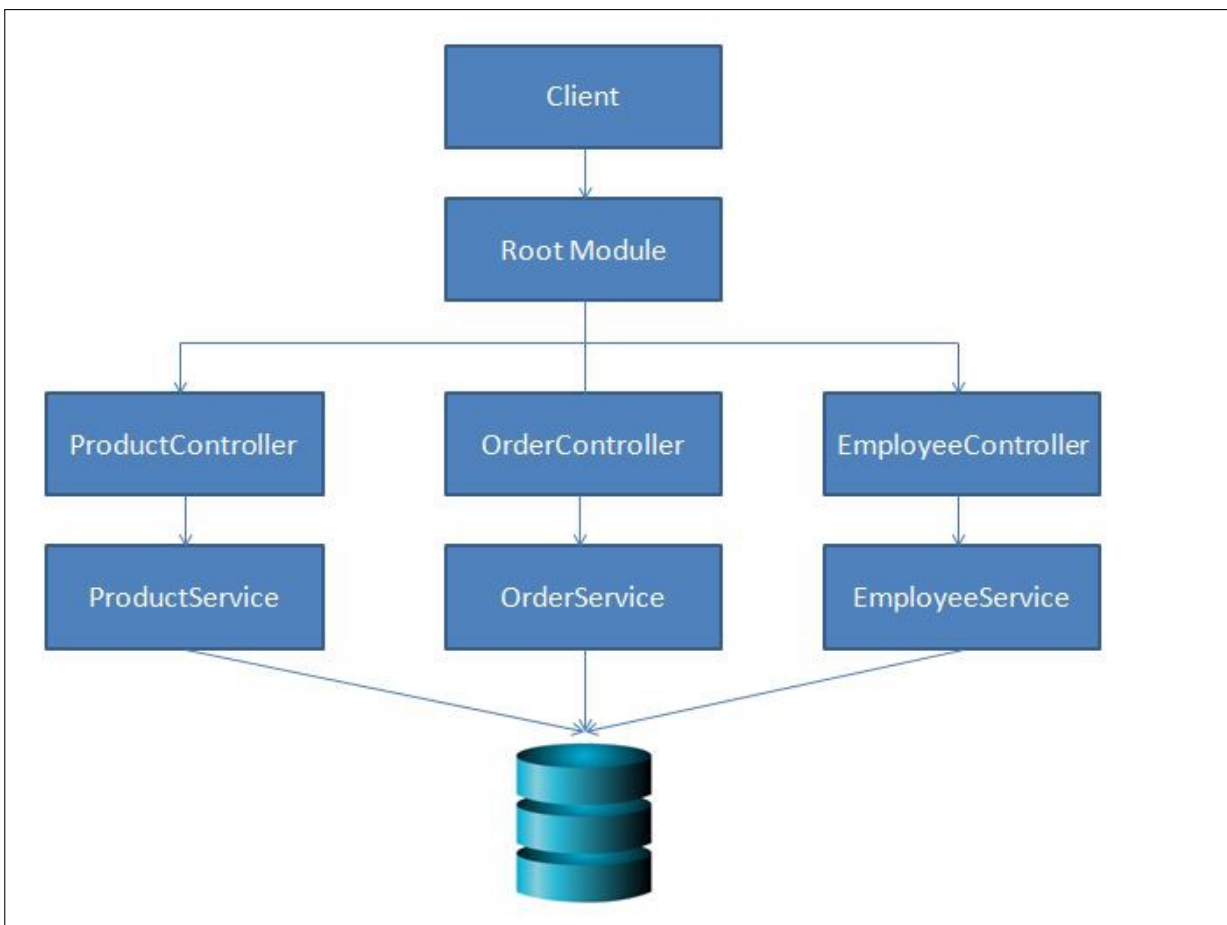


Figure 3.7: Architecture and workflow of NestJS

### 3.2.3. API'S

An API (Application Programming Interface) is a set of protocols and tools that allows different software components to communicate and share data. For example, a weather app on your phone might use an API provided by a weather service to access current weather data for your location. The API acts as an intermediary between the app and the weather service's database, providing a way for the app to access the data without having direct access to the database.



Figure 3.8: GraphQL vs REST

In fact as technology never stops advancing there's always alternative solution we mention the most powerful and widely-used API techniques GraphQL and REST .The table 3.1 shows a comparative study between GraphQL and REST.

Aspect	REST APIs	GraphQL APIs
Data Fetching	Fixed data structure, retrieves complete resources	Flexible data structure, retrieves specific data with queries
Over-fetching/Under-fetching	Common (retrieve more or less data than needed)	Rare (retrieve exactly what is specified in the query)
Multiple End-points	Multiple endpoints for different resources	Single endpoint for all data fetching
Network Efficiency	Potential for inefficient network requests due to fixed structure	Efficient network requests due to flexible queries
Caching	Caching at the endpoint level (HTTP caching)	Caching at the field level (fine-grained caching)
Tooling	Mature tooling and extensive ecosystem support	Growing tooling and ecosystem support
Learning Curve	Relatively straightforward to understand and implement	Requires understanding of GraphQL concepts and querying language
Documentation	Well-defined endpoints and documentation	Self-documented schema and strong typing

Table 3.1: Comparison between REST APIs and GraphQL APIs

## GraphQL

### A more efficient alternative to REST

1. increased internet usage creates needs for efficient data fetching
2. variety of front end frameworks(one Single endpoint)

To wrap up GraphQL was developed to cope with the need for more flexibility and efficiency in client server communication. So this flexibility is the first criteria when choosing technologies in an application that uses scraping.

### 3.2.4. Database

A database is a collection of information that is organized and stored in a way that allows for easy access and management. This information is managed using a database management system (DBMS), which provides tools for storing, retrieving, and manipulating the data in the database.



## Relational Database VS Non-Relational Database

When selecting a database, one important consideration is whether to use a relational (SQL) or non-relational (NoSQL) data structure. These two types of databases have different strengths and weaknesses, and the choice between them often depends on the specific needs of the application. A comparison between these two structures can help in making out decision.

Feature	SQL (Relational)	NoSQL (Non-relational)
Type	Relational	Non-relational
Data	Structured, stored in tables	Unstructured, stored in JSON files
Schema	Rigid, predefined	Flexible, not predefined
Scalability	Vertical (increase the load on a single server)	Horizontal (sharing or adding servers)
Transaction	ACID	CAP Theorem (Consistency, Availability, Partition tolerance)

Table 3.2: Comparison between SQL (Relational) and NoSQL (Non-relational) databases

The main advantage of SQL systems is their ability to store and manipulate highly structured data while respecting the four ACID criteria: Atomicity, Coherence, Isolation, and Durability.

On the other hand, NoSQL databases are better suited for web scraping because they can handle unstructured data more efficiently. They are designed to handle large volumes of data and can easily scale horizontally. This makes them a good choice for storing and processing data from web scraping(which is the case for our project).

## MongoDB

MongoDB is the most popular non-relational database, and there are several reasons behind its popularity.

1. Document-based: MongoDB stores data in flexible, JSON-like documents.
2. High performance: MongoDB provides high performance for read and write operations.
3. High availability: MongoDB's replication facility, called replica sets, provides automatic failover and data redundancy.

4. Horizontal scalability: MongoDB can be scaled horizontally through sharding.
5. Rich query language: MongoDB provides a rich query language that supports CRUD operations as well as data aggregation and text search.
6. First in the ranking of non-relational databases as shown below in the figure 3.8 [3]









Rank			DBMS	Database Model	Score		
May 2023	Apr 2023	May 2022			May 2023	Apr 2023	May 2022
1.	1.	1.	Oracle +	Relational, Multi-model 	1232.64	+4.36	-30.18
2.	2.	2.	MySQL +	Relational, Multi-model 	1172.46	+14.68	-29.64
3.	3.	3.	Microsoft SQL Server +	Relational, Multi-model 	920.09	+1.57	-21.11
4.	4.	4.	PostgreSQL +	Relational, Multi-model 	617.90	+9.49	+2.61
5.	5.	5.	MongoDB +	Document, Multi-model 	436.61	-5.29	-41.63
6.	6.	6.	Redis +	Key-value, Multi-model 	168.13	-5.42	-10.89
7.	7.	7.	IBM Db2	Relational, Multi-model 	143.02	-2.48	-17.31
8.	8.	8.	Elasticsearch	Search engine, Multi-model 	141.63	+0.56	-16.06
9.	9.	↑ 10.	SQLite +	Relational	133.86	-0.68	-0.87
10.	10.	↓ 9.	Microsoft Access	Relational	131.17	-0.20	-12.27

Figure 3.9: Top 10 databases

### 3.2.5. Scraping Tools

#### What's web scraping?

Web scraping, also known as web extraction or harvesting, is a technique used to extract data from the World Wide Web (WWW) and save it to a file system or database for later retrieval or analysis. It involves retrieving web resources through HTTP requests or web browsers and then extracting the desired information from the acquired data. [11]

In the table 3.3 we show the results of a comparative research between scraping tools based on JS.

Tool	Features	Description
Cheerio	Lightweight, jQuery-like syntax	A fast and efficient HTML parsing library. It provides a simple and familiar API for traversing and manipulating the DOM using a jQuery-like syntax.
Puppeteer	Full browser automation	A high-level Node.js library that provides control over a headless Chrome or Chromium browser. It allows you to automate interactions with web pages, including rendering, navigation, and data extraction.
Apify	Web scraping and automation platform	A comprehensive platform that offers tools and infrastructure for web scraping, data extraction, and automation. It provides features like automatic scaling, proxy rotation, and data storage.
Node.js	Runtime environment for web scraping	A server-side JavaScript runtime environment that enables running JavaScript code outside of a web browser. It can be used for building web scrapers and running JavaScript-based scraping scripts.
Nightmare	High-level browser automation	A high-level browser automation library that uses Electron under the hood. It provides a simplified API for browser interactions, including navigation, form filling, and screenshot capturing.

Table 3.3: Comparison of JavaScript Web Scraping Tools

One of the popular web scraping tools mentioned earlier, Puppeteer, offers a compelling reason for its use. It provides full browser automation, allowing you to control a headless

Chrome or Chromium browser. This means you can interact with web pages just like a real user, navigating, filling forms, clicking buttons, and extracting data. Puppeteer's ability to handle JavaScript rendering, its JavaScript integration, and its support for headless operation make it a powerful choice for web scraping tasks involving dynamic content and browser automation.

### 3.3. Architectural Model

### 3.4. Software Environment

Software	Description	Utilization
<b>Design Environment</b>		
Visual Paradigm UML	Visual Paradigm is a design tool. It supports UML diagrams.	Creates UML diagrams (use case diagram, sequence diagram, class diagram...)
Photoshop	Photoshop is an editor for graphics and photo processing.	Design of logos, images, and figures.
Figma	Wireframing and UI designing software.	Build and design prototypes.
Diagrams.net	Diagrams.net is a free and open-source cross-platform graph drawing software.	Design the database diagrams.
<b>Development Environment</b>		
Visual Studio Code	A free open-source IDE for programming.	Develop the front-end part using React JS.and backend using Nestjs
<b>Testing Environment</b>		
Graphql Playground	GraphQL Playground is an in-browser IDE for exploring GraphQL APIs. It provides a graphical interface for developers to interact with a GraphQL server .	Test APIs.
<b>Report Editor</b>		
Overleaf(LaTeX)	Overleaf is an online LaTeX editor that allows users to create and edit LaTeX documents in their web browser.	Drafting the report.

Table 3.4: Software Tools

## Conclusion

In this chapter, we discussed the frameworks and tools used for development and design. We also outlined how data will be stored within the application. The following chapter will focus on implementing the first version of our project.

## 4 | Release 1:Administrator

Thanks to the `cleveref` package, there is no need to use `\eqref`. Remember that Equations have to be numbered only if they are referenced in the text.

Equations (4.1), (4.2), (4.3), and (4.4) show again Maxwell's equations without brace:

$$\nabla \cdot \mathbf{D} = \rho, \quad (4.1)$$

$$\nabla \times \mathbf{E} + \frac{\partial \mathbf{B}}{\partial t} = \mathbf{0}, \quad (4.2)$$

$$\nabla \cdot \mathbf{B} = 0, \quad (4.3)$$

$$\nabla \times \mathbf{H} - \frac{\partial \mathbf{D}}{\partial t} = \mathbf{J}. \quad (4.4)$$

Equation (4.5) is the same as before, but with just one label:

$$\left\{ \begin{array}{l} \nabla \cdot \mathbf{D} = \rho, \\ \nabla \times \mathbf{E} + \frac{\partial \mathbf{B}}{\partial t} = \mathbf{0}, \\ \nabla \cdot \mathbf{B} = 0, \\ \nabla \times \mathbf{H} - \frac{\partial \mathbf{D}}{\partial t} = \mathbf{J}. \end{array} \right. \quad (4.5)$$

## 4.1. Figures, Tables and Algorithms

Figures, Tables and Algorithms have to contain a Caption that describe their content, and have to be properly referred in the text.

### 4.1.1. Figures

For including pictures in your text you can use `TikZ` for high-quality hand-made figures, or just include them as usual with the command

```
\includegraphics[options]{filename.xxx}
```

Here xxx is the correct format, e.g. `.png`, `.jpg`, `.eps`, ....





Figure 4.1: Caption of the Figure to appear in the List of Figures.

Thanks to the `\subfloat` command, a single figure, such as Figure 4.1, can contain multiple sub-figures with their own caption and label, e.g. Figure 4.2a and Figure 4.2b.



Figure 4.2: This is a very long caption you don't want to appear in the List of Figures.

### 4.1.2. Tables

Within the environments `table` and `tabular` you can create very fancy tables as the one shown in Table 4.1.

Title of Table (optional)			
	column 1	column 2	column 3
row 1	1	2	3
row 2	$\alpha$	$\beta$	$\gamma$
row 3	alpha	beta	gamma

Table 4.1: Caption of the Table to appear in the List of Tables.

You can also consider to highlight selected columns or rows in order to make tables more

readable. Moreover, with the use of `table*` and the option `bp` it is possible to align them at the bottom of the page. One example is presented in Table 4.2.

	column1	column2	column3	column4	column5	column6
row1	1	2	3	4	5	6
row2	a	b	c	d	e	f
row3	$\alpha$	$\beta$	$\gamma$	$\delta$	$\phi$	$\omega$
row4	alpha	beta	gamma	delta	phi	omega

Table 4.2: Highlighting the columns

	column1	column2	column3	column4	column5	column6
row1	1	2	3	4	5	6
row2	a	b	c	d	e	f
row3	$\alpha$	$\beta$	$\gamma$	$\delta$	$\phi$	$\omega$
row4	alpha	beta	gamma	delta	phi	omega

Table 4.3: Highlighting the rows

### 4.1.3. Algorithms

Pseudo-algorithms can be written in  $\text{\LaTeX}$  with the `algorithm` and `algorithmic` packages. An example is shown in Algorithm 4.1.

---

#### Algorithm 4.1 Name of the Algorithm

---

```

1: Initial instructions
2: for for – condition do
3:   Some instructions
4:   if if – condition then
5:     Some other instructions
6:   end if
7: end for
8: while while – condition do
9:   Some further instructions
10: end while
11: Final instructions

```

---

## 4.2. Theorems, propositions and lists

### 4.2.1. Theorems

Theorems have to be formatted as:

**Theorem 4.1.** *Write here your theorem.*

*Proof.* If useful you can report here the proof.

### 4.2.2. Propositions

Propositions have to be formatted as:

**Proposition 4.1.** *Write here your proposition.*

### 4.2.3. Lists

How to insert itemized lists:

- first item;
- second item.

How to insert numbered lists:

1. first item;
2. second item.

## 4.3. Use of copyrighted material

Each student is responsible for obtaining copyright permissions, if necessary, to include published material in the thesis. This applies typically to third-party material published by someone else.

## 4.4. Plagiarism

You have to be sure to respect the rules on Copyright and avoid an involuntary plagiarism. It is allowed to take other persons' ideas only if the author and his original work are clearly mentioned. As stated in the Code of Ethics and Conduct, Politecnico di Milano *promotes the integrity of research, condemns manipulation and the infringement of*

*intellectual property*, and gives opportunity to all those who carry out research activities to have an adequate training on ethical conduct and integrity while doing research. To be sure to respect the copyright rules, read the guides on Copyright legislation and citation styles available at:

<https://www.biblio.polimi.it/en/tools/courses-and-tutorials>

You can also attend the courses which are periodically organized on "Bibliographic citations and bibliography management".

## 4.5. Bibliography and citations

Your thesis must contain a suitable Bibliography which lists all the sources consulted on developing the work. The list of references is placed at the end of the manuscript after the chapter containing the conclusions. We suggest to use the BibTeX package and save the bibliographic references in the file `Thesis_bibliography.bib`. This is indeed a database containing all the information about the references. To cite in your manuscript, use the `\cite{}` command as follows:

*Here is how you cite bibliography entries: [5], or multiple ones at once: [6, 8].*

The bibliography and list of references are generated automatically by running BibTeX [2].

## 5 | Conclusions and future developments

A final chapter containing the main conclusions of your research/study and possible future developments of your work have to be inserted in this chapter.



# Bibliography

- [1] agile alliance.
- [2] CTAN. BiBTeX documentation, 2017. URL <https://ctan.org/topic/bibtex-doc>.
- [3] db engines.
- [4] geekflare.
- [5] D. E. Knuth. Computer programming as an art. *Commun. ACM*, pages 667–673, 1974.
- [6] D. E. Knuth. Two notes on notation. *Amer. Math. Monthly*, 99:403–422, 1992.
- [7] S. Kottwitz. *LaTeX Cookbook*. Packt Publishing Ltd, 2015.
- [8] L. Lamport. *LaTeX: A Document Preparation System*. Pearson Education India, 1994.
- [9] T. Oetiker, H. Partl, I. Hyna, and E. Schlegl. The not so short introduction to latex2 $\epsilon$ . *Electronic document available at <http://www.tex.ac.uk/tex-archive/info/lshort>*, 1995.
- [10] unknown. React vs angular vs vue.js — what is the best choice in 2022?
- [11] B. Zhao. Web scraping. *Encyclopedia of Big Data*, 2017. doi: 10.1007/978-3-319-32001-4\_483-1.





# A | Appendix A

If you need to include an appendix to support the research in your thesis, you can place it at the end of the manuscript. An appendix contains supplementary material (figures, tables, data, codes, mathematical proofs, surveys, . . . ) which supplement the main results contained in the previous chapters.



## B | Appendix B

It may be necessary to include another appendix to better organize the presentation of supplementary material.



## List of Figures

1.1	Wimobi logo . . . . .	4
1.2	standing of the websites of promotions in tunisia . . . . .	5
1.3	home page of trend.tn . . . . .	6
1.4	home page of 9adhity.com . . . . .	7
1.5	home page of lescatalogues.tn . . . . .	8
1.6	scrum framework process . . . . .	12
1.7	XP process . . . . .	14
2.1	general use case . . . . .	20
2.2	Detailed use case diagram for the actor "User" . . . . .	21
2.3	Detailed use case diagram for the actor "Administrator" . . . . .	22
2.4	Sequence Diagram "Authenticate" . . . . .	23
2.5	Sequence Diagrams "Add Promotion" . . . . .	25
3.1	Most popular programming languages . . . . .	32
3.2	Most popular Front-End technologies . . . . .	33
3.3	React JS architecture . . . . .	34
3.4	Angular architecture . . . . .	35
3.5	Vue Js architecture . . . . .	36
3.6	Comparative study between Angular JS, React JS, and Vue JS in 2022 . .	37
3.7	Architecture and workflow of NestJS . . . . .	38
3.8	GraphQL vs REST . . . . .	39
3.9	Top 10 databases . . . . .	42
4.1	Caption of the Figure to appear in the List of Figures. . . . .	49
4.2	Shorter caption . . . . .	49



## List of Tables

1.1	information about the Wimobi. . . . .	4
1.2	Software Development Methodologies . . . . .	15
2.1	Authenticate Scenario . . . . .	24
2.2	Add Promotions Scenario . . . . .	26
2.3	Project Team . . . . .	27
2.5	Planning of Releases and Sprints . . . . .	30
3.1	Comparison between REST APIs and GraphQL APIs . . . . .	40
3.2	Comparison between SQL (Relational) and NoSQL (Non-relational) databases	41
3.3	Comparison of JavaScript Web Scraping Tools . . . . .	43
3.4	Software Tools . . . . .	45
4.1	Caption of the Table to appear in the List of Tables. . . . .	49
4.2	Highlighting the columns . . . . .	50
4.3	Highlighting the rows . . . . .	50





# List of Symbols

Variable	Description	SI unit
$\boldsymbol{u}$	solid displacement	m
$\boldsymbol{u}_f$	fluid displacement	m



# Acknowledgements

Here you might want to acknowledge someone.

