



HONORIS UNITED UNIVERSITIES






2022-2023

AURORA

SPECIALTY : Software Engineering

Conception and development of a camping website

Realized by: Bit BY Bit

-  Ben Nasr Ines
-  Dhaou Jawhar
-  Hammouda Hanen
-  Kharroubi Hazem
-  Yacoubi Islem

Thanks

Before proceeding with the presentation of our work, we would like to express our gratitude to the people who, from near and far, helped us to carry out this project.

In particular, we would like to thank all the readers who carefully read our report and respected our effort.

We express our deep gratitude and thanks to our ESPRIT supervisors for the time they devoted to supervising and monitoring this work.

Finally, we would like to thank the members of the jury for the honor of judging this work.

Table of contents

General Introduction	6
Chapter 1: General Framework of The Project	7
General Context of the Project	8
1.1 Introduction:.....	9
1.2 Problematic:.....	9
1.3 Study and Criticism of the existing:	9
1.4 Proposed solution:	10
1.5 Added Value :	10
1.6 Methodology Adopted :.....	10
1.7 Planning (Gantt Diagram):.....	11
1.8 Conclusion :.....	12
Chapter 2: Needs Analysis.....	13
2.1 Introduction:.....	14
2.2 Identification of Actors :.....	14
2.3 Functional Requirements (by users) :	14
2.4 Non-Functional Requirements :	16
2.5 Identifying Use Cases:.....	16
2.5.1 Global Use Case Diagram:	16
2.5.2 Global Class Diagram :	18
2.5.3 System Sequence Diagram:	19
2.6 Conclusion:	20
Chapter 3: Conception.....	21
3.1 Introduction:.....	22
3.2 Physical Architecture:.....	22
3.3 Logical Architecture:	23
3.5 Deployment Diagram:	25
3.6 Conclusion:	25
Chapter 4: Realization.....	26
4.1 Introduction:.....	27
4.2 Working environment:.....	27
4.2.1 Software environment:	27
4.2.2 Hardware environment:	30

4.3.MOCK- UPS:	31
.....	33
4.4 Conclusion:	33
General Conclusion:	34

Table of figures

Figure 1:Task Board.....	11
Figure 2:Gantt Chart.....	12
Figure 3:Global use case diagram	17
Figure 4:Global Diagram of classes	18
Figure 5:Sequence Diagram	19
Figure 6:Physical Architecture	22
Figure 7:Logical Architecture	23
Figure 8:Packaging Diagram.....	24
Figure 9:DeploymentDiagram.....	25
Figure 10:Logo of SpringBoot	27
Figure 11:Logo of IntelliJ	28
Figure 12:Logo of MySQL.....	28
Figure 13:Logo of Swagger.....	28
Figure 14: Logoof Angular.....	29
Figure 15: Logoof Visual Studio Code	29
Figure 16:Logo of GitHub.....	30
Figure 17:Authentication Interface.....	31
Figure 19: Creatingan account.....	32
Figure 18:Home Page.....	32
Figure 20: center interface.....	33
Figure 21: Shop Interface	33

General Introduction

The great outdoors has always been a popular destination for individuals seeking adventure and a break from the monotony of daily life. Camping, in particular, has been a favored activity for people of all ages. It provides an opportunity to connect with nature, disconnect from technology, and spend quality time with family and friends.

Lately, camping has become increasingly popular, with more and more people seeking to escape the hustle and bustle of city life and immerse themselves in the beauty of the natural world.

This trend has led to a growing demand for camping facilities, such as campgrounds, RV parks, and other outdoor recreation areas.

In recent years, the rise of technology has had a profound impact on the way people plan and organize their camping trips.

In fact, the internet and mobile devices have made it easier to research camping destinations, book sites, and purchase equipment.

However, despite these advances, the camping experience remains fragmented, with users having to use multiple websites and apps to plan and book their trips.

This can lead to confusion and frustration, especially for those who are new to camping or lack technical know-how.

This report aims to examine the current state of the art in camping-focused applications and explore the opportunities for innovation and growth in this field. It will provide an overview of the general framework of the project and the key components that are necessary for success in this competitive market.

Chapter 1: General Framework of The Project

General Context of the Project

This camping management application allows users to find and book camping centers that meet their specific needs.

They can search for camping centers based on location, available amenities, and offered activities.

The application also includes an online store where users can purchase or rent camping gear if they don't have their own.

The application provides tools for camping center managers to promote their centers by adding events and activities, and updating information about their center.

Campers can provide feedback about their camping experiences through reviews and comments, and can communicate with other campers and camping center managers through a built-in chat feature.

The application also has a complaint management system to help resolve any issues that may arise during a camping trip.

Admins have full control over the application and its users, including managing user accounts and application settings.

The goal of the application is to make camping easier and more accessible for everyone by providing a comprehensive solution for all their camping needs.

1.1 Introduction:

In this chapter, we aim to provide a comprehensive overview of the project and its context. This includes a thorough analysis of the current state of the problem, along with a comprehensive evaluation of the proposed solution.

Subsequently, we will delve into the design aspects of the solution and present the logical and physical architecture that has been developed to address the limitations of the current state of the problem.

1.2 Problematic:

The general problem that a camping management application aims to solve is the difficulty and inefficiency in finding and booking camping centers that meet specific needs and preferences, as well as the hassle of coordinating gear rental or purchases for the camping trip.

1.3 Study and Criticism of the existing:

Website	Advantages	Disadvantages
Campsitefinderonline	-Helps users find and book camping sites	-Does not provide comprehensive information about each site. -Limited site options. -Does not offer gear rental services. -Critics note the app has these limitations.
Camping App	- Provides users with information about camping sites including reviews and photos	- Does not have a gear rental service. -Information is not always up to date.
Tentrr	- Connects users with private landowners for camping sites	- Limited selection of camping sites. -Prices can be higher than traditional camping sites.
Hipcamp	-The app includes not only traditional camping options but also unique experiences like glamping and farm stays.	-Limited selection of sites. -Incomplete or inaccurate information about each site.

1.4 Proposed solution:

Developers of this application aim to make the process of finding and booking camping centers more efficient, by allowing users to search and compare centers based on specific criteria, and easily coordinate gear rental or purchases through the same platform. This helps to streamline the camping experience and make it more enjoyable for users. Additionally, the inclusion of features such as user feedback, communication with camping center managers, and event listings, further enhances the camping experience by providing users with a more complete and personalized experience.

In general, the goal of a camping management application is to make the process of planning and enjoying a camping trip easier and more efficient for users, by providing a centralized platform for booking and gear coordination.

1.5 Added Value :

- **Demo videos:** Offer demo videos for the most popular products to help customers better understand how they work and the benefits they offer.
- **Customization:** Offer the possibility to customize certain products, such as tents or jackets, with color, size, or material options.
- **Route Planner:** Provide users with information about travel routes and nearby tourist attractions, based on their camping destination.

1.6 Methodology Adopted :

The UP (Unified Process) methodology is a flexible, iterative and incremental approach to software development that is well-suited for the development of modern, complex applications, such as our app.

This methodology is designed to adapt to changing requirements and changing technology, making it a great choice for projects like ours.

One of the key benefits of the UP methodology is that it enables the development team to deliver high-quality, functional software in a timely and cost-effective manner. This is achieved by breaking down the development process into smaller, manageable phases and by incorporating regular feedback and testing at each stage.

For example, in the initial phase of the UP methodology, our team will focus on gathering requirements and defining the overall architecture of the application.

This stage is critical in ensuring that our app meets the needs of the user and that it can be developed efficiently.

In the next phase, our development team will focus on the design and implementation of the key functionality of the app.

Throughout the development process, regular testing and feedback will be integrated to ensure that the app is functional, reliable and of high quality.

In conclusion, the UP methodology offers a proven and effective approach to software development that is well-suited to the needs of modern, complex applications.

By incorporating regular testing and feedback, the UP methodology helps to ensure that our app will be functional, reliable and of high quality, providing us with the best possible outcome for our project.

1.7 Planning (Gantt Diagram):

Mode Tâche	Nom de la tâche	Durée	Début	Responsable	Fin	Prédécesseurs
	initialization phase	16 jours	Mar 24/01/23	All the team	Mar 14/02/23	
	collection of needs	5 jours	Mar 24/01/23	All the team	Lun 30/01/23	
	feasibility study	2 jours	Sam 28/01/23	All the team	Lun 30/01/23	
	Coaching	1 jour	Mar 31/01/23	Coachs	Mar 31/01/23	5;6
	specification of requirements	6 jours	Mer 01/02/23	All the team	Mer 08/02/23	7
	create the specification book	3 jours	Jeu 09/02/23	All the team	Lun 13/02/23	8
	Validation of phase 0	1 jour	Mar 14/02/23	Coachs	Mar 14/02/23	9;8
	Back end development phase	20 jours	Mer 15/02/23	All the team	Mar 14/03/23	
	Design of the API	4 jours	Mer 15/02/23	All the team	Lun 20/02/23	10
	Feature implementation	11 jours	Mar 21/02/23	All the team	Mar 07/03/23	12
	Unit tests	2 jours	Mer 08/03/23	All the team	Jeu 09/03/23	13
	Integration test	2 jours	Ven 10/03/23	All the team	Lun 13/03/23	14
	Validation of phase 1	1 jour	Mar 14/03/23	All the team	Mar 14/03/23	15
	Front End development phase	40 jours	Mer 15/03/23	All the team	Mar 09/05/23	
	User interface design	10 jours	Mer 15/03/23	All the team	Mar 28/03/23	16
	Implementof the functionalities	20 jours	Mer 29/03/23	All the team	Mar 25/04/23	18
	Unit tests	6 jours	Mer 26/04/23	All the team	Mer 03/05/23	19
	Integration test	3 jours	Jeu 04/05/23	All the team	Lun 08/05/23	20
	Validation of phase 2	1 jour	Mar 09/05/23	Coachs	Mar 09/05/23	21

Figure 1: Task Board

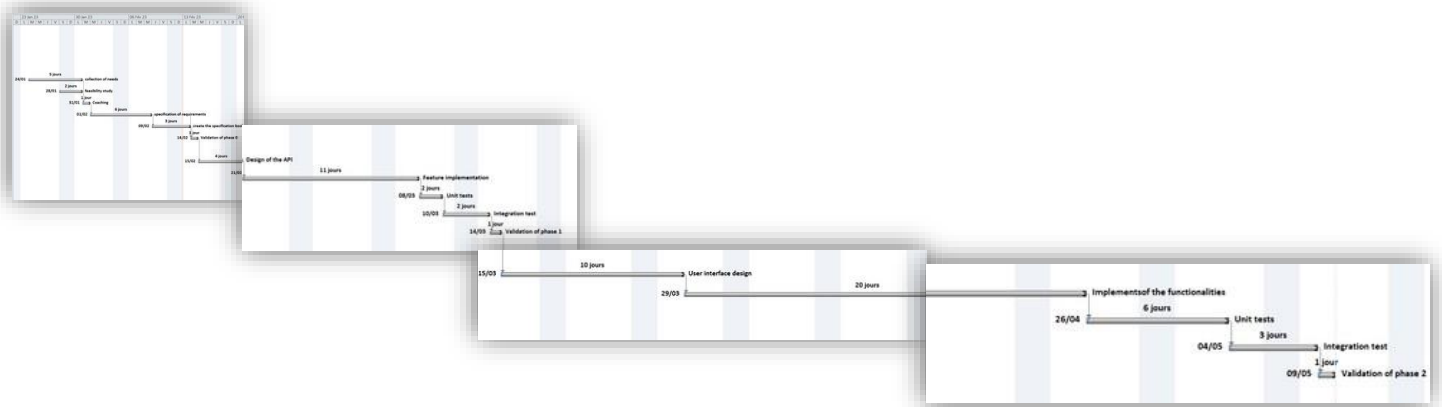


Figure 2:Gantt Chart

1.8 Conclusion :

In this initial section of the report, we first introduced the overarching concept behind the project.

Subsequently, we conducted a thorough analysis of existing solutions, highlighting their limitations. Our proposed solution was then presented.

In the next chapter, we will identify and specify the needs of our systems.

Chapter 2: Needs Analysis

2.1 Introduction:

The success of any project depends on the quality of its start. Therefore, the step of specifying the needs constitutes the starting point of our work, it must describe without ambiguity the application to be developed. To ensure the expected objectives, it is essential that we achieve a clear view of the different expected needs of our project.

During this chapter, we will identify the functionalities that will be put into service for the different actors of our application by defining the different use cases and some scenarios that explain these cases.

2.2 Identification of Actors :

- Admin
- Camp Manager
- Shop Manager
- Customer

2.3 Functional Requirements (by users) :

Web Site management:

- Admin should be able to view all website details through the dashboard.
- Admin should be able to manage events

Activity management:

- Users must be able to consult the activities offered by the campsite (hiking, fishing, etc.) register for activities, and consult their registration history.

Reservation management:

- Users should be able to view their booking details (dates, location, etc.), pay online, and receive email confirmations and reminders.

Claims management:

Users can:

- Receive and inform the party responsible for the claim.
- consult list of claims.
- Add, delete, modify, and search for claims.
- Report problems or maintenance needs.

Service management:

- Users must be able to consult the services offered by the campsite(showers, toilets, etc.).

Review management:

- Users should be able to give feedback on the campsite and activities,review other users' reviews, and report inappropriate reviews.

Profile management:

- Users should be able to create a profile, view and update their personal information, and view their booking history.

Statistics management:

- Administrators and users could be able to view statistics on bookings,activities and reservations.

Account management:

- Users and administrators must be able to see their account information and they can modify some personal information (surname,first name, phone number, etc.), and delete or deactivate their accounts.

Shop management:

- Users be able to check the product list and add some products to thewishlist to receive a notification when the product is on sale.
- Users be able to buy products from the Shop
- Shop manager be able to create his store and manage his products.

Technical support:

- The app should provide technical support to users throughvarious channels, such as email, chat, and phone.

2.4 Non-Functional Requirements :

User-friendly interface:

- The app should have an intuitive and attractive interface that is easy to navigate.

Performance:

- The app should be fast and responsive, with minimal lag time when searching for camping sites or accessing information.
- Using weak coupling.

Security:

- The app should securely store user data and protect against unauthorized access with TOKEN.

Scalability:

- The app should be able to handle an increase in users and data without sacrificing performance or reliability.

2.5 Identifying Use Cases:

2.5.1 Global Use Case Diagram:

The use case diagram represents the structure of the functionalities required by the users of the system. It splits the functionality of the system into coherent units, use cases, making sense to the actors.

In this part we will present the different actors likely to interact with the system, and before that we give a little definition of the actor. An actor is the idealization of a role played by an external person, an external element interface that interacts with the system.

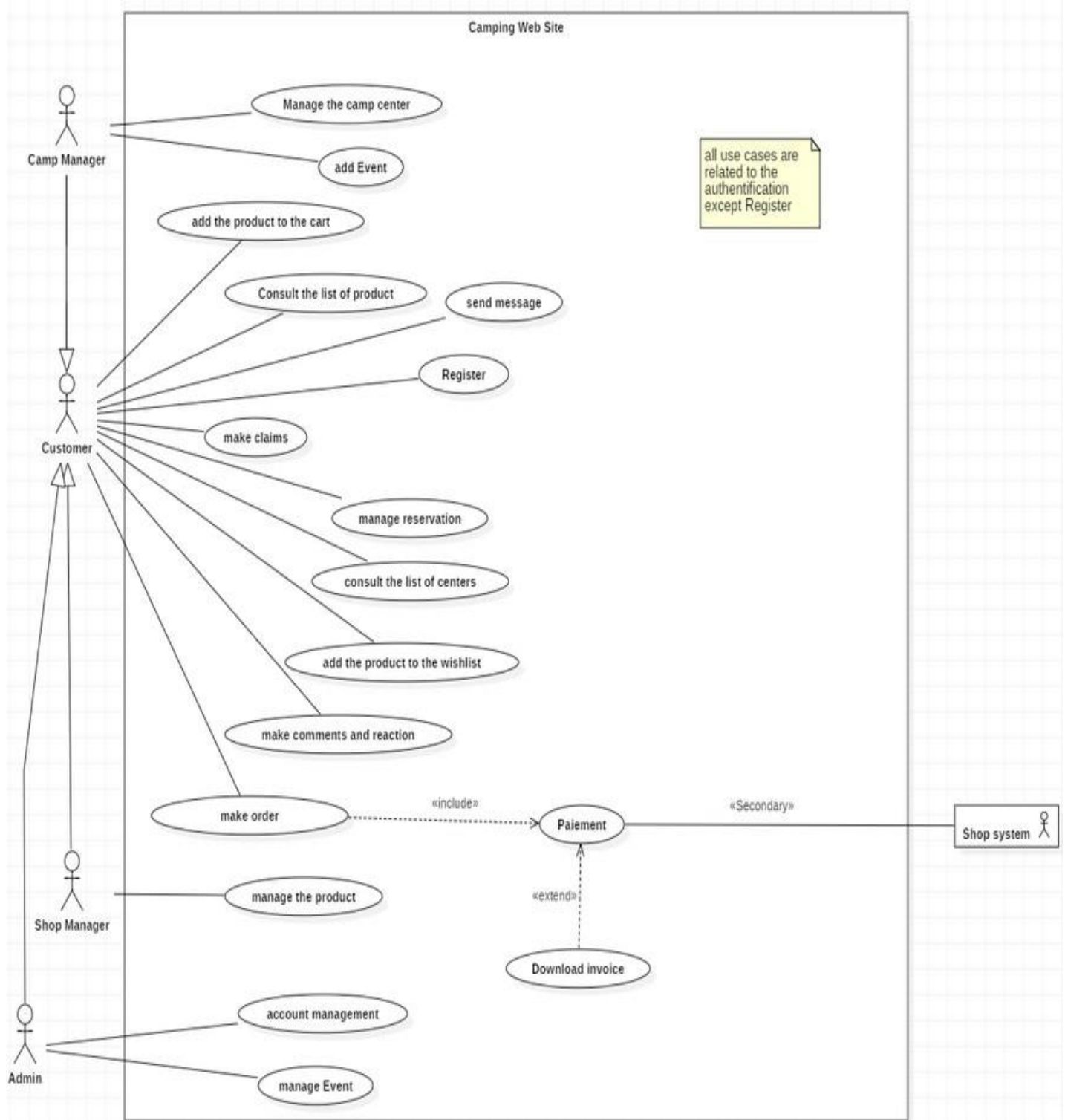


Figure 3:Global use case diagram

2.5.2 Global Class Diagram :

It is used to illustrate and create functional diagram of the system classes and serves as a system development resource within the software development life cycle.

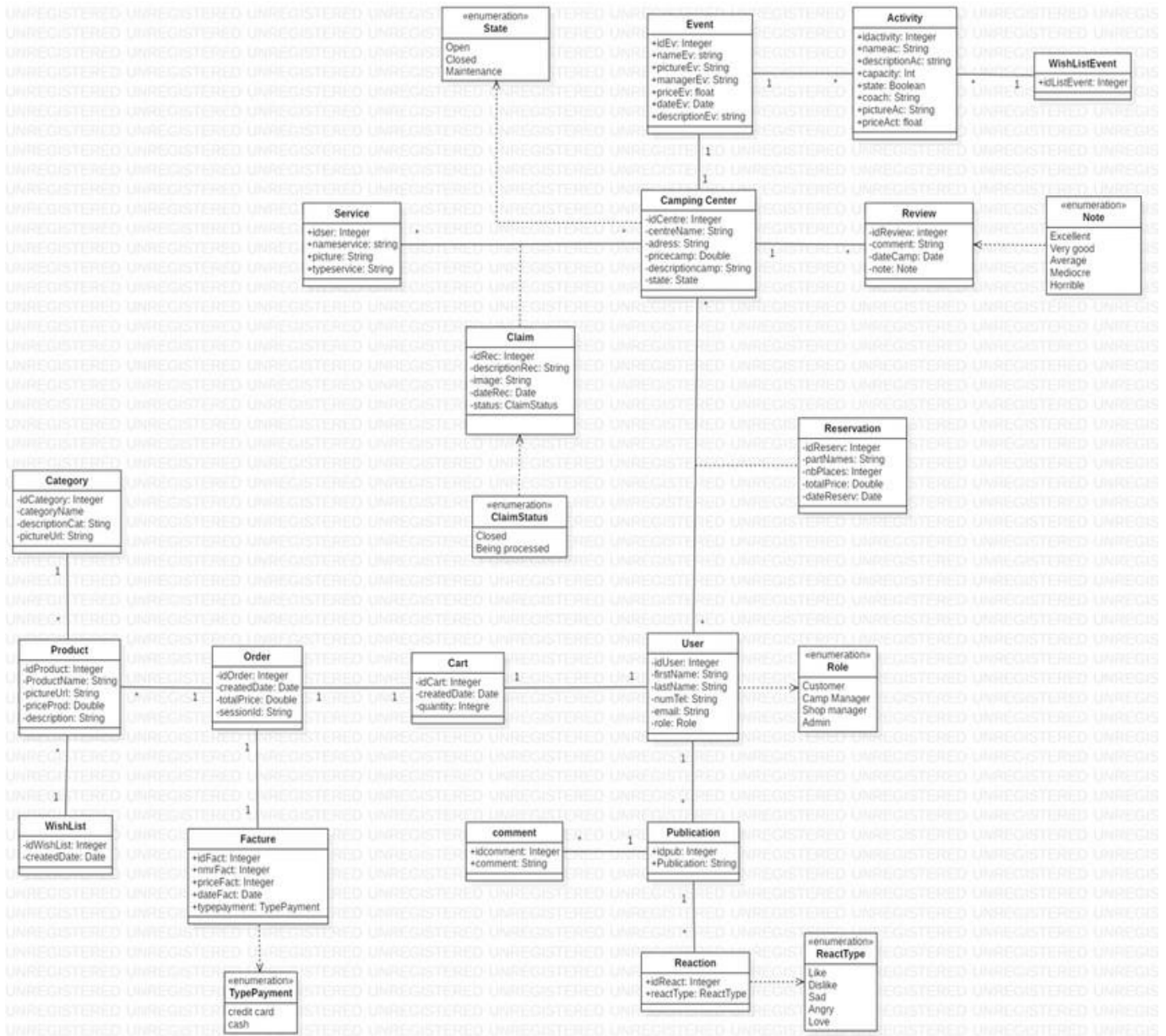


Figure 4:Global Diagram of classes

2.5.3 System Sequence Diagram:

An interaction diagram which describes how operations are carried out and in what order a group of objects work together.

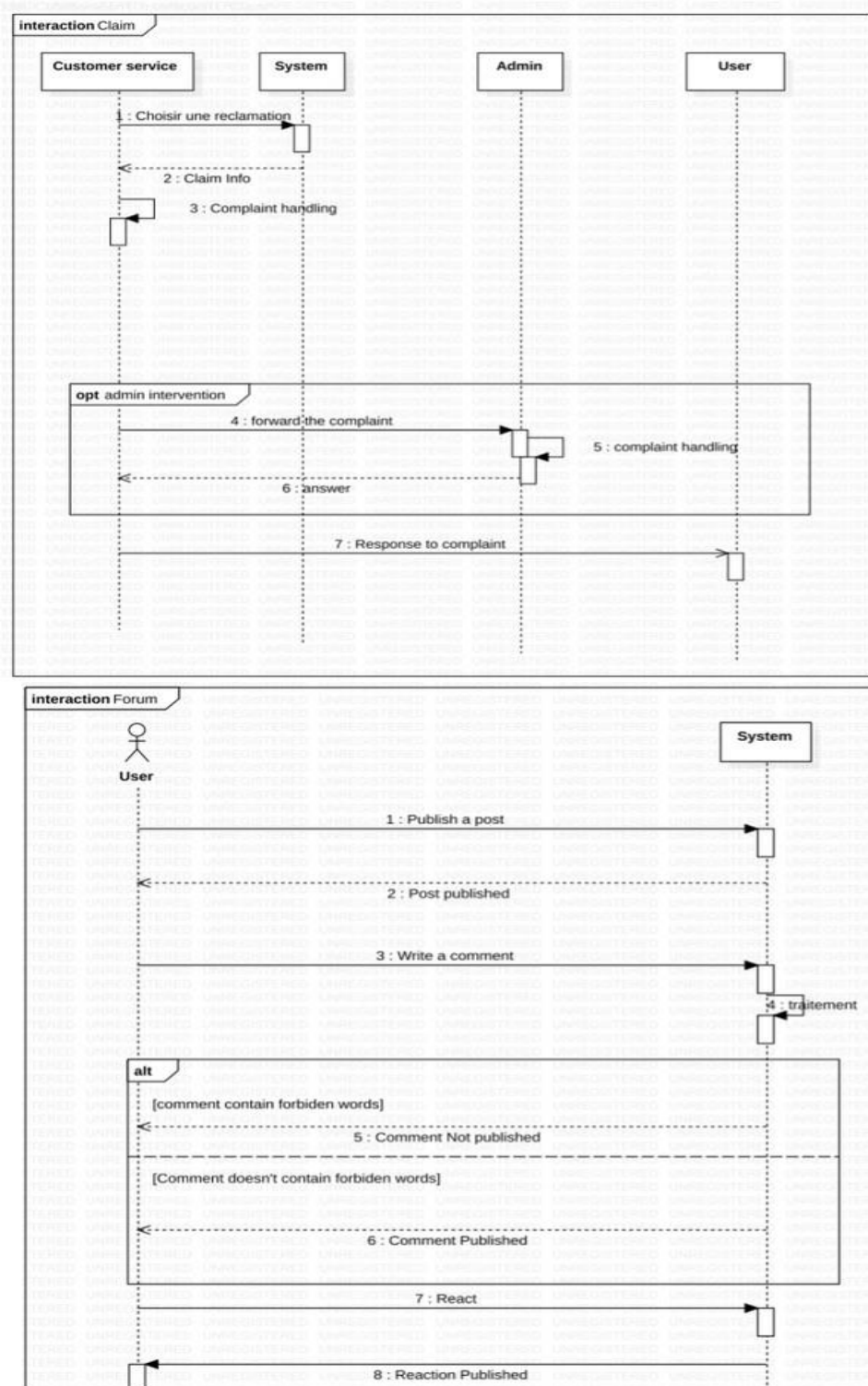


Figure 5:Sequence Diagram

2.6 Conclusion:

In this chapter, we have identified and specified the needs of our systems. In the following chapter, we will delve deeper into the design aspects, providing a comprehensive examination of both the physical and logical architecture.

Chapter 3: Conception

3.1 Introduction:

In this third chapter, we delve deeper into the technical aspect of the project. We begin by outlining the physical architecture of the system, including its hardware and software components. This is followed by a comprehensive examination of the logical architecture, which details the functionalities and relationships between the various system components. We then present the package diagram, which provides a visual representation of the system's structure and dependencies. Finally, we conclude with a discussion of the deployment diagram, which outlines the deployment of the solution and its components in the target environment.

3.2 Physical Architecture:

The chosen architecture consists of a client (web browser), a Tomcat server, a MySQL database, and a Node.js application with 4 tiers (client, Angular, embedded Apache server, database server). The communication protocol for secure web-based client-server communication is HTTPS (HTTP Secure), which uses SSL/TLS encryption to protect the data transmitted between the client and server.

HTTPS typically uses port 443 for communication,

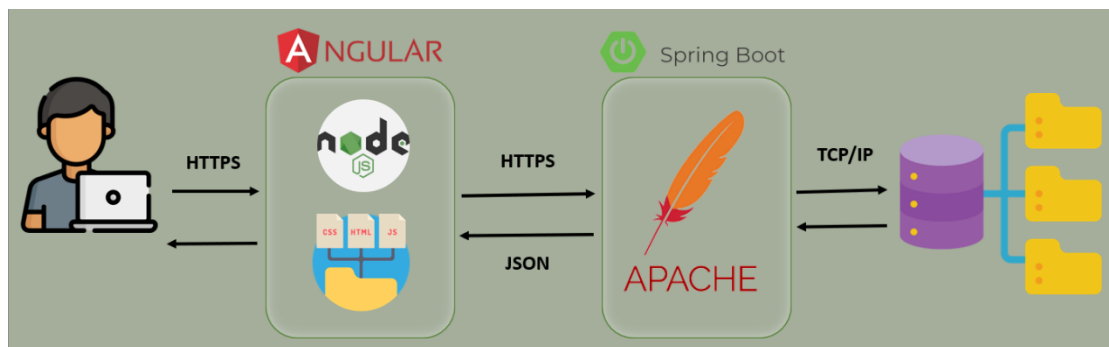


Figure 6: Physical Architecture

3.3 Logical Architecture:

MVVM stands for Model-View-ViewModel and is a software architecture pattern used to separate an application's user interface (UI) code from its business logic.

why we use MVVM:

Separation of concerns: MVVM separates the user interface (UI) from the application's business code, making it easier to maintain and upgrade the code.

Modularity: With MVVM, each part of the application has a clear and well-defined responsibility, making it easy to add new functionality or modify the UI without affecting the rest of the application.

Testability: MVVM facilitates unit testing by allowing separate testing of different layers of the application.

Code reuse: The ViewModel can be reused for different views, saving development time.

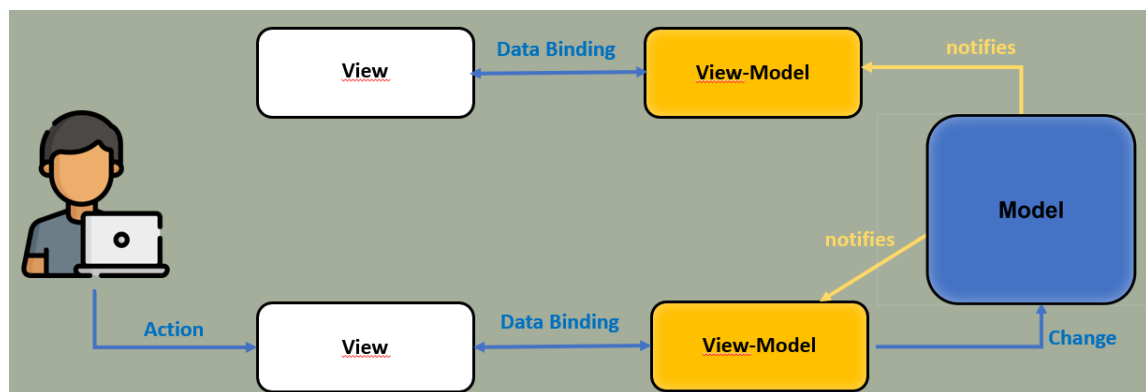


Figure 7: Logical Architecture

3.4 Packaging Diagram:

Package diagram is a structural diagram used to show the organization and arrangement of various model elements in the form of packages.

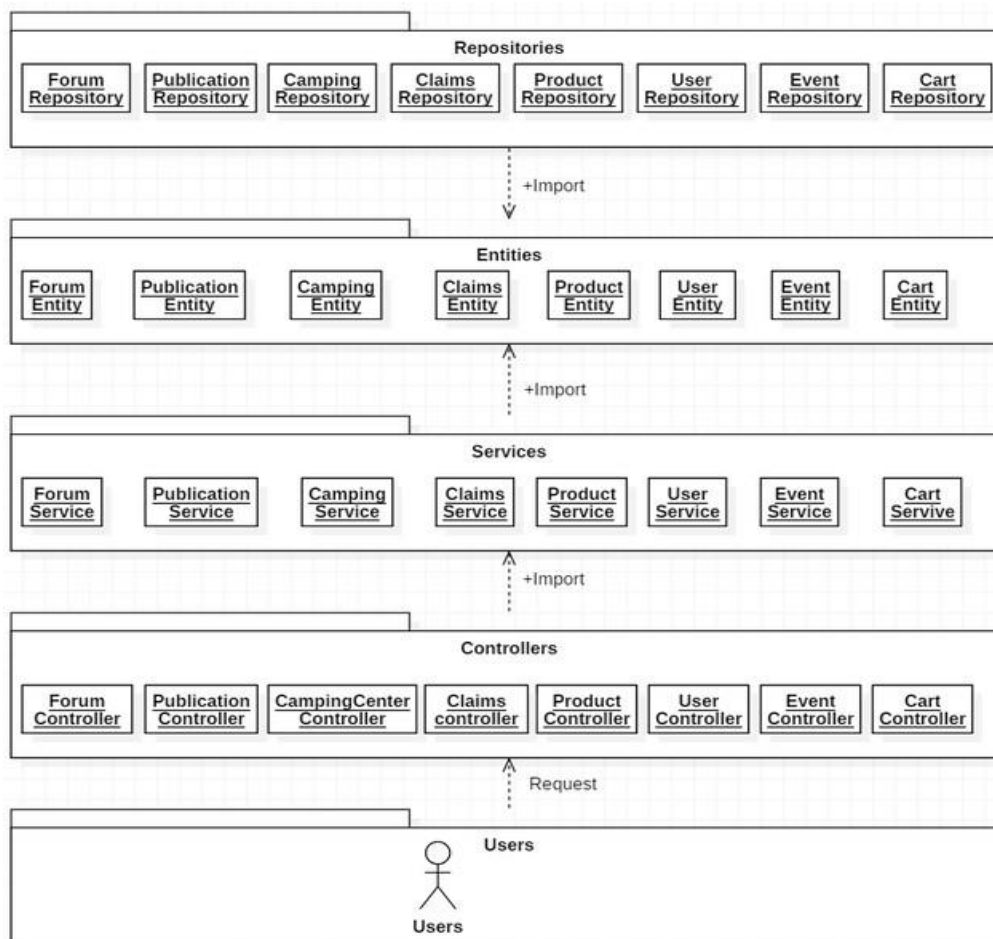


Figure 8:Packaging Diagram

3.5 Deployment Diagram:

A deployment diagram is a static view that is used to represent the use of the physical infrastructure by the system and how the components of the system are distributed and their relationships to each other.

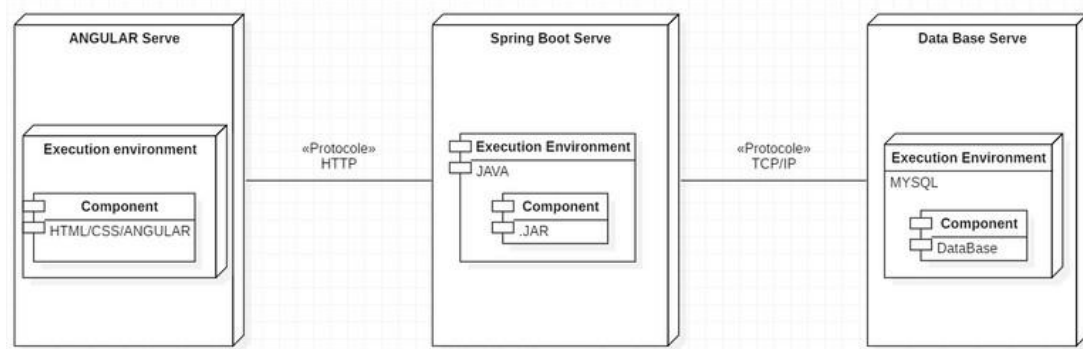


Figure 9:DeploymentDiagram

3.6 Conclusion:

In conclusion, the conception chapter was dedicated to understanding the needs and requirements for our system. Through thorough analysis and evaluation, we were able to identify the necessary components and features that will form the foundation of our design. With this foundation in place, we are now prepared to move forward to the realization phase of our system.

Chapter 4: Realization

4.1 Introduction:

This chapter focuses on the realization of the project, which includes the working environment, hardware environment, software environment and interfaces of the application. We will discuss the different components of the project and how they work together to create a successful outcome. We will also look at the different tools and technologies used to create the project and how they interact with each other.

Finally, we will explore the different interfaces of the application and how they can be used to create a user-friendly experience .

4.2 Working environment:

4.2.1 Software environment:

❖ Back end:

- **Spring Boot:**

To be able to design an application as desired, we chose to work with the Java Spring Boot framework. Spring Boot is a Framework particularly suited to development of micro-services in JAVA.

In addition to eliminating all the configuration work typical of Spring applications (Auto-configuration), it makes it easy to obtain perfectly autonomous micro-services. With Spring boot we will also get to know the Starters which allow us to start the development of our application.



Figure 10: Logo of SpringBoot

- **IntelliJ:**

is an integrated development environment (IDE) that is used for Java development, including the development of Spring Boot applications.

It provides a range of features and tools for software development, including code editing, debugging, and testing.



Figure 11:Logo of IntelliJ

- **MySQL :**

It is an open-source relational database management system.

It is widely used for web applications and provides robust features for data storage, retrieval, and management.

The database configured on Mysql can be connected with our Spring Boot java code thanks to the integration of the "spring.jpa.hibernate.ddl" configuration in the application. properties file located in the following path: src/ main/resources.



Figure 12:Logo of MySQL

- **Swagger:**

Swagger is an open-source framework used to design, build, document, and test RESTful APIs. It provides a set of tools and a standardized format to define APIs in a machine-readable way, which can be used to generate client libraries, documentation, and server stubs automatically. The latest version of Swagger is called OpenAPI, which is an industry standard for API specification.



Figure 13:Logo of Swagger

❖ Front end:

- **Angular:**

Angular is a popular open-source JavaScript-based framework for building

dynamic and interactive web applications. It is developed and maintained by Google and has a large community of developers and users.

Angular is used to build complex and scalable single-page applications, making it easier for developers to create dynamic and responsive user interfaces.

With its powerful features and tools, Angular enables developers to create high-quality, fast, and user-friendly applications that can run seamlessly on various platforms and devices.

Additionally, Angular has a rich set of libraries and tools that make it easier for developers to add functionality, test their applications, and maintain the codebase over time.



Figure 14: Logo of Angular

- **Visual Studio:**

It is an integrated development environment (IDE) for building applications on the Microsoft platform.

It provides a range of features and tools for software development, including code editing, debugging, and testing.



Figure 15: Logo of Visual Studio Code

- **Collaboration Tool:**

GitHub is a web-based platform for version control and collaboration that enables developers to store and manage their source code repositories.

It offers a range of features, including pull requests, code review, bug tracking, project management, and more.

With GitHub, developers can work together on the same codebase, streamline their workflow, and easily share their projects with others.

Additionally, the platform has a large community of developers and users, making it a great place for developers to network, find inspiration, and contribute to open-source projects.



Figure 16: Logo of GitHub

4.2.2 Hardware environment:

For this ongoing project, we will use different types of laptops where its average configuration is :

1-Device name: DESKTOP-TRI2SR4



Processor: 11th Gen Intel® Core™ i7-11800H @ 2.30GHz 2.30GHz Installed Ram

Memory: 32.0 GB

System Type: 64-bit operating system, x64 processor

2-Device name: DESKTOP-7ASQEG8



Processor: 11th Gen Intel® Core™ i7-1165G7 @ 2.80GHz 2.80 GHz Installed Ram

Memory: 16,0 GB

System Type: 64-bit operating system, x64 processor

3-Device name: LAPTOP-8I0AE9GM



Processor: Intel® Core™ i7-10750H CPU @ 2.60GHz 2.59 GHz Installed Ram
Memory: 16,0 GB
System Type: 64-bit operating system, x64 processor

4- Device name: DESKTOP-D82K9I9



Processor: Intel® Core™ i7-7700HQ CPU @ 2.80GHz 2.81 GHz Installed Ram
Memory: 16,0 GB
System Type: 64-bit operating system, x64 processor

5-Device name: DESKTOP-D82K9I9



Processor: Intel® Core™ i7-10750H CPU @ 2.60 GHz 2.59 GHz Installed Ram
Memory: 24,0 GB
System Type: 64-bit operating system, x64 processor

4.3.MOCK- UPS:

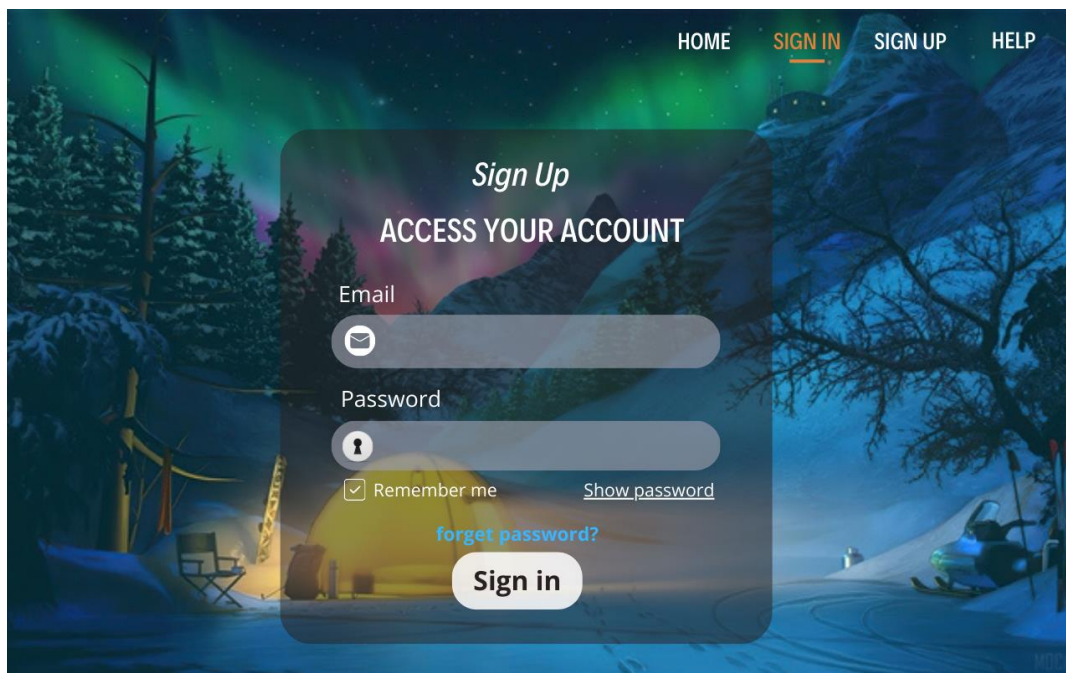
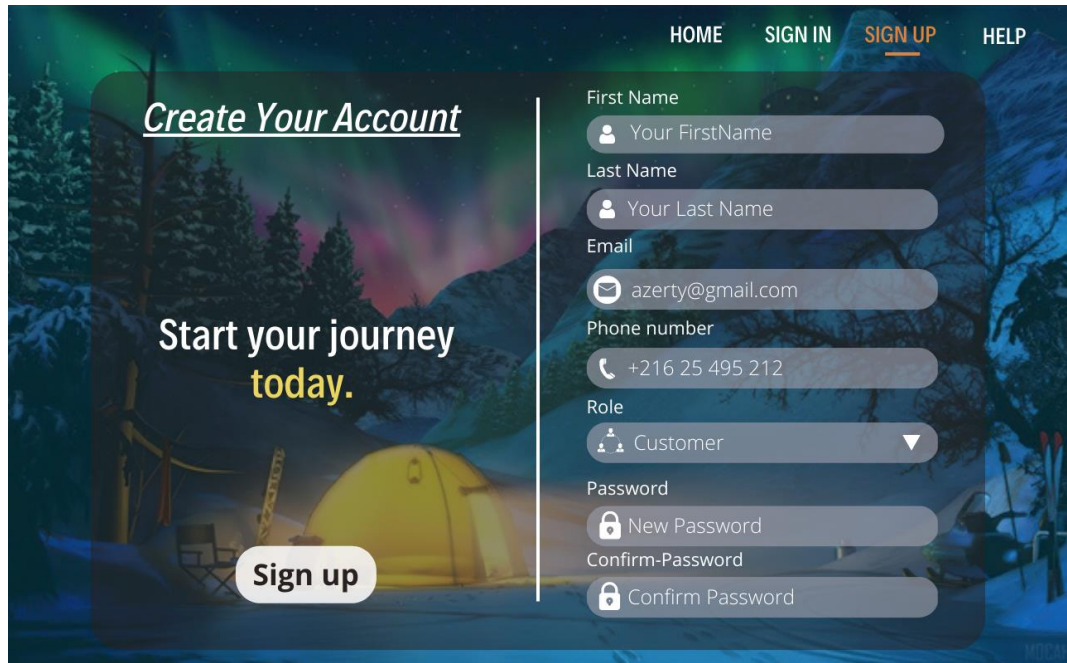


Figure 17:Authentication Interface



The image shows a 'Create Your Account' form on a website. The background is a dark, snowy forest with a yellow tent and a campfire. The form is on the right side of the page. It has a navigation bar at the top with 'HOME', 'SIGN IN', 'SIGN UP' (highlighted), and 'HELP'. The form fields are: First Name (placeholder: Your FirstName), Last Name (placeholder: Your Last Name), Email (placeholder: azerty@gmail.com), Phone number (placeholder: +216 25 495 212), Role (dropdown menu with 'Customer' selected), Password (placeholder: New Password), Confirm-Password (placeholder: Confirm Password), and a 'Sign up' button.

HOME SIGN IN **SIGN UP** HELP

Create Your Account

Start your journey today.

Sign up

First Name
Your FirstName

Last Name
Your Last Name

Email
azerty@gmail.com

Phone number
+216 25 495 212

Role
Customer ▼

Password
New Password

Confirm-Password
Confirm Password

Figure 19: Creating an account

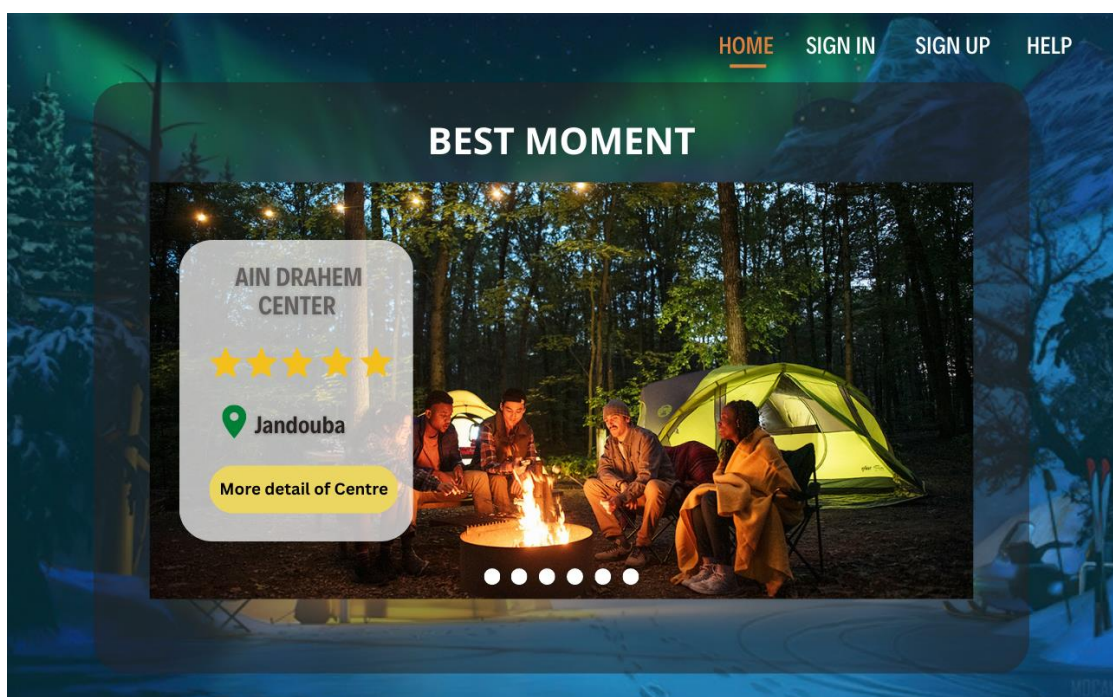


Figure 18: Home Page

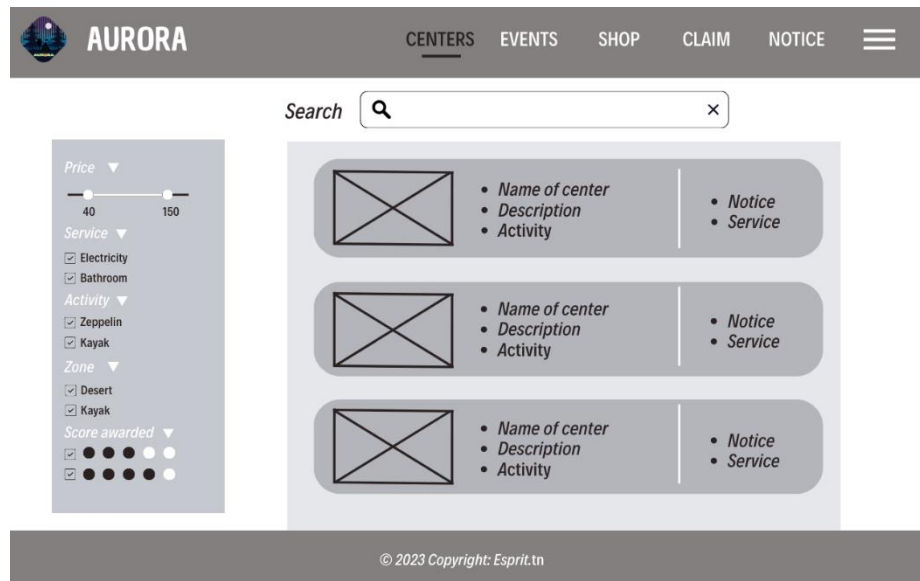


Figure 20: center interface

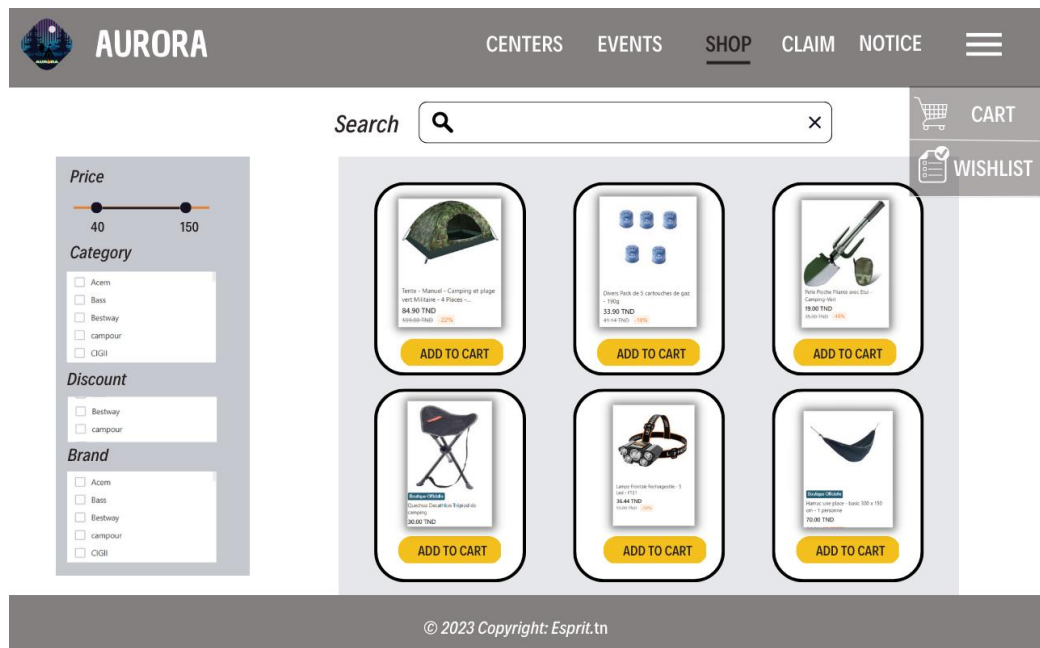


Figure 21: Shop Interface

4.4 Conclusion:

In conclusion, this chapter has outlined the technical aspects of the project or system, including the material environment, environment software, technologies, and scenario with mock-ups. These elements are important considerations in the development and implementation of the project. The information presented serves as a foundation for future steps and highlights the significance of technical considerations

in the project life cycle.

General Conclusion:

In conclusion, the field of technology has greatly impacted and transformed the way we live and interact with the world around us.

This includes the outdoor recreation industry and specifically, camping.

By incorporating technology into the camping experience, it has opened doors to new possibilities and enhanced the overall enjoyment for individuals and families.

The purpose of this report was to provide a comprehensive overview of the integration of technology into camping and the opportunities it has created for innovation and growth in the industry.

From a detailed study of the existing systems, to the proposed solution and its architecture, the report aimed to present a clear picture of how technology can be used to improve the camping experience.

We highlighted the various stages of the project, including the general context, the methodology adopted, and the conception, as well as the physical and logical architecture.

We also discussed the technological environment and the environment software that will be used during the project's implementation phase.

As the outdoor recreation industry continues to grow, it is important to stay on the cutting edge of technology and provide innovative solutions that cater to the demands and needs of campers.

This report serves as a foundation for future research and development in the field of technology and camping, and highlights the potential for continued growth and improvement.