Dedication

To our dear fathers,

No words can express the feelings I have towards you. It is thanks to your love, trust, patience, sacrifices, and encouragement throughout my journey that I have reached this stage in my life. I hope I have been worthy of your affection and trust.

To our dear mothers,

My reason for being, my reason for living, the lantern that illuminates my path and fills me with sweetness and love, words fail me to express all the gratitude, pride, and profound love I have for you for the sacrifices you have made for my success. May God preserve your good health and long life.

To our siblings

Thank you for your encouragement and advice. I wish you a life full of success, joy, and may God keep us united.

To all our friends,

Your friendship was great, your love and respect have been and will remain in my heart... You never hesitated to lend me a hand so that we could overcome mountains together...

To all those I love and who love me,

It is for you that I dedicate this work, hoping to have fulfilled your wishes to see me succeed.

Table of Contents

List of Figures :

List of tables :

GENERAL INTRODUCTION

The project aims to help users to choose their next destination by providing a comprehensive and reliable platform that helps them make informed decisions. Entertainment planning can be a daunting task, particularly when there are so many options to choose from and limited information available about the placesto visit. The project provides users with access to a vast database of places, complete with ratings and user-generated comments, to assist in the decision- making process.

The project features a chatbot that users can interact with to get personalized recommendations about places that match their interests. The chatbot is built using Dialogflow, a natural language processing tool .

One of the critical features of the application is the ability for users to leave feedback on places they have visited, including ratings and comments. This information is then used to provide other users with valuable insights into these places . The user-generated comments and ratings also provide an opportunity for businesses and organizations to improve their services based on user feedback.

In this document, we present four chapters. The first chapter covers the host university, our working methodology, and an analysis of the existing situation.

In the second chapter, we focus on the functional requirements specification, non-functional requirements specification, and a presentation of actors with a global use case diagram.

The third chapter includes a class diagram with a table to represent each class, sequence diagrams of both user and admin actors, and a relational data schema.

The last chapter discusses the definition of the MVC architecture, the working environment, and the view of our application with different interfaces for Reviewhub and the dashboard.

CHAPTER 1 : General Presentation

1-Introduction

In this introductory chapter, we start with the presentation of the general ideaof our project.. The second part will be devoted to the presentation of the host university. Third part will be about the working methodology . Afterwards, we approach the study of existing solutions with their criticisms and we introduce the proposed solution.

2-Presentation of the host university :

The Faculty of Sciences of Monastir is a higher education institution located in Monastir, Tunisia. It was established in 2004 and offers undergraduate and graduate programs in various fields of science, including mathematics, physics, chemistry, computer science, life sciences, earth sciences, and environmental sciences. The faculty also conducts research in different areas of science and collaborates with national and international institutions.

Figure 1 : Logo of Faculty of sciences of monastir

Contact information :

Table 1 : FSM's Contact information

Address	l'environnement 5019 Monastir -TUNISIE - 5019 Monastir
Phone	+216 73 500 276
Email	fsm@fsm.rnu.tn
Website	https://fsm.rnu.tn/

3-Working methodology :

3.1-Scrum agile :

4-Study of the existing

4.1Description of the existing

REVIEWHUB is a web-based platform that allows users to share reviews and ratings for various types of entertainment venues. Users can sign up for an account on the platform and log in to add their reviews. Users can rate venues on a scale of 1 to 5 stars. The platform allows users to view reviews and ratings for various

entertainment venues such as restaurants, cafes, malls, and amusement parks. Users can search for venues based on location, category. Each review includes details such as the review title, description, images, location, and category. Users can like or dislike reviews, and view the number of likes and dislikes. The platform provides a centralized location for users to share their experiences and help others make informed decisions when choosing entertainment venues to visit.

4.2-Criticism of the existing :

There are several shortcomings that need to be addressed. Firstly, there is a lack of a comprehensive and centralized platform for users to provide reviews and ratings of amusement places such as restaurants, cafes, malls, and amusement parks. This leads to users having to search for reviews across multiple platforms, which can be time-consuming and frustrating. Secondly, the existing solution does not provide a way for users to interact with each other by liking or disliking reviews, and reporting inappropriate or fake reviews. This results in unreliable reviews and ratings, which can mislead users who are seeking recommendations. Furthermore, the existing solution does not provide a chatbot feature to assist users in their review and rating process. This can be a significant drawback, as users may have questions or require guidance while reviewing and rating amusement places.

4.3-Solution of the existing

The proposal is to develop a web application called REVIEWHUB. This application will allow users to publish reviews and ratings on various entertainment venues such as restaurants, cafes, shopping centers, and amusement parks. Additionally, the application will have a chatbot feature to assist users in easily posting their reviews and ratings. Reviews can be rated on a scale of 1 to 5 stars, and users can

also like or dislike other users' reviews. Reported reviews will be sent by email to the administrator for review. The benefits of this solution are numerous:

- Better usability for users thanks to the simple and intuitive user interface.
- A centralized platform for all reviews and ratings of various entertainment venues.
- A chatbot feature that will allow users to easily post their reviews and ratings.
- A rating feature that will allow users to rate reviews using a scale of 1 to 5 stars.
- A like/dislike feature that will allow users to react to other users' reviews.

5-Conclusion

In this chapter we focused on the introduction of the project and the host organization of the study of the existing solutions and the proposal of a new web application called REVIEWHUB. The existing solutions were found to have several shortcomings, including a lack of a centralized platform for reviews and evaluations of various entertainment venues.

CHAPTER 2 : Specification of Requirements

1. Introduction

The Specification of Requirements is an important document in software development projects that sets the foundation for the entire project and acts as a guide for the development team. This document is crucial as it outlines thefeatures, user interface, database design, and other technical specifications that the application needs.

The purpose of this document is to clearly define what the application needs to do, and what is required to make it successful. It provides a comprehensive overview of the project so that everyone involved understands what needs to be done to meet the needs of the users.

This document will be used as a reference throughout the development process to ensure that the project stays on track and meets the user's needs. By defining the requirements at the beginning, the development team will work more efficiently, with clear guidelines to follow, preventing misunderstandings, delays, and rework. Ultimately, the Specification of Requirements is critical to the project's success, ensuring that the application is delivered on time, within budget, and meets the satisfaction of its users.

2-Functional Requirements Specification :

Administrator:

- User Management
    - View Users
    - Update User
    - Delete User

- Reviews Management
    - View Reviews
    - Delete Review

- Posts Management
    - View Posts
    - Create Post
    - Update Post
    - Delete Post

User :

- Search Places
- Add Reviews
- Update Reviews

- Delete Reviews
- Add to Favorite
- Delete From Favorite
- Update Account Infos

Chatbot:

- Answer to users questions

3-No Functional Requirements Specification :

- Usability: The application is easy to use and navigate, with clear and concise instructions provided to the users.
- Performance: The application is fast and responsive, with minimal lag or delay in loading pages and processing user requests.
- Security: The application is secure and it protects user data from unauthorized access or attacks, using appropriate encryption and access control measures.
- Reliability: The application should be reliable and available at all times, with minimal downtime for maintenance or upgrades.
- Friendly: The application should be compatible with a wide range of devices, browsers, and operating systems, to ensure that it can be used byas many users as possible.
- Scalability: The application should be designed to handle a large number of users and requests, without slowing down or crashing.

4- Presentation of Use Cases :

4.1- Presentation of Actors :

4.1.1- Administrator : The administrator is a user who has full control over the website's content and functionality. They can add, edit or delete any content that is published on the website. They also have the ability to manage user accounts and moderate reviews that are submitted by users.The administrator is responsible for ensuring that the website is functioning correctly and that it is providing a positive user experience.

4.1.2- User : The visitor is a user who is registered on the website and canaccess all the pages of our website. They can post and view the reviews and ratings of various amusement places and can like or dislike any review, and can also use the chatbot feature to ask questions related to amusement places.

5- Global Use Case Diagram :

After specifying the actors who will interact with our system, in this part we will translate the different interactions between the different actors and the system into a graphical representation, namely the use case diagram that illustrates the interaction between the actor and the system.

5.1- Functionality of each actor :

Table 2 : Actors Functionality

Actors	Functionality
Admin	User management Review management Places management

User	Authentication Registration Update account Add/Remove or update Review to place Like/Dislike a review Report a review Add/Remove place to favorite

5.2-Global Use Case Diagram :

The diagram below (Figure 1) represents the use case diagram in a global way for each actor in our system.

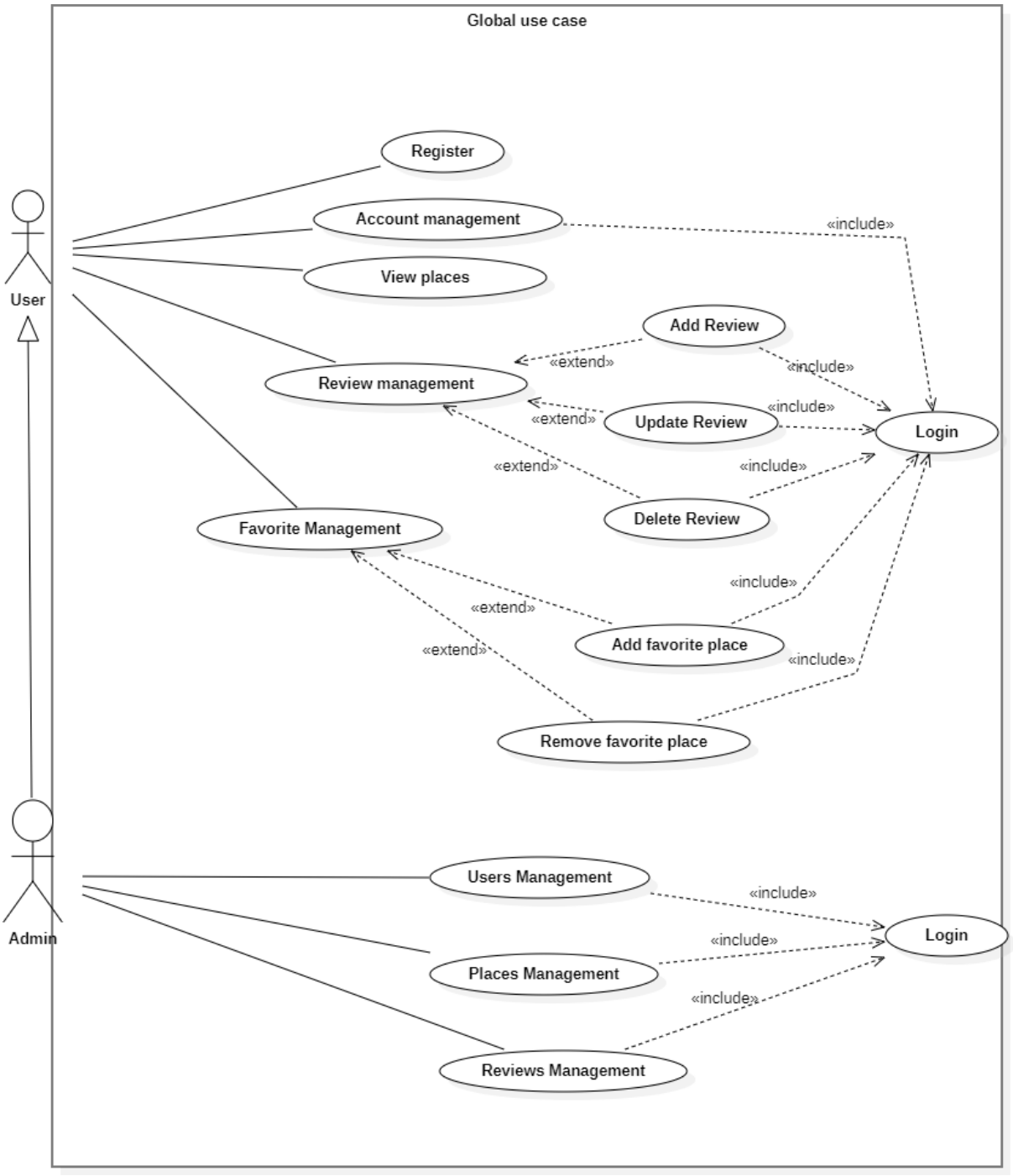

Figure 2 : Global use case

This table represents user functionality:

Table 3: user functionality

User Functionality	
Actor	User
precondition	Authenticated user
Functionality	
User can add or remove any place from favorite	
User can add,update or remove review	
User can Update his account informations	

5.3-User account management Use case diagram :

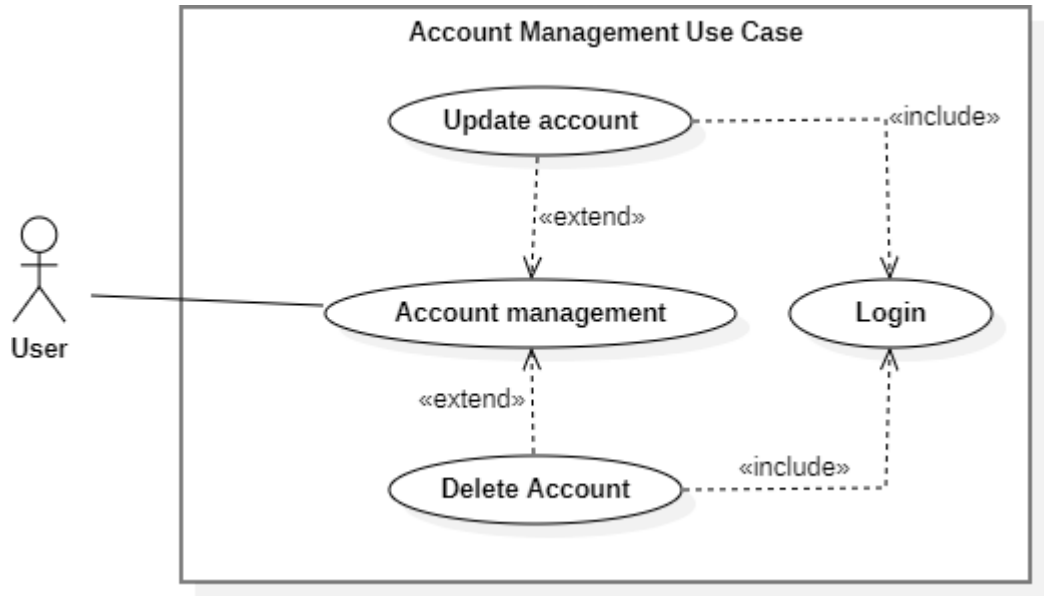

Figure 3 : Account Management use case:

This table represents user's account management :

Table 4: User Account Management

User account management	
Actor	User
precondition	Authenticated user
Functionality	
User can update his account informations	

User can delete his account

5.4-Admin places management use case diagram :

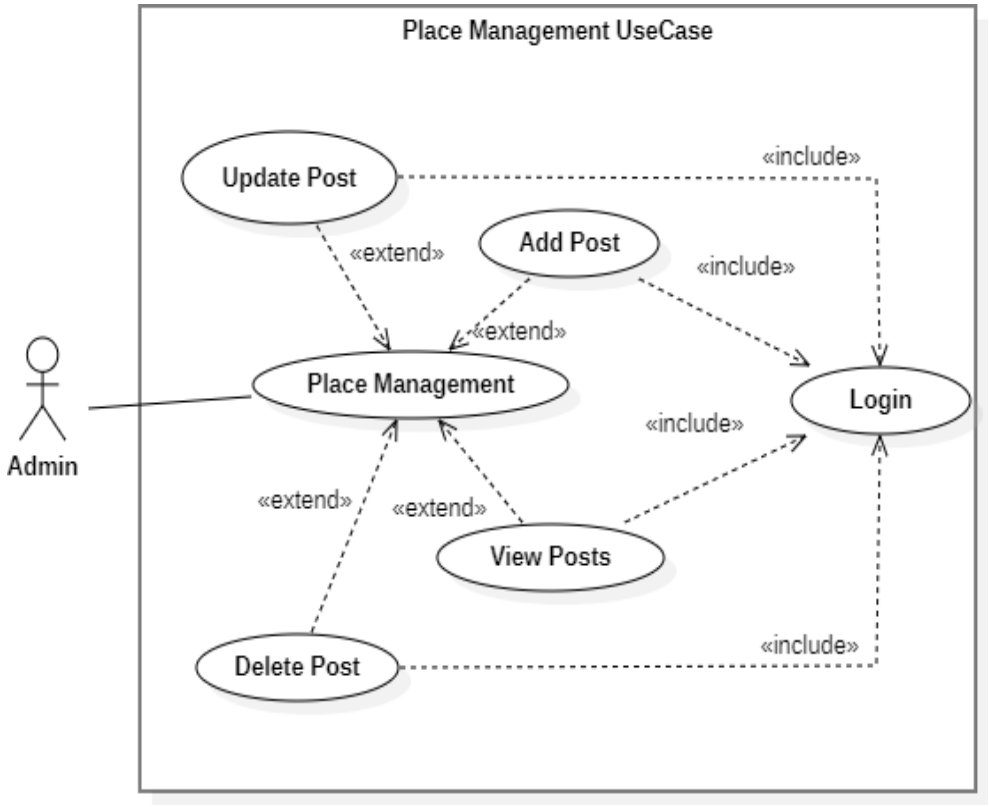

Figure 4: Place Management use case

This table represents admin place management :

Table 5: Admin Place Management

Admin management	
Actor	Admin

Functionality
Admin can update post informations
Admin can delete or add new place
Admin can view places

5-5:Admin places management use case diagram :

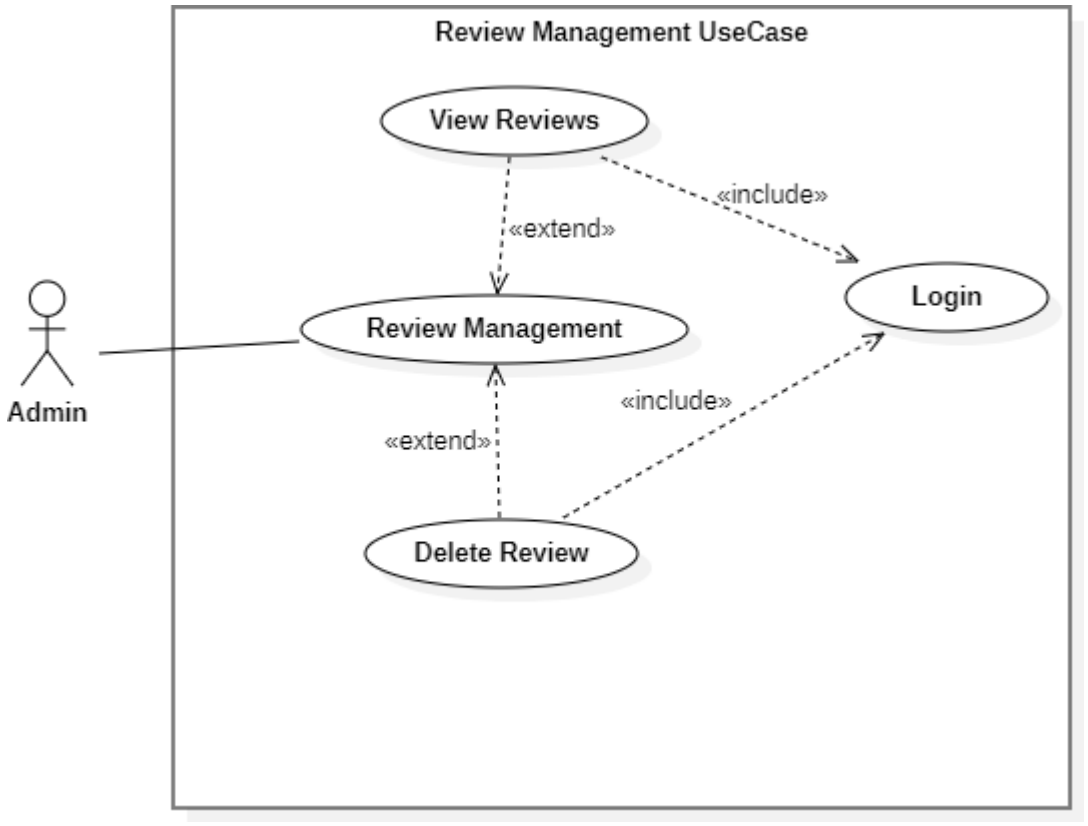

Figure 5: Review Management use case

This table represents admin place management :

Table 6 : Admin Place Management

Admin management	
Actor	Admin
Functionality	
Admin can update post informations	
Admin can delete or add new place	
Admin can view places	

Chap 3 : Conception

1-Introduction :

In this chapter, the student must model their application from a static and dynamic perspective. For dynamic modeling, sequence diagrams, collaboration diagrams, and state diagrams should be included. To model the static aspect, the class diagram and deployment diagram should be presented at the end of this chapter.

2-Architecture :

In this section, we will describe the architecture to be used in both parts of our application.

2.1MVC : Model-view-controller

MVC stands for Model-View-Controller. It is a design pattern used in software development to organize code and separate the concerns of the application.

The Model represents the data and business logic of the application. The View displays the data to the user and handles user input. The Controller acts as an intermediary between the Model and the View, and manages user input and updates to the Model.

By separating the concerns of the application, it becomes easier to maintain and modify the code.

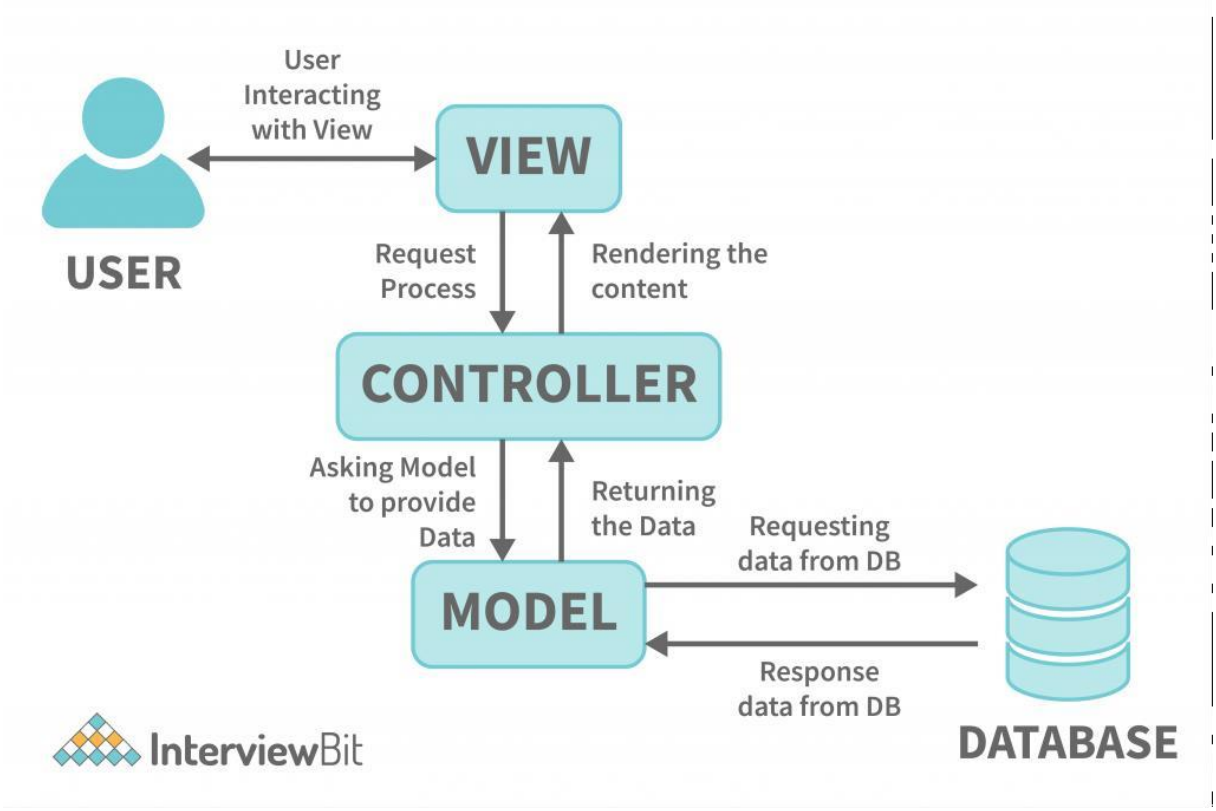

Figure 6: MVC conception Model

- ➔ Model : The model represents the data and business logic of the application. It is responsible for managing the application's data, processing user input, and responding to user requests.
- ➔ View : The view is responsible for rendering the user interface and displaying data to the user. It receives input from the user and sends it to the controller for processing.
- ➔ Controller : The controller acts as an intermediary between the model and the view. It receives user input from the view, processes it using the model, and then updates the view with the results. It is responsible for coordinating the flow of data between the model and the view, and for controlling the overall behavior of the application.

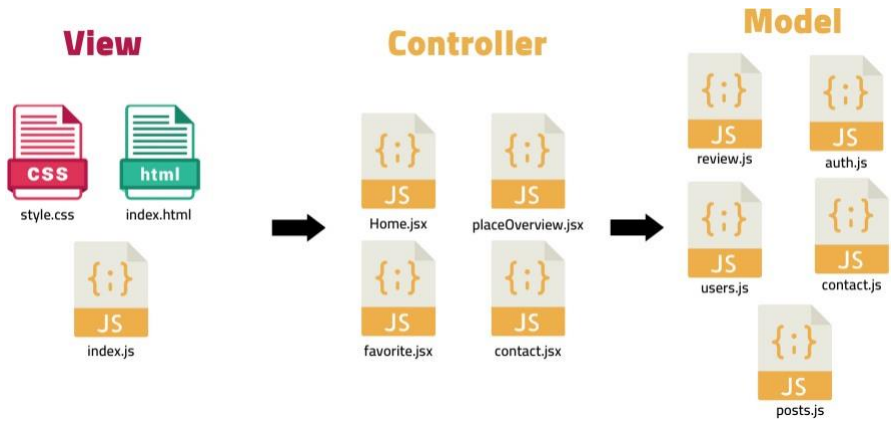

Figure 7: MVC Model of our web app

as appears in the figure The Model represents the data and the business logic, which is implemented using Node.js and MongoDB. The View is responsible for presenting the data to the user, and in our case, it is implemented using React.js and Tailwind CSS. Finally, the Controller is responsible for coordinating the flow of data between the Model and the View, and for controlling the overall behavior of your application. It receives user input from the View, processes it using the Model, and updates the View with the results. In this way, the MVC pattern helps to keep our code organized, maintainable, and scalable.

3 - Class Diagram :

A class diagram is a visual representation of the classes, interfaces, and relationships in a system. It shows how different parts of the system are related and how they interact with each other. It helps in understanding the overall architecture of the system and makes it easier to identify potential problems or areas for improvement.

This class diagram represents the different entities in your project. The "User" class holds information about users such as their username, email, password to authenticate to the website. The "Review" class holds information about reviews such as the person who wrote the review, the post it was written for, and the rating and comments. The "Post" class holds information about posts such as the category it belongs to, its title, content, images, likes, and dislikes. The "Category" class holds information about the different categories that posts can belong to, and the "Contact" class holds information about user inquiries such as their email, subject, and message.

The "Admin" class represents the admin user, who has a username and password to log in and manage the content on the website. The "Chatbot" class represents a feature that can be implemented to interact with users and provide assistance.

The navigability between the classes

- every user can post many reviews
- every place can have 0 to many reviews
- every place has a category
- every user has an chatbot

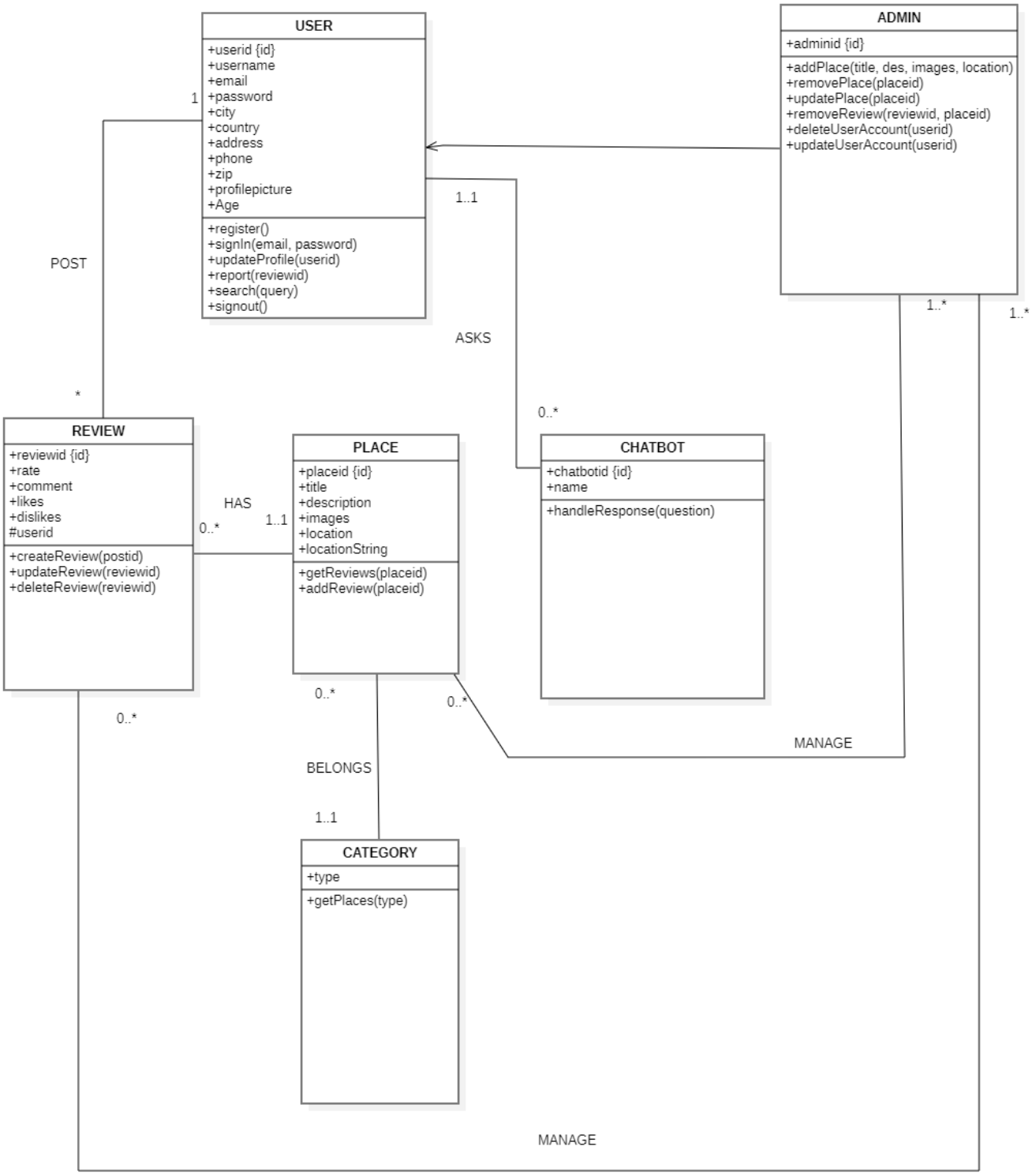

Figure 8:: Class Diagram

- User class :

  This table represents user class :

Table 7 : User Class

Attributs	type	Description
userid	String	The id of the user
username	String	The username of the user
email	String	The email address of the user
age	Int	The age of the user
password	String	The password the user account
city	String	The city of the user
country	String	The country of the user
address	String	The address of the user
phone	String	The phone number of the user
zip	String	The zip code of the user
profilepicture	String	The profile picture of the user

- Review class :

This table represents review class :

Table 8 :Review Class

Attributs	Type	Description

reviewid	String	the id of a review
rate	String	The rate posted in the place
comment	String	The comment posted in the place
Likes	Array	The users who like a specific place
Dislikes	Array	The users who dislike a specific place
userid	String	The id of the user who posted this review

- Place class

this table represents place class :

Table 9 : Place Class

Attributes	Type	Description
placeid	String	the id of the place
Title	String	The title of the place
Description	String	The description of the place
Images	Array	The images of the place
Location	String	The location url in google maps of the place
LocationString	String	The location of the place

- Admin class :

this table represents Admin class:

Table 10 : Admin Class

Attributs	Type	Description
adminid	String	the id of the admin

- Category class:

this table represents Category class:

Table 11 : Category class

Attributs	Type	Description
type	String	TheType of place

4- Sequence Diagram :

A sequence diagram is a visual representation of the interactions between objects or components in a system, showing the order in which they occur. It describes how objects or components communicate with each other to achieve a specific goal or perform a certain task. Sequence diagrams are used to illustrate the dynamic behavior of a system, including how data is exchanged between objects and the sequence of events that occur during a particular operation or scenario. In other words, sequence diagrams show the flow of messages between objects and the timeline of their execution. They are useful for designing, testing, and debugging complex systems, as well as for documenting their behavior.

4.1- <<Authentication>> Sequence Diagram :

The figure represents the authentication sequence diagram , the user enters his username and passwords in the sign in form , Then The controller send request to the model to find out that the user is exist or no

if the data is correct the controller send request to the view to redirect to the home page , else it display an error

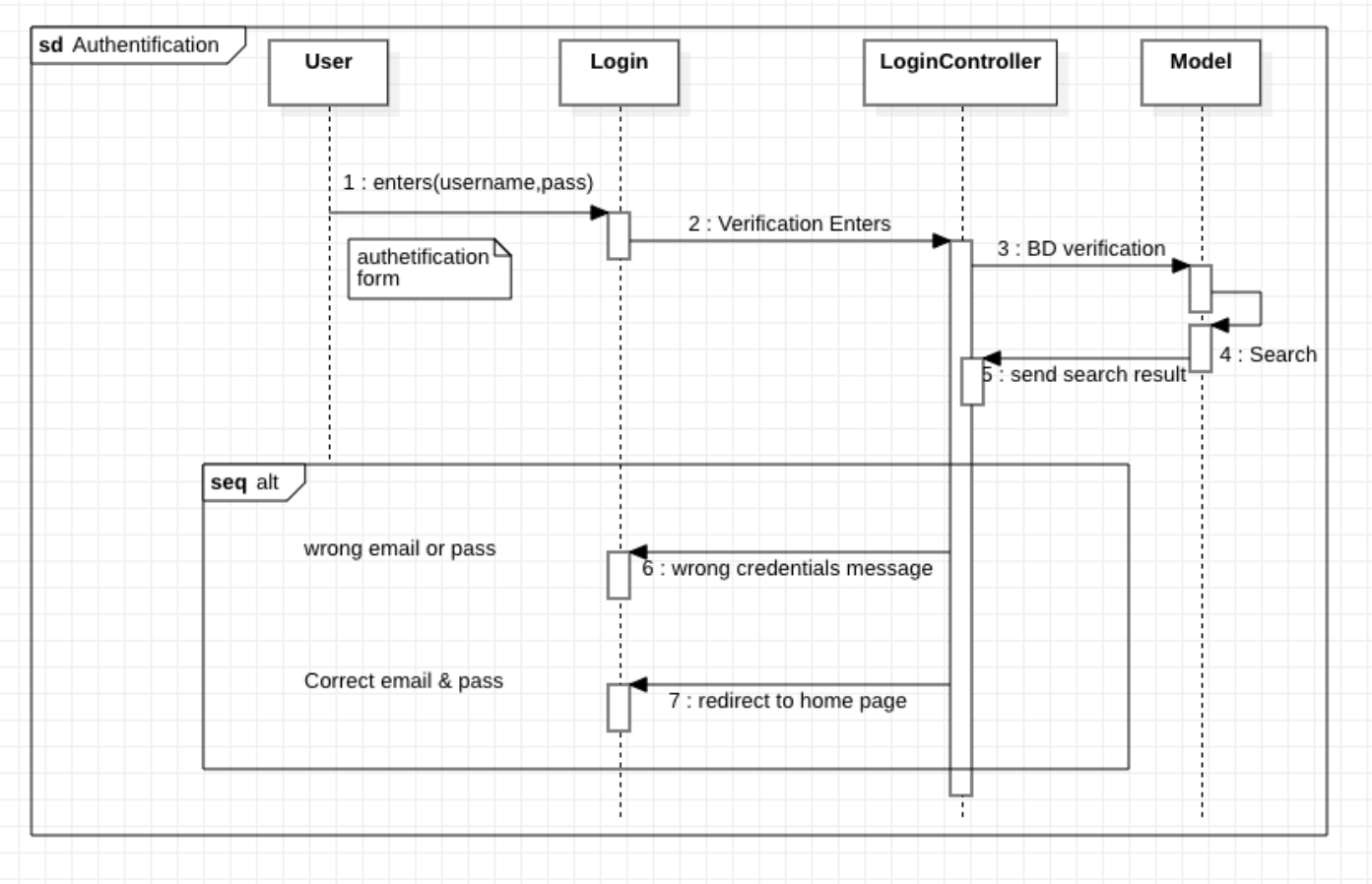

Figure 9 : :<<Authentication>> Sequence Diagram

4.2- <<Add a Place>> Sequence Diagram :

The figure shows the diagram for adding a place. The administrator enters information related to the place he wants to add in the add form presented by the view. If the data is verified by the controller, it sends an add request to the database."

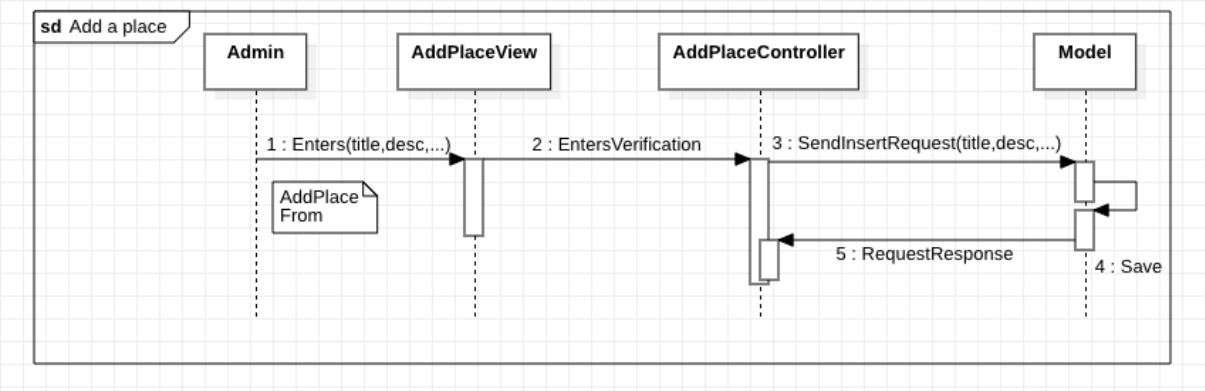

Figure 10: <<Add a Place>> Sequence Diagram

4.3- <<Update Place>> Sequence Diagram :

The figure explains the process of updating a place . The administrator fills out the information related to the place they want to update in the update form presented by the view. If the data is validated by the controller, it sends an update request to the database

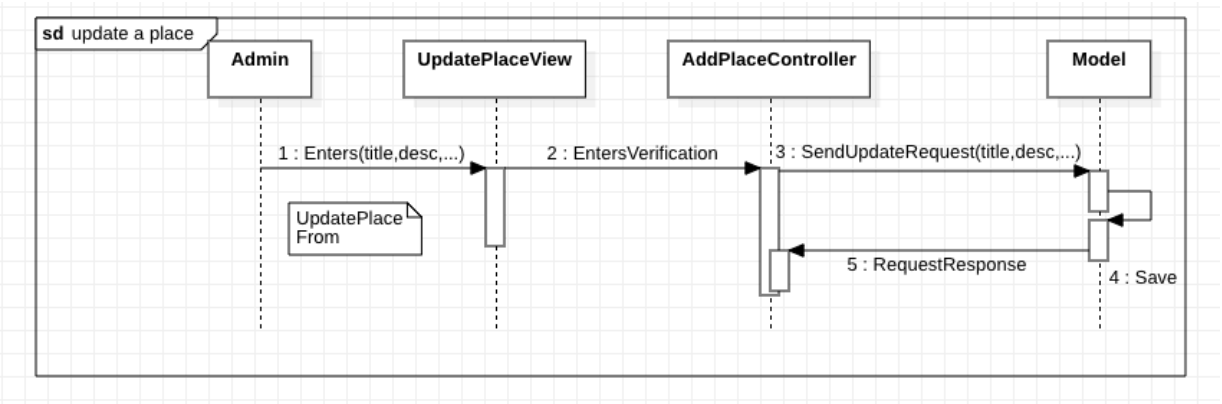

Figure 11: <<Update Place>> Sequence Diagram

4.4- <<Add Review>> Sequence Diagram :

The diagram shows the process of adding a review. The user fills out the review information in the form presented by the view. If the controller verifies the data, an add request is sent to the database

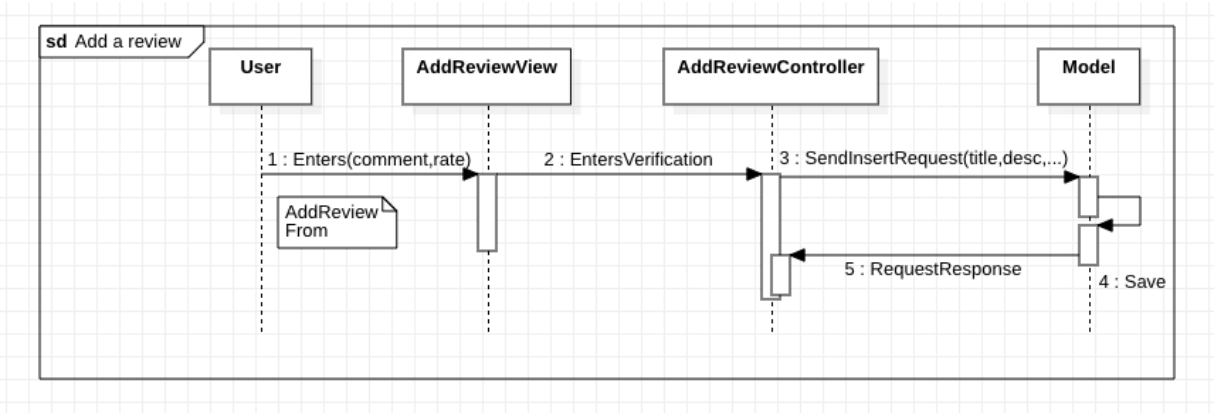

Figure 12: << Add Review>> Sequence Diagram

4.5- << Update Review>> Sequence Diagram :

The Figure shows the diagram of the process of updating a review. The user enters the relevant information into the update form provided by the view. If the controller validates the data, an update request is sent to the database

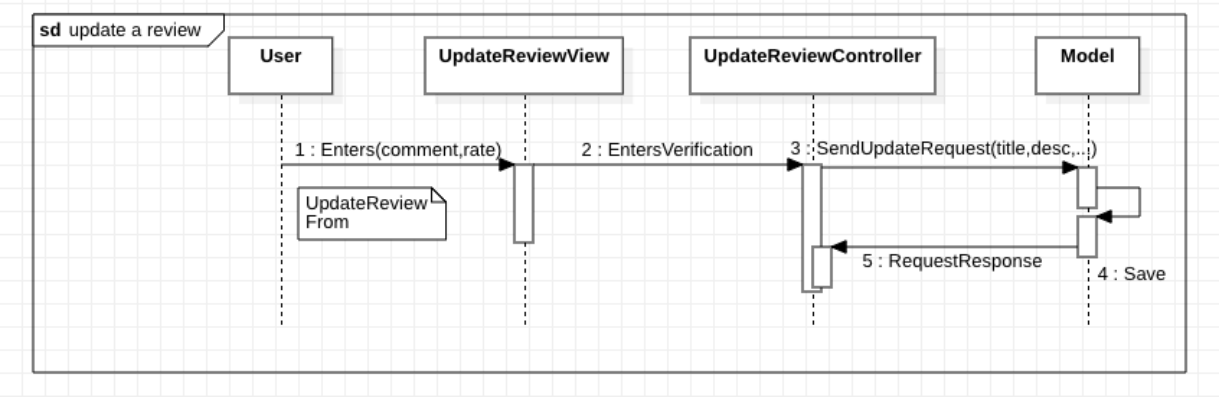

Figure 13: << Update Review>> Sequence Diagram

5-Relational data schema :

user(userid, username,email,password,age,city,country,address,phone,zip,profilepicte)

review(reviewid , rate, comment, likes , dislikes, #userid)

place(placeid , title , description, , images , location , locationString)

admin(adminid)

category(type)

chatBot (id,name)

6-Conclusion:

In this chapter we've created various diagrams, including class diagrams, sequence diagrams, use case diagrams, and a relational data schema. These diagrams helped

us identify the important components, their relationships, and necessary functions and interactions. With this understanding of the system, we're now well-prepared to move on to the implementation and testing phase.

Chap 4 : Realization

1-Introduction :

The realization phase is the operational phase of the creation of a web app. It consists in the concretization and materialization of the entire analysis phase.

In this part we focus on the realization of the application. We begin by describing the hardware environment and software environment which contains all the technologies used to develop our application by illustrating our approach through the interfaces produced.

2-Development environment: we going to represent the workenvironment that we used to work on our project

2.1-Hardware environment :

Table 12 : Hardware environment

	PC1	PC2
CPU	i5 1,4 GHZ	Intel Core i7-6700HQ 6è Gén, up to 3.5 Ghz
RAM	8gb	8GB
DISQUEDUR	128 GB SSD	1TB HDD, 128GB SSD
Graphic Card	Intel HD Graphics 5000	GeForce GTX 950M

2.2-Software Environment:

StarUML : is a modeling tool that allows you to create and edit UMLdiagrams such as use case diagrams, class diagrams, sequence diagrams, and more. It provides an intuitive interface for drawing diagrams, and supports a variety of export options. The tool is commonly used by developers and software engineers to design, document, and communicate software systems. It can help to improve the quality of software development by providing a visual representation of system design, facilitating collaboration and communication among team members, and helping to identify potential issues and areas for improvement

Figure 14 : Staruml's logo

VSCode : is a widely used source code editor with a range of powerful features. It provides a user-friendly interface, syntax highlighting, code completion, and debugging capabilities, among other things. VSCode also has a wide range of extensions and plugins available that can be used to extend its functionality and enhance the development experience. It supports multiple programming languages and provides excellent integration with version control systems such as Git. Overall, VSCode plays a crucial role in streamlining the development process and improving productivity for developers.

Figure 15 :visual studio code logo

MongoDB : is a NoSQL document-based database that stores data inflexible and scalable JSON-like documents. It is designed for ease of use, high performance, and scalability, and is widely used for web and mobile applications, real-time analytics, and content management systems. MongoDB uses a dynamic schema, which

means that you can add fields to documents without having to update a centralized schema or taking down the database. It also offers a range of features such as replication, sharding, and aggregation, that make it an attractive choice for building modern applications.

Figure 16 : MongoDB logo

ReactJS : is a JavaScript library that is commonly used for building user interfaces for web applications. It allows developers to create reusable UI components and provides a fast and efficient way to update the user interface in response to changes in data. ReactJS makes use of a virtual DOM (Document Object Model) to manage the rendering of components, which makes the application faster and more responsive. It is widely used in modern web development and is supported by a large and active community of developers.

Figure 17 : Reactjs logo

Node.js : is an open-source, cross-platform runtime environment that allows developers to build server-side applications using JavaScript. It provides an event-driven, non-blocking I/O model that makes it lightweight and efficient, making it a popular choice for building scalable and high-performance web applications. Node.js comes with a built-in library of modules, making it easy to build web

servers, APIs, and other server-side applications. It is also compatible with various databases and provides an easy way to interact with them using packages such as mongoose for MongoDB. Overall, Node.js provides a fast and efficient way to build server-side applications using JavaScript

Figure 18 : Nodejs logo

Tailwind CSS: is a popular utility-first CSS framework that makes it easy to create custom user interfaces. It provides a large set of pre-

designed CSS classes that can be used to quickly style HTML elements without having to write custom CSS. Tailwind CSS also allows for easy customization and extension of the default styles through configuration files and custom classes. It is often used in combination with front-end frameworks like React or Vue.js to streamline the development of responsive and modern web applications.

Figure 19 : Tailwindcss logo

Dialogflow : is a natural language processing (NLP) platform that allows developers to build conversational interfaces, such as chatbots, voice assistants, and automated customer service systems. It provides a graphical interface for creating custom language models and integrates with various messaging platforms, including Facebook Messenger, Slack, and Google Assistant. Dialogflow uses machine learning algorithms to understand user input and generate appropriate responses, making it easy to build

conversational experiences that can understand and respond to natural language queries.

Figure 20 : Dialogflow logo

3-App Implementation :

In this section, we will present some interfaces of our application while explaining their uses and functionalities

3.1User Space :

3.1.1- HomePage :

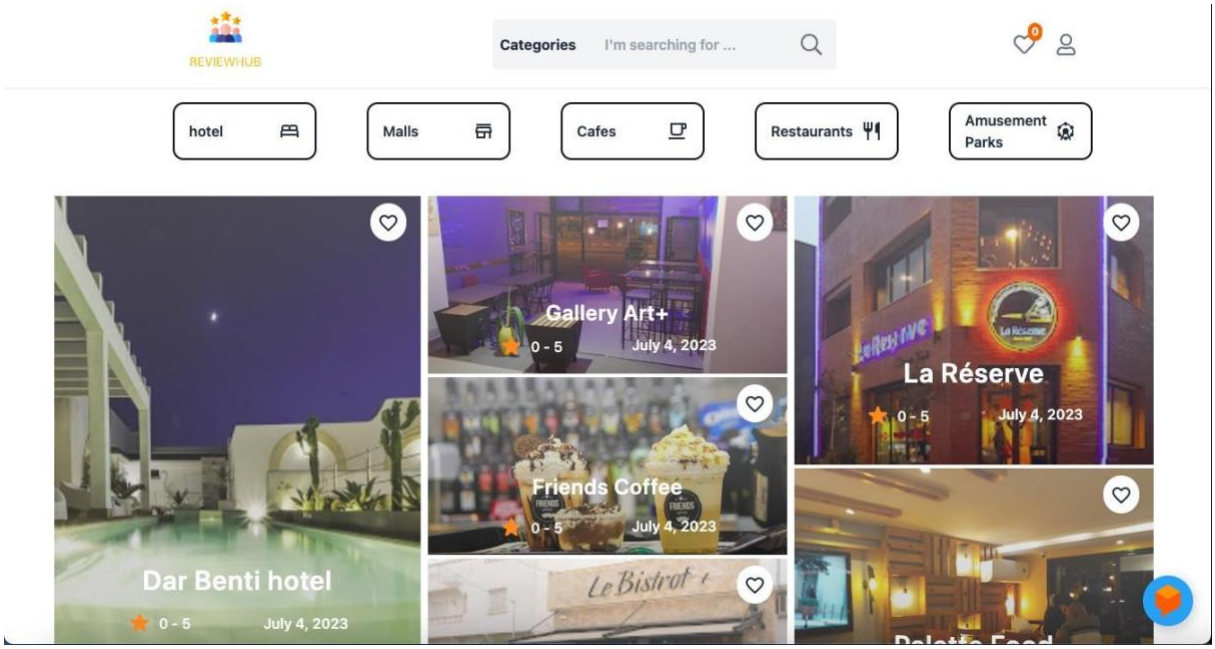

Figure 21 :Home Page Interface

3.1.2-Authentication Interface :

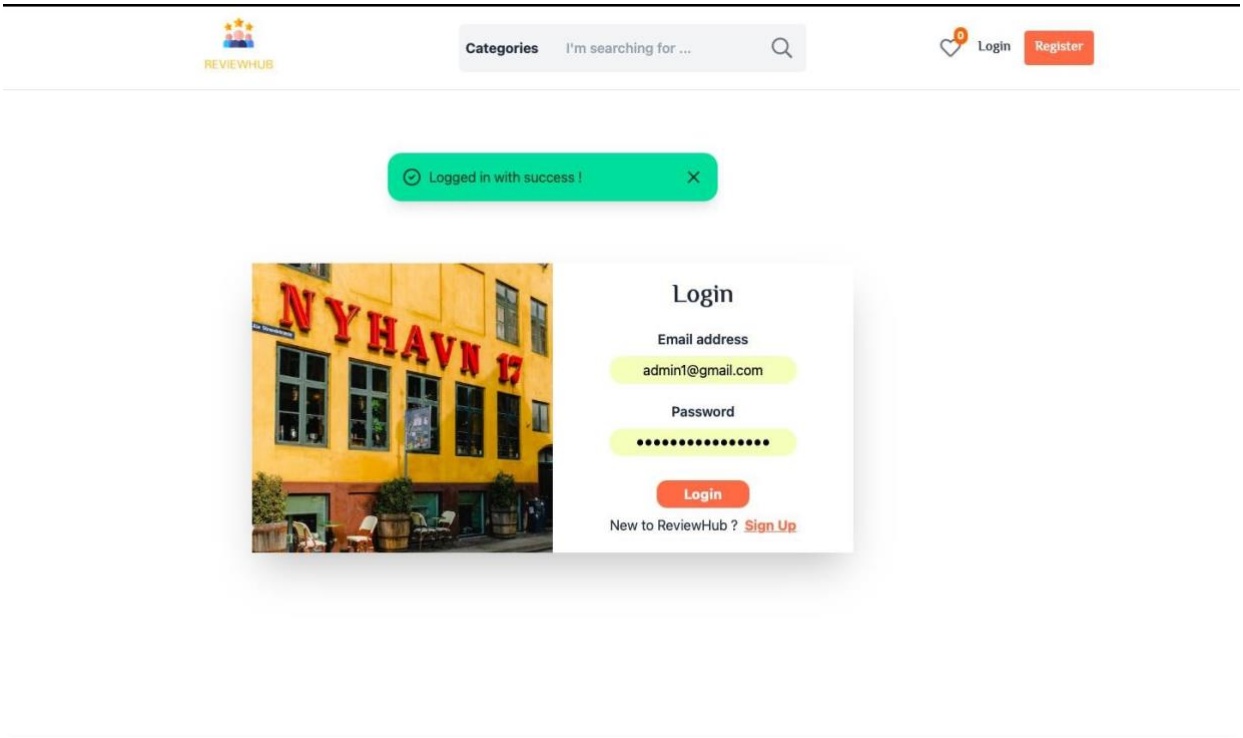

Figure 22 : Login Interface

Like with every app, security is essential. The figure depicts that users can authenticate themselves by entering their username and password to gain access to our website.

3.13 - Register Interface :

That figure express the first authentication step to our web app

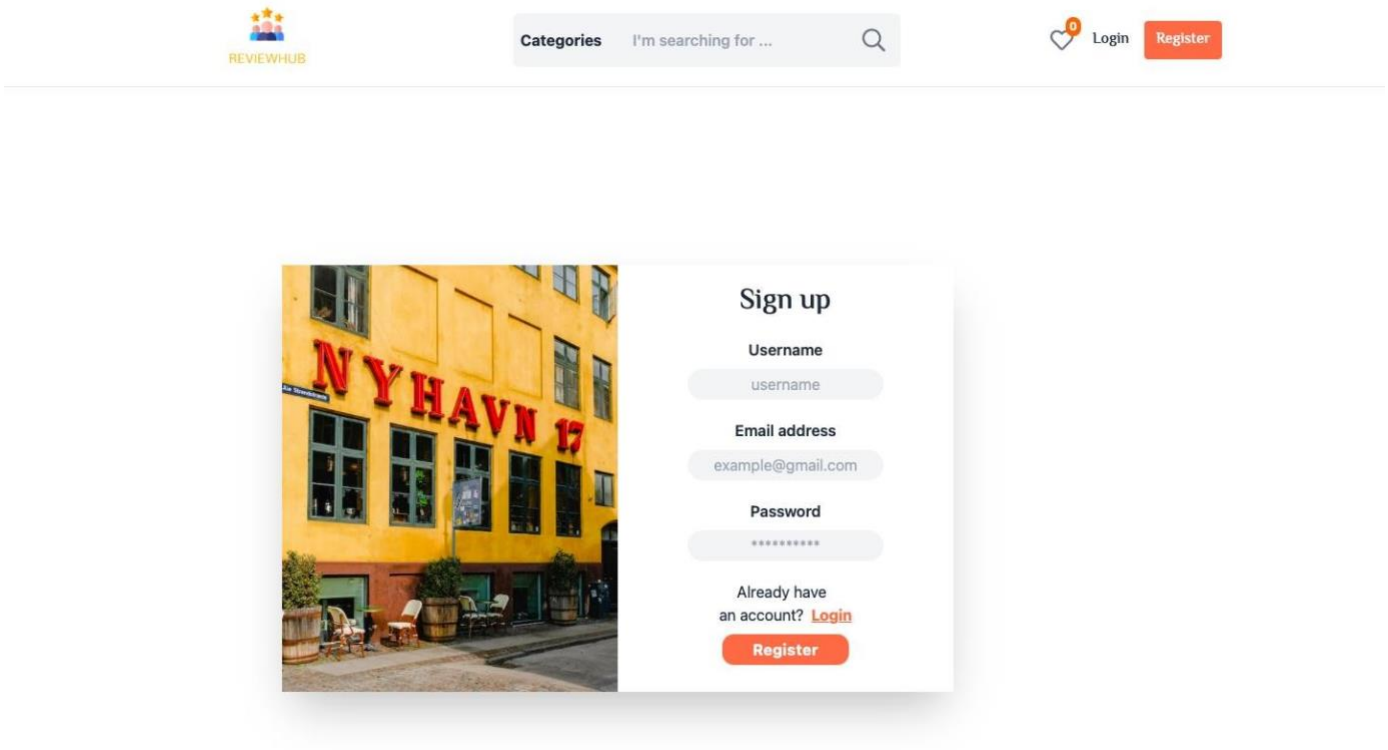

Figure 23: Register Interface

3.1.4- Add Review Interface :

In this figure you can see the form where you can add review and the rate to any post also you can like or dislike or report any review through the report icon

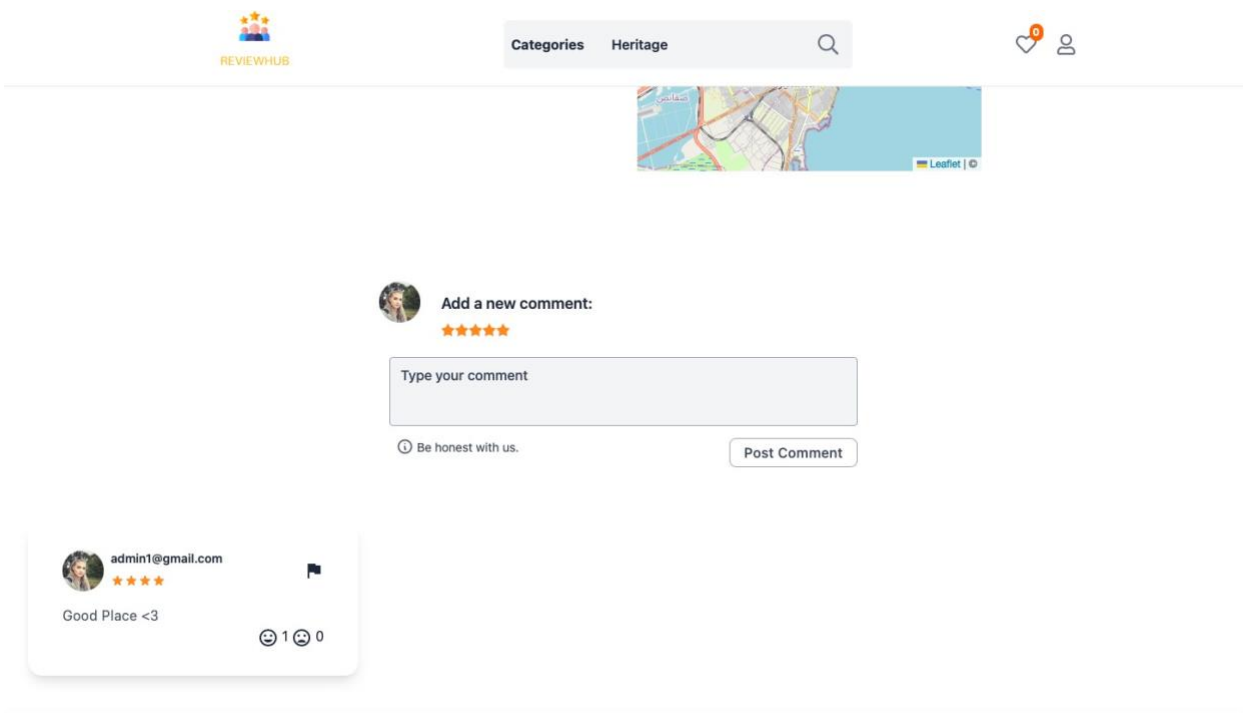

Figure 24 : Add Review Interface

3.1.5 chatbot interface :

Figure 25 : Chatbot Interface

3.2-Admin Space :

3-2.1: home page :

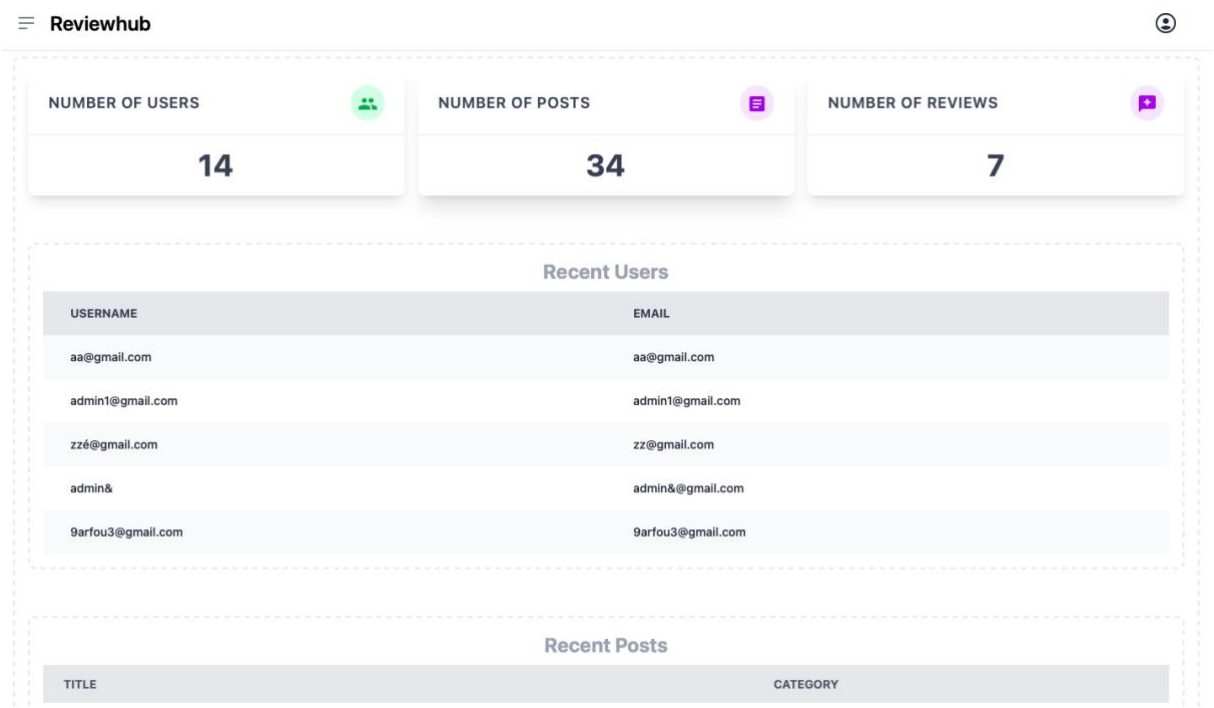

Figure 25 : Dashboard Interface

3.2.2- Add Post :

like what appears in the screenshot , administrator only has access to add post posts to the main website by entering place informations(title , location , category , description) through this form

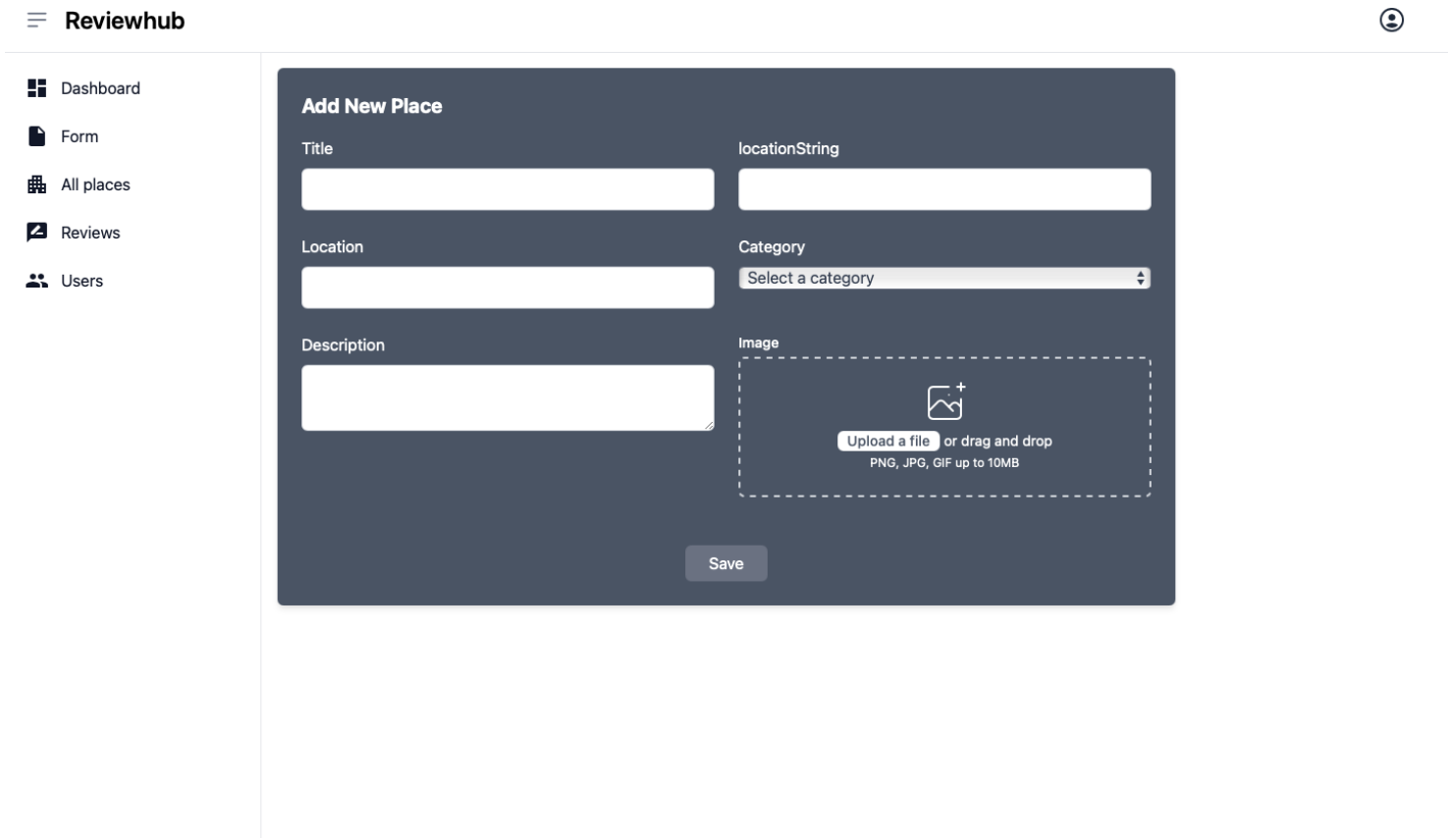

Figure 26 : Add Post Interface

3.2.3- Posts Management :

As an admin, you have the power to manage all posts on the website. This includes the ability to update posts, such as fixing mistakes or adding new information, and delete posts that are no

longer relevant or inappropriate. This helps ensure that the website is accurate and credible, which is essential for building a positive

online reputation. The ability to manage posts is particularly important for websites with user-generated content, as it allows administrators to maintain high standards and provide a quality experience for visitors.

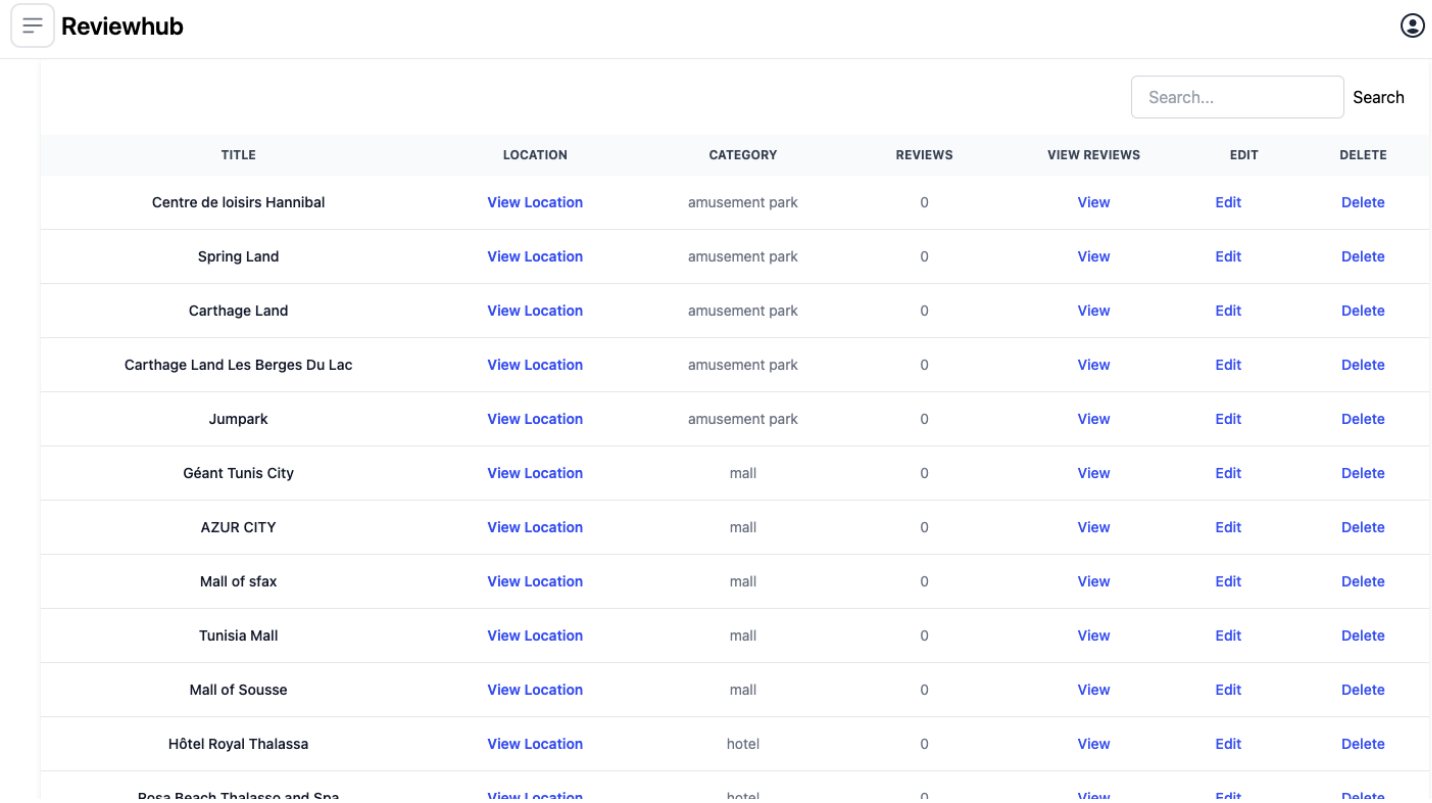

Figure 27 : Places Management Interface

3.2.4: Users Management :

As an admin, you have the ability to update and delete user accounts on the website. This means you can modify user information, such as email address or password, and remove user accounts if necessary. Effective user management helps maintain the security and integrity of the website, protecting both users and the website itself.

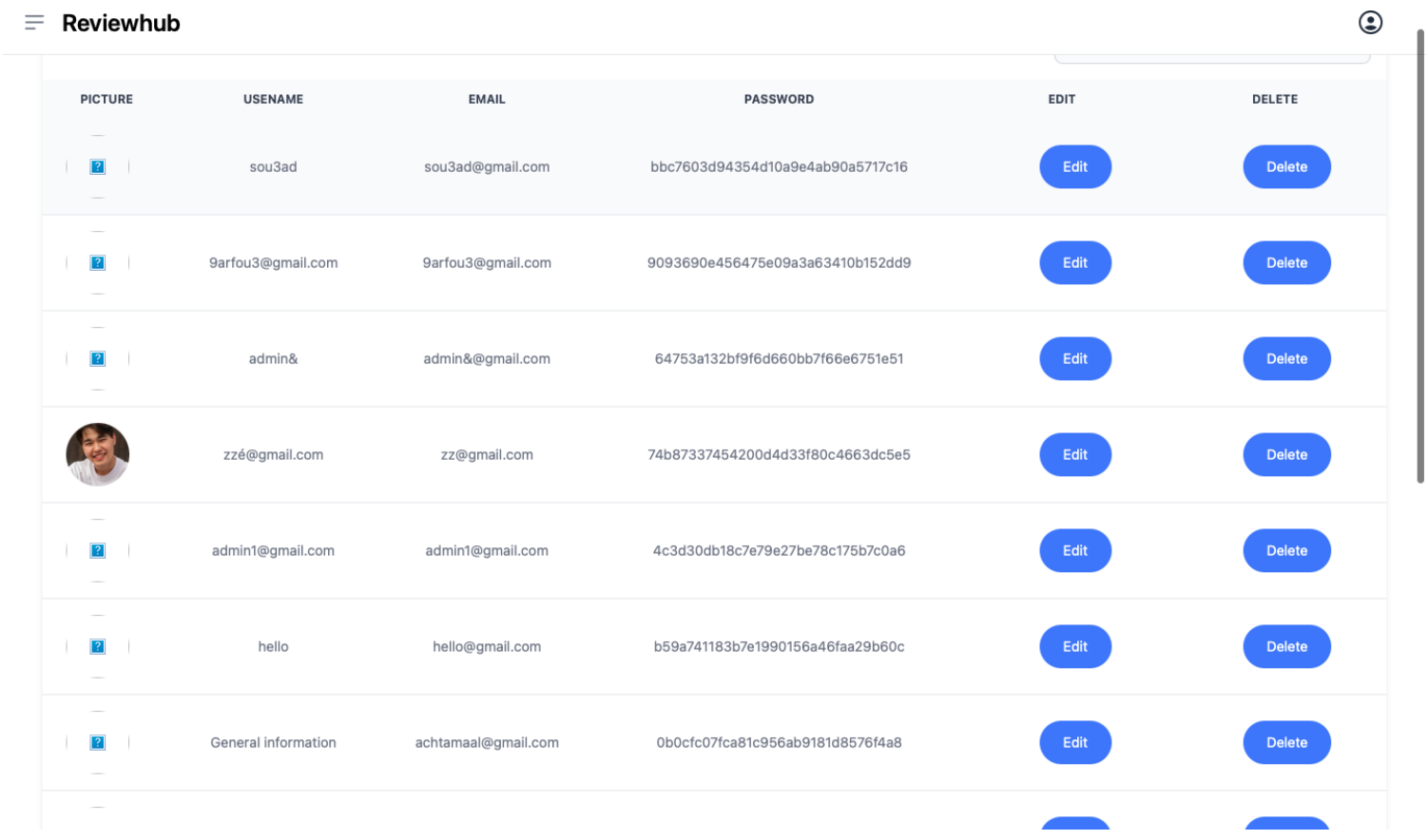

Figure 28 : Users Management Interface

3.2.5 : Reviews Management :

As an admin, you can control all reviews on the website by updating or deleting them as necessary. This is crucial for keeping the website accurate and

trustworthy, which is important for building a good online reputation. It's particularly important for user-generated content, because it helps maintain high standards and creates a great experience for visitors. If any reviews seem inappropriate or go against website policies, users can easily report them

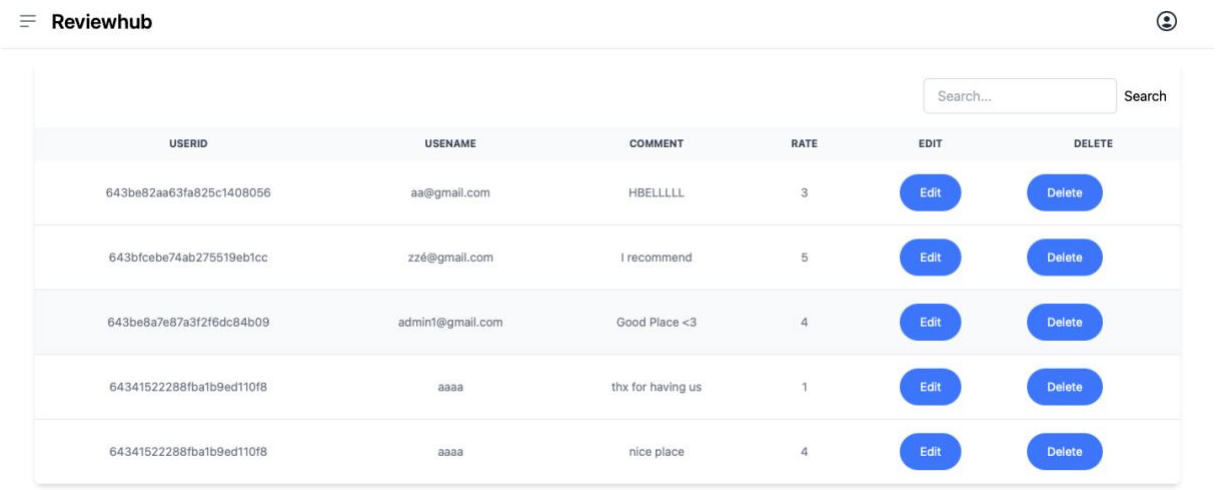

Figure 29 : Reviews Management Interface

Conclusion :

This chapter provides an overview of the work environment, which comprises two main components: hardware and software. Hardware refers to the physical components of a computer system. Software includes various programs and applications that run on the hardware, such as StarUML, Node.js, React.js, and Tailwind CSS. also we have presented some of our web app functionalities by providing some interfaces.

conclusion and perspectives

This end-of-studies project has been created within the Faculty of Sciences of Monastir, utilizing many frameworks such as Node.js, React.js, HTML, and CSS. For the database, we used MongoDB, and StarUML for the conception phase, which wasdivided into three parts: class diagrams, use cases, and sequences.

Our primary objective was to develop a comprehensive review website that would assist people in selecting their ideal destinations based on their preferences. This would be achieved by providing feedback from many people's experiences.

The realization of this project involved many steps, beginning with the first chapter that provided a general overview of the website and its objectives. In the second chapter, we discussed our actors' functionalities and what was required to achieve them,In the third chapter, we detailed the project's conception, including class, use case, and sequence diagrams, to explain how the essential functionalities ,along withthe methodology we followed to simplify our work. . Finally, in the last chapter, we presented screenshots from various parts of our web app to showcase the final product.

We hope to continue making progress on this website in the near future by improvingit, adding more functionalities, fixing any weaknesses, and further developing its strengths.