

# GL Project Report



---

## YAP SHOP:

## AN E-COMMERCE REACT WEB APP

Abderrahmane Aarab · Youssef Saklab

---

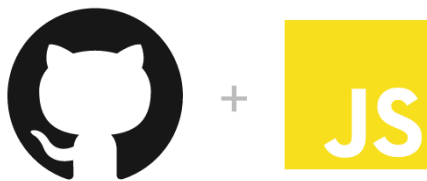
A report submitted in partial fulfillment of the  
Software Development Master's class project



Department of Computer Science  
Moulay Ismail University, Meknes

# Abstract

The YAP Shop is an online pet store developed as a project for a software development class. The e-commerce Web application was built using React JS, Express JS, Node JS, and PostgreSQL. The project aimed to create a platform for customers to purchase pets online. The report provides a comprehensive analysis of the project, including the requirements analysis and design, implementation, testing, and conclusion. The requirements were gathered and used to create a design and architecture for the system, which was then implemented using the technologies mentioned above. The implementation involved the development of the front-end and back-end components, and their connection to provide a seamless user experience. The system was tested to ensure it met the requirements and any issues were resolved. In conclusion, the YAP Shop successfully met its goals and provided a platform for customers to purchase pets online. The report concludes with recommendations for future work and potential improvements.



## AbdoAarab/YAP-PetShop

YAP Shop is an online pet shop made using react



# Table of Contents

Abstract.....	1
List of Figures.....	3
List of Abbreviations.....	4
Introduction .....	5
1.1 Objectives .....	6
1.2 E-commerce.....	7
Chapter 1: Requirements Analysis & Design .....	10
1.1 Project functionalities.....	10
1.2 Conception & Modelling.....	11
1.3 Use-Case Diagram.....	13
1.4 Class Diagram.....	14
1.5 Sequence Diagrams.....	15
1.6 Activity Diagram .....	22
1.7 Global Interaction Diagram .....	23
Chapter 2: Implementation .....	24
2.1 Front-end development with React JS .....	24
2.2 Back-end development with Express JS and Node JS .....	27
2.3 Project Code & Structure.....	29
Chapter 3: Web Application Preview .....	34
3.1 Home Page.....	34
3.2 Pet Categories .....	34
3.3 Pet Catalog .....	35
3.4 Pet Search .....	35
3.5 Pet Details Page .....	36
3.6 Shopping Cart .....	36
3.7 Login.....	37
3.8 Add New Pet .....	37
3.9 Update Pet Details .....	38
Conclusions & Perspectives.....	39

## List of Figures

Figure 1: Git Repository .....	1
Figure 2: Use Case diagram .....	13
Figure 3: Class diagram.....	14
Figure 4: Sequence diagram – Authentification .....	15
Figure 5: Sequence diagram - View the pets .....	16
Figure 6: Sequence diagram - Maintain the catalog .....	17
Figure 7: Sequence diagram - Create an account .....	18
Figure 8: Sequence diagram - Manage the shopping cart .....	19
Figure 9: Sequence diagram - Search for a pet.....	20
Figure 10: Sequence diagram - Place an order .....	21
Figure 11: Activity diagram - Make an order .....	22
Figure 12: Global interaction diagram .....	23
Figure 13: Project code structure .....	29
Figure 14: Front-end code structure .....	30
Figure 15: GET pets request to the API .....	31
Figure 16: Back-end code structure .....	32
Figure 17: Pet data routes .....	33
Figure 18: Home Page preview .....	34
Figure 19: Pet Categories preview .....	34
Figure 20: Pet Catalog preview .....	35
Figure 21: Pet Search preview .....	35
Figure 22: Pet Details Page preview.....	36
Figure 23: Shopping Cart preview .....	36
Figure 24: Login preview .....	37
Figure 25: Add New Pet preview .....	37
Figure 26: Update Pet Details preview.....	38

## List of Abbreviations

**API:** Application Programming Interfaces, make software development and innovation simpler by making it easy and safe for programs to share data and functions.

**DOM:** Document Object Model is a programming interface for HTML and XML documents. It shows what the page looks like so that programs can change the document's structure, style, and content. It shows the document with nodes and objects.

**JSON:** JavaScript Object Notation format is used in data storing and transmitting

**JSX:** JavaScript XML is an extended syntax that allows programmers to write HTML in React easily.

**NPM:** Node package manager is a Node.js tool to construct and administer JavaScript programming libraries.

**REST:** Representational State Transfers is an architectural style applied to networked applications. It exists as a series of constraints applied to implementations of network elements, allowing for unified interface semantics, rather than application-specific syntax and implementations.

**UI:** User Interface is everything a user interacts with when using a digital product or service.

# Introduction

An e-commerce web application is an online portal that is used to sell, buy, and exchange goods and services over the Internet and is also known as an electronic business system. Day by day it is gaining lots of popularity and so that reason I give concern towards the e-commerce platform and develop a site that will be beneficial for the users.

E-commerce web applications for businesses are the focus of this report. The outcome is to build the prototype of an e-commerce web application. It's possible to create dynamic websites with JavaScript, which responds to user requests and improves the user experience. React, a JavaScript library is utilized to construct a friendly and efficient user interface.

We give priority to the main concept of e-commerce platform which is business in a web app and there are four types of e-commerce web platform e.g. business-to-business (B2B), consumer-to-consumer (C2C), business-to-consumer (B2C) and consumer-to-business (C2B). Our project focuses on the business to consumer (B2C) type and which has a remarkable capacity for businessmen and consumers.

## 1.1 Objectives

Behind the report, there are general objectives and specific objectives to fulfill which are demonstrated below:

### **General Objective:**

This report is created for completing the project requirement in the software development class of the Master's program under the Faculty of Science (FSM), University of Moulay Ismail Meknes.

### **Specific Objective:**

- i. Learn and get knowledge about the working style with the client and proper utilization.
- ii. The initial target to learn how to manage, coordinate, and communicate teamwork and make a profitable product.
- iii. Develop our skills, as developers, implementation and maintenance of web app, and time management.

## 1.2 E-commerce

### 1.2.1 Definition

Ecommerce, often known as electronic commerce or online commerce, refers to the buying and selling of things and services through the internet. The term "ecommerce" was initially used in the 1960s. With the rising popularity of mobile devices, social media has become a powerful affirmation of the strength and growth of the web page after years of development. Commercial launchers help to expedite the growth of trade (E-commerce).

### 1.2.2 Categories

There are now several types of e-commerce:

**Business to Business** refers to transactions between businesses, corporations, and organizations. Over three quarters of current e-commerce operations is in this group.

**Business to Consumer** Also known as retail ecommerce, this business model comprises sales between online firms and customers.

**Consumer to Company** refers to an individual who sells goods and services to a business or organization.

**Consumer to Consumer** refers to transactions among consumers. There are more variations, such as G2C, G2B, etc., although they are less common than these four.



### 1.2.3 Advantages

**Elevating Buying Process:** E-commerce has accelerated the entire purchasing procedure for consumers. They can purchase things from the comfort of their own homes, without the need to visit actual stores. It saves enormous amounts of time and expedites transactions.

**Minimizing the cost:** Businesses do not need to create physical stores during conducting E-commerce model. There are substantial costs associated with operating a store, including rent, utilities, other bills, and employee compensation. It eliminates the physical costs and conducts all commercial operations via an internet platform.

**Personalizing Shopping Experiences:** Consumers could have a customized shopping experience. Consumers can browse for a large range of products based on their preferences and needs without restriction. On e-commerce websites, customers are presented products depending on their interests and geography.

**24/7 Availability:** The option to shop online is available all the time, 24 hours a day, 7 days a week. One of the best things about e-commerce is that customers can buy things online at any time. Unlike traditional stores, this site doesn't have set times for when it opens and when it closes.

**Global connections:** Without geographical restrictions, online firms can reach and engage with customers in remote locations. People can place orders from any location and receive delivery to their location.

**Clarified product information:** Customers can discover a comprehensive product description. It gives detailed information so that customers can easily compare it to other products and select the best one.

**Retargeting Customers:** Online purchasing has simplified the process of retargeting customers for businesses. While clients are shopping online, the ecommerce enterprise collects a large amount of information about them. Periodically, customers can be contacted by sending them personalized emails, messages, promotions, and discounts.

### **1.2.4 Disadvantages**

In the case of online purchasing, customers lack the ability to touch and feel. Customers are sometimes happier with in-person purchases since they may inspect the product before to purchase.

Customers are unable to obtain assurance regarding the quality of the products. They may be deceived by businesses and acquire defective goods.

Security Concerns: During online purchasing online, customers have the risk of being stolen crucial credentials. Hackers may steal client information and lead to serious financial lost.

Another significant disadvantage of e-commerce is that buyers must wait longer to receive their packages. With offline shopping, clients receive immediate delivery of their purchases.

Customers are not able to test out a product before making a purchase when they shop online. This is a disadvantage. They cannot negotiate prices and acquire additional information about the product's usage and qualities, as they would in a real store with a salesperson.

# Chapter 1: Requirements Analysis & Design

A good project plan and design makes any project a lot easier. Project design is an initial stage of the project. In the following sections, we give full concern towards the UML diagrams of the project, project functionalities and implementation, Database structure, and Project Analysis.

## 1.1 Project functionalities

This gives us an idea on the features we want our Web application to have and then use them as programming goals to measure our progress. The Web application supports the following features:

- Add pets to your cart
- Modify quantity of pets in your cart
- Checkout and pay for your order as you see the total cost.
- Log in to create products.
- Log in to edit and delete products

## 1.2 Conception & Modelling

As a first benchmark, we set the features that our Web application should have. Then we start with an overview of the Web application to determine how the UI and UX are going to play along with each other. Followed by that, we use UML (Unified Modeling Language), which is a general-purpose, developmental, modeling language in the field of software engineering, to provide a standard way to visualize the design of our application system.

### 1.2.1 Unified Modelling Language (UML)

UML, short for Unified Modelling Language, is a standardized modelling language consisting of an integrated set of diagrams, developed to help system and software developers for specifying, visualizing, constructing, and documenting the artifacts of software systems. The UML is a very important part of developing object oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects. Using the UML helps project teams communicate, explore potential designs, and validate the architectural design of the software.

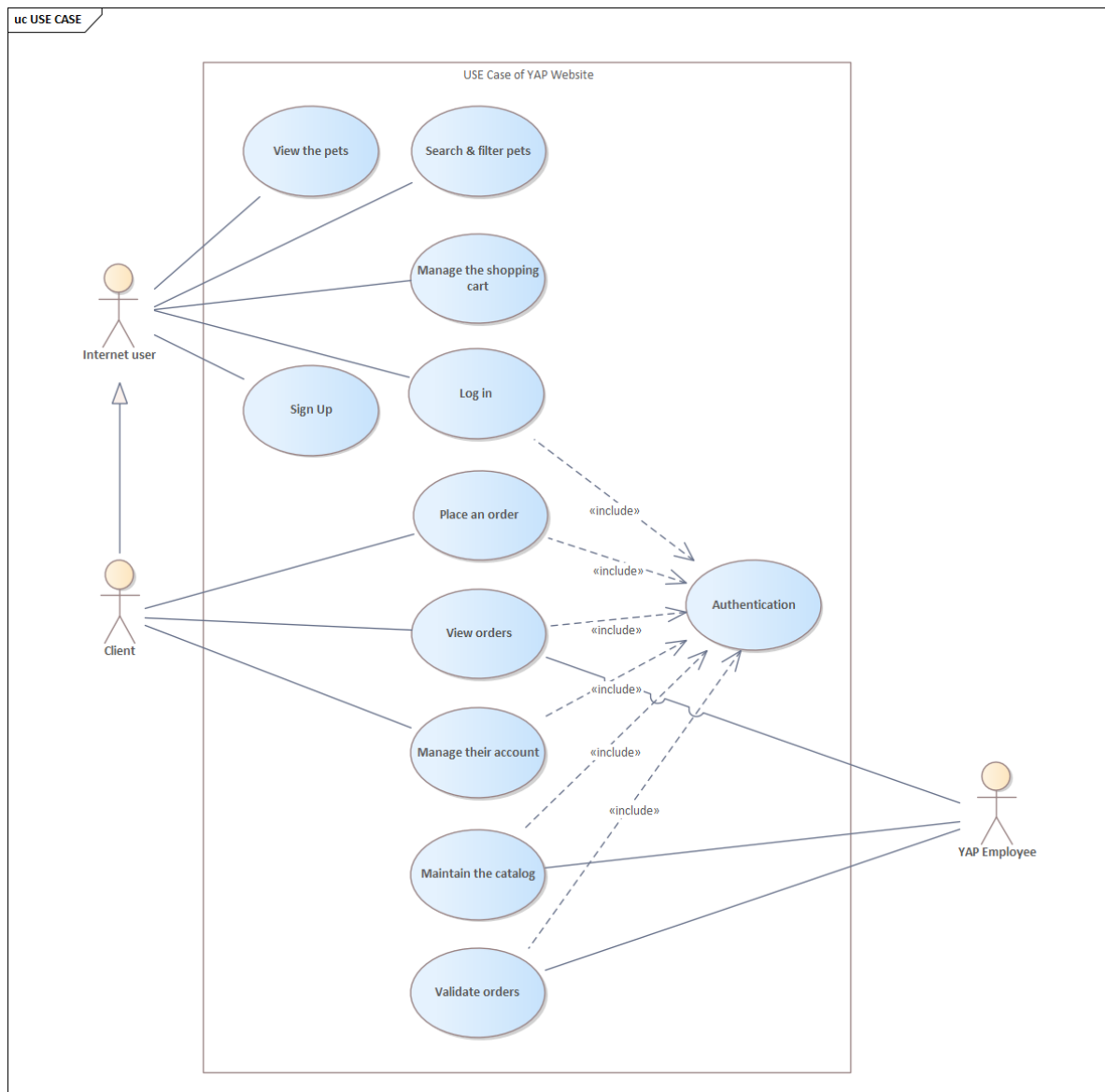
According to the Object Modelling Group (OMG), “modelling is **the designing of software applications before coding.**” In model-based software design and development, software modelling is used as an essential part of the software development process. In the next section, we use UML to describe the different structure and concepts of our Web application application before programming it.

### 1.2.2 Modelling Tool

Enterprise Architect is a comprehensive UML analysis and design tool for UML, SysML, BPMN and many other technologies. Covering software development from requirements gathering through to the analysis stages, design models, testing and maintenance. EA outfits your entire team, including analysts, testers, project managers, quality control staff and deployment team, for a fraction of the cost of some competing products. EA is a multi-user, Windows-based, graphical tool designed to help you build robust and maintainable software. It also features flexible and high quality documentation output.



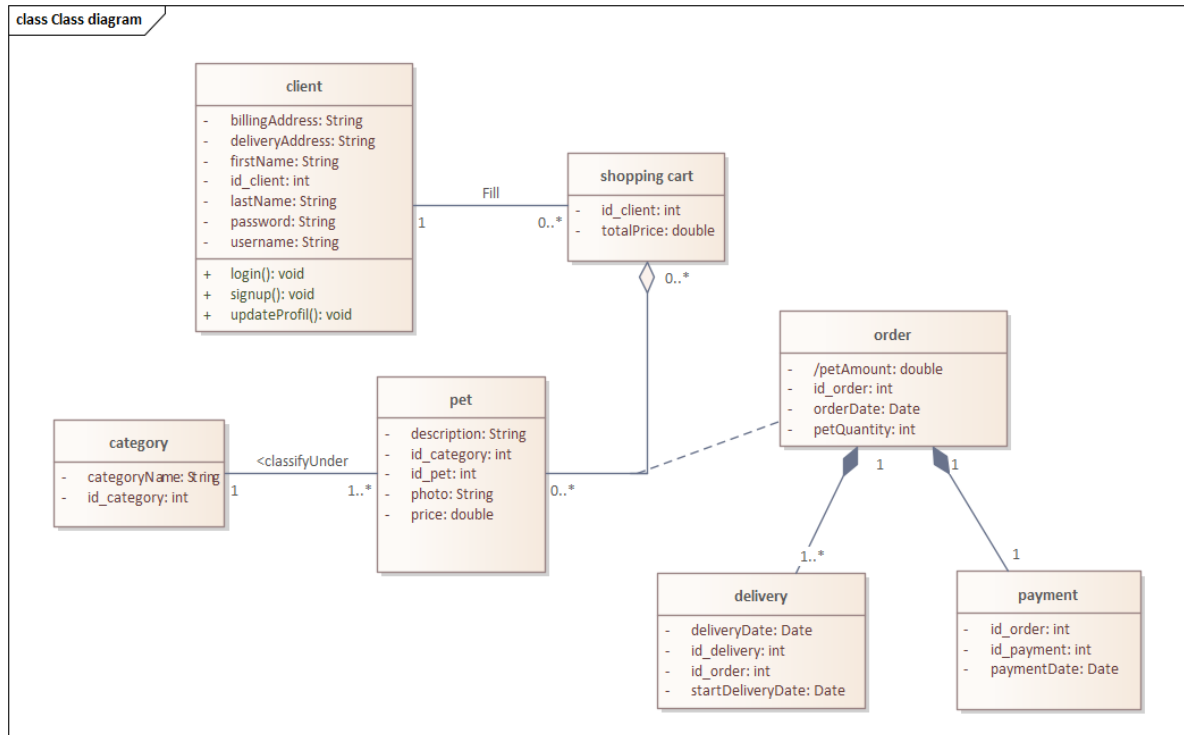
## 1.3 Use-Case Diagram



**Figure 2: Use Case diagram**

The Use Case Diagram given above shows a use case diagram is a graphic depiction of the interactions among the elements of a system. Use Case Diagram portrayed the system's functionality and necessity by using actors and use cases. It is represented by either circles or ellipses.

## 1.4 Class Diagram

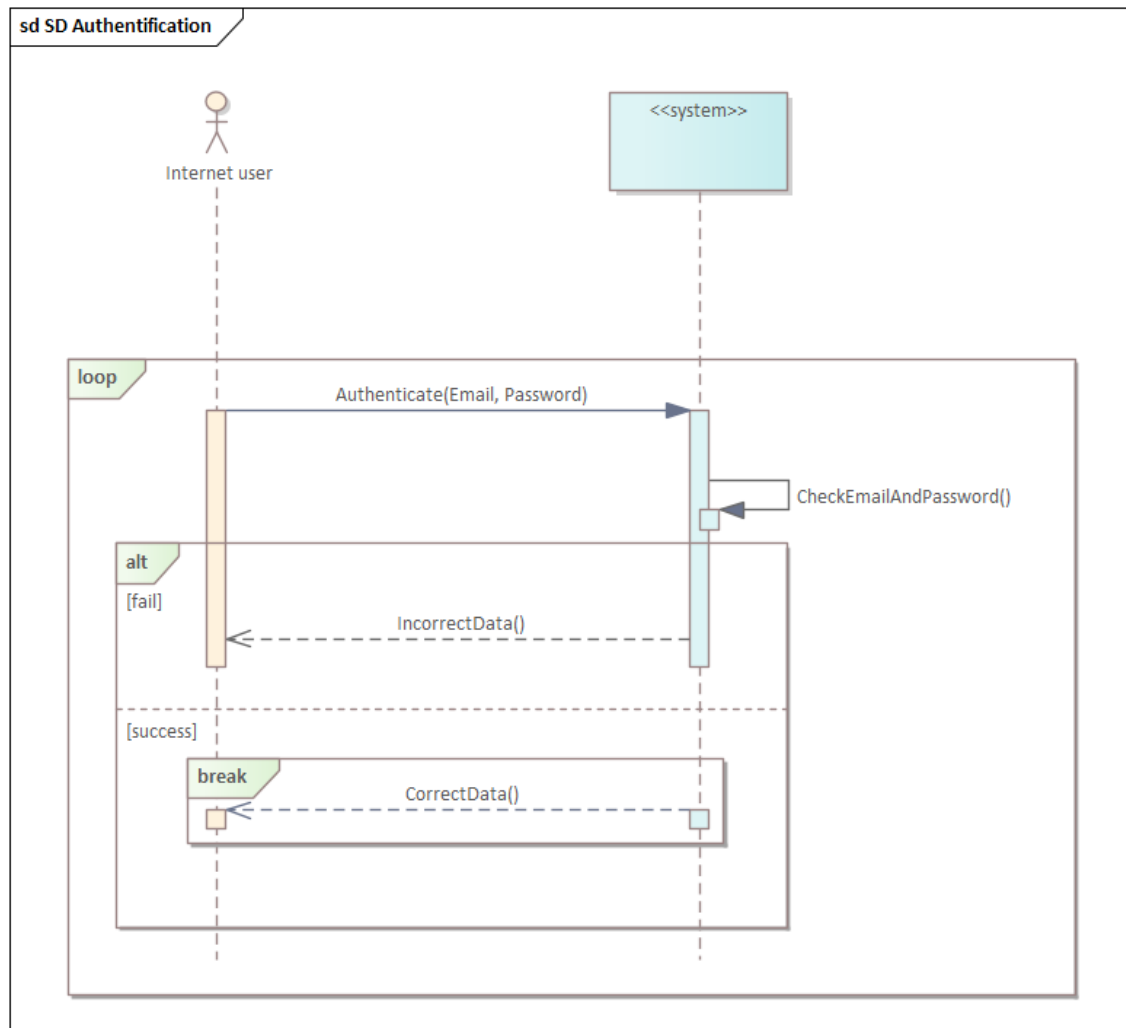


**Figure 3: Class diagram**

The class diagram is given above shows a class and the functions used in the project.

## 1.5 Sequence Diagrams

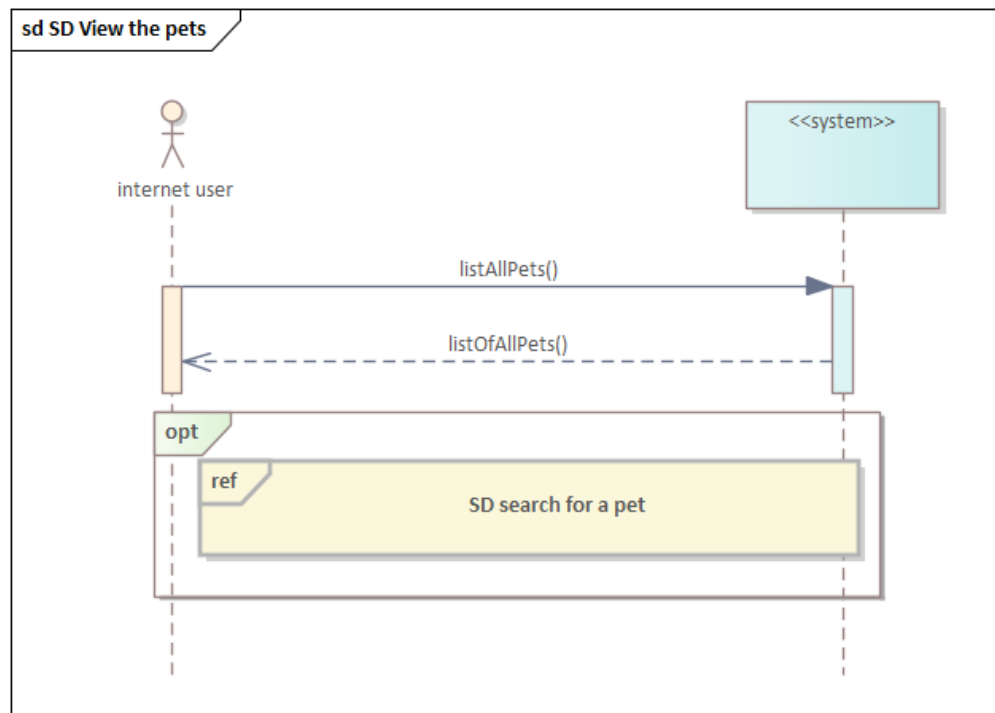
### 1.5.1 Authentication



**Figure 4: Sequence diagram – Authentication**

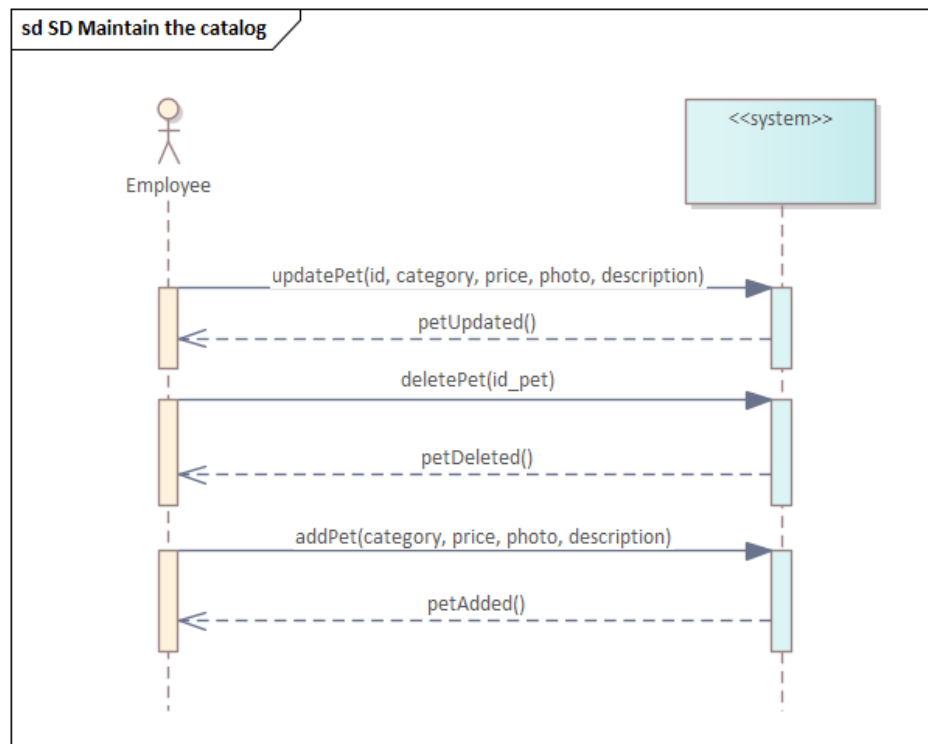


### 1.5.2 View the pets



**Figure 5: Sequence diagram - View the pets**

### 1.5.3 Maintain the catalog



**Figure 6: Sequence diagram - Maintain the catalog**

### 1.5.4 Create an account

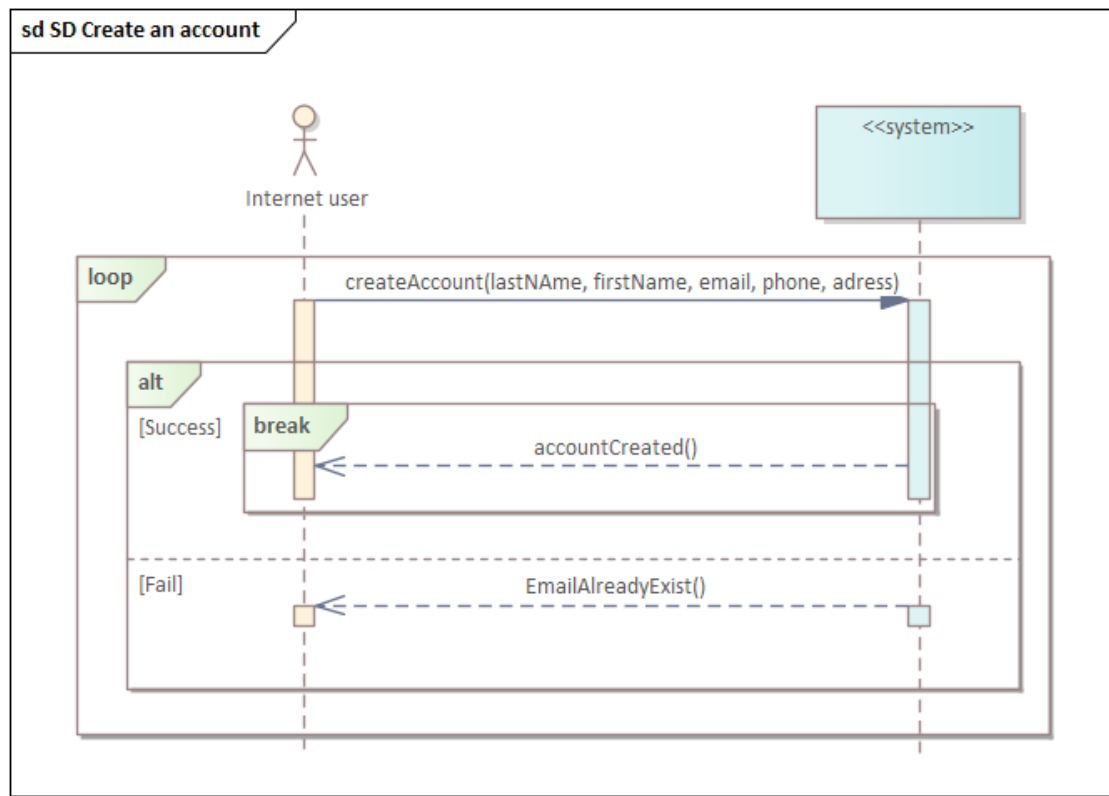
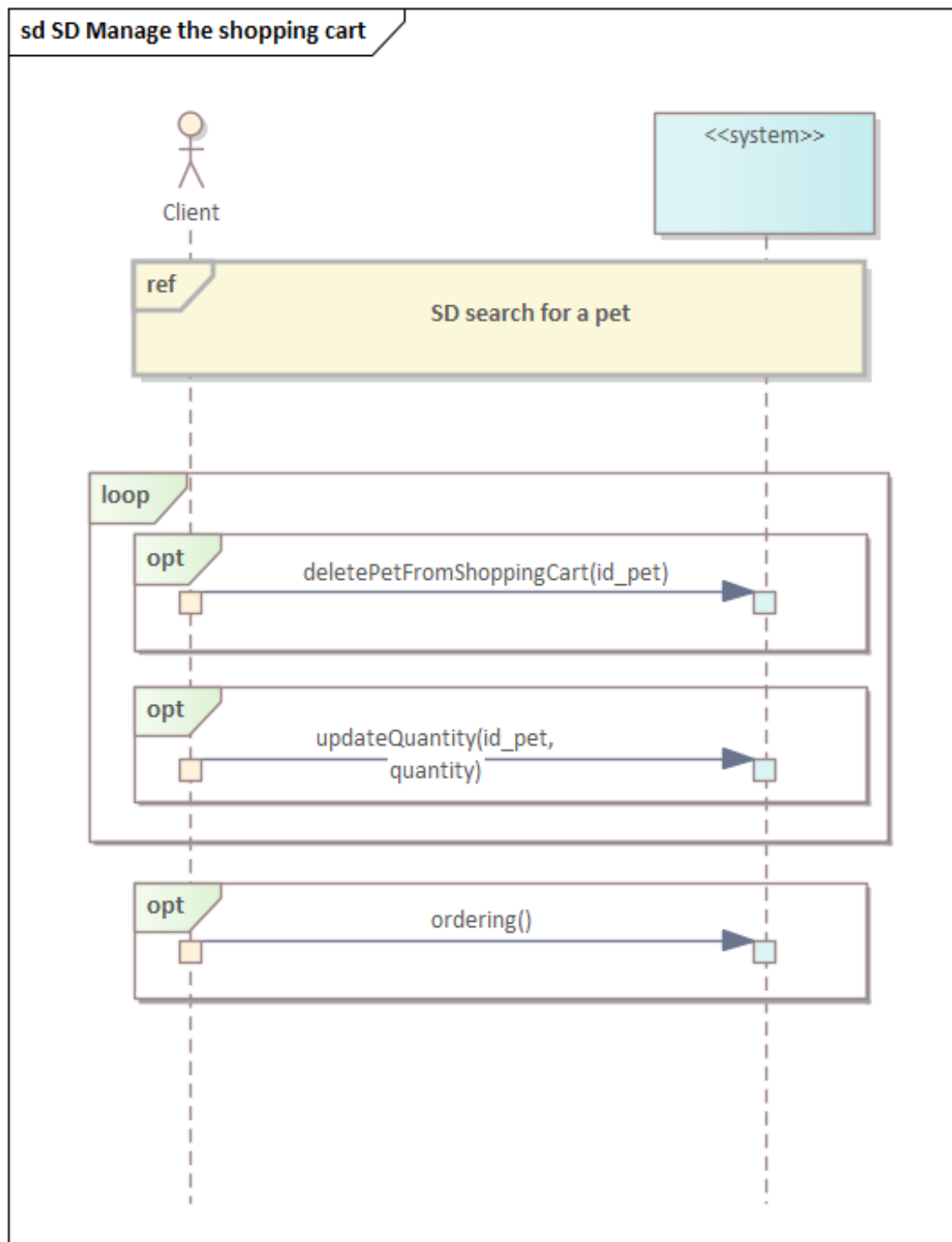


Figure 7: Sequence diagram - Create an account

### 1.5.5 Manage the shopping cart:



**Figure 8: Sequence diagram - Manage the shopping cart**

### 1.5.6 Search for a pet

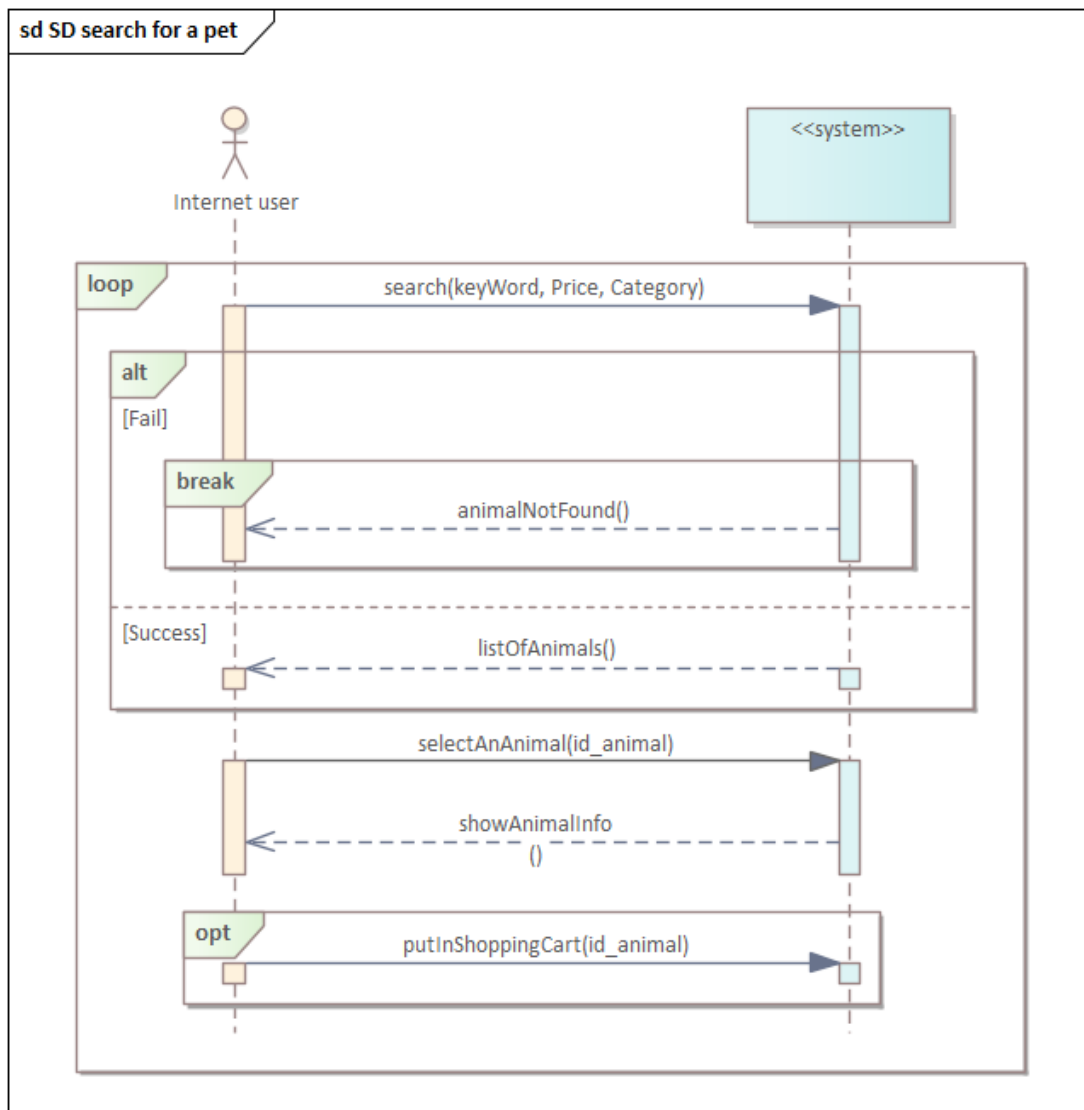


Figure 9: Sequence diagram - Search for a pet

### 1.5.7 Place an order

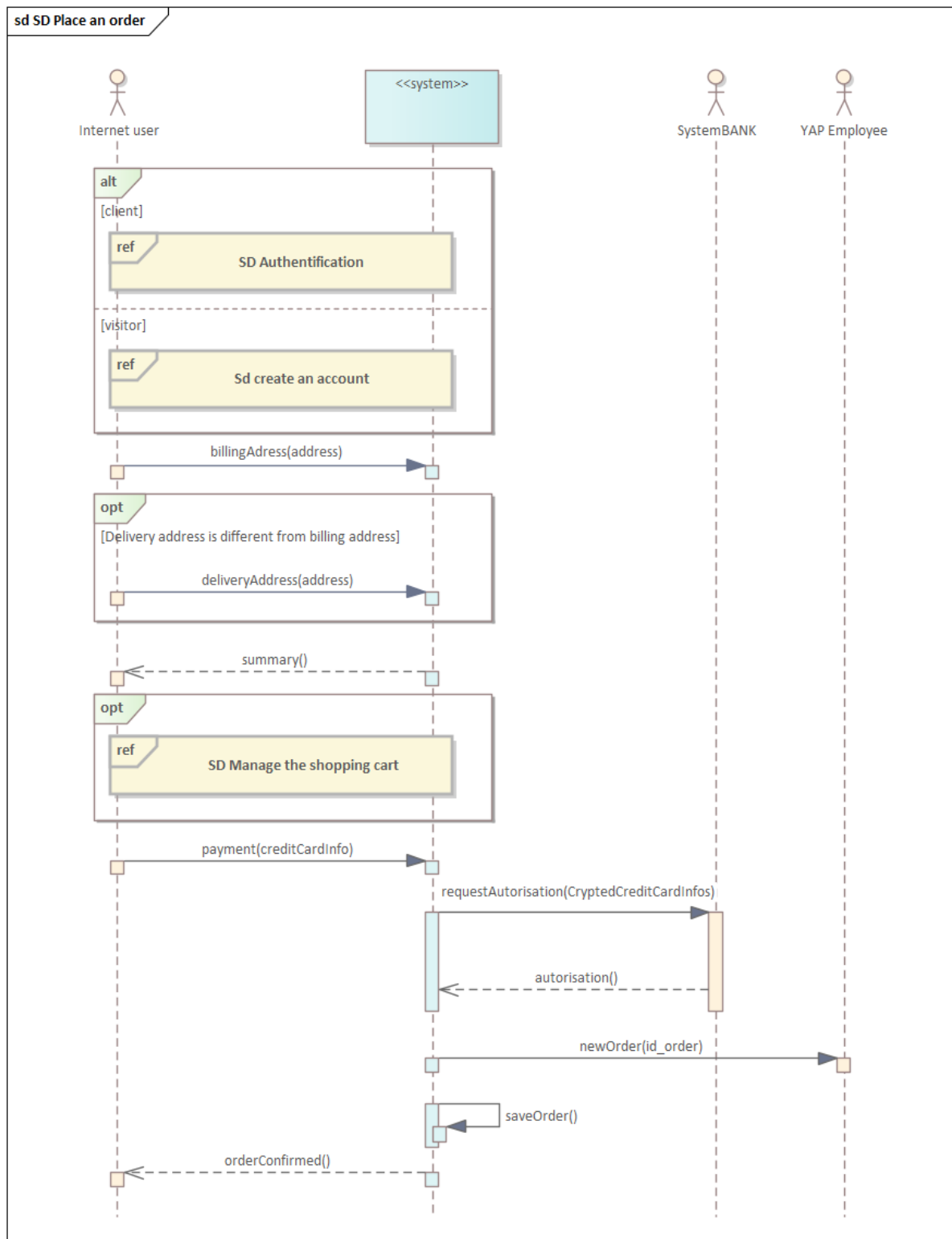


Figure 10: Sequence diagram - Place an order

## 1.6 Activity Diagram

### 1.6.1 Make an order

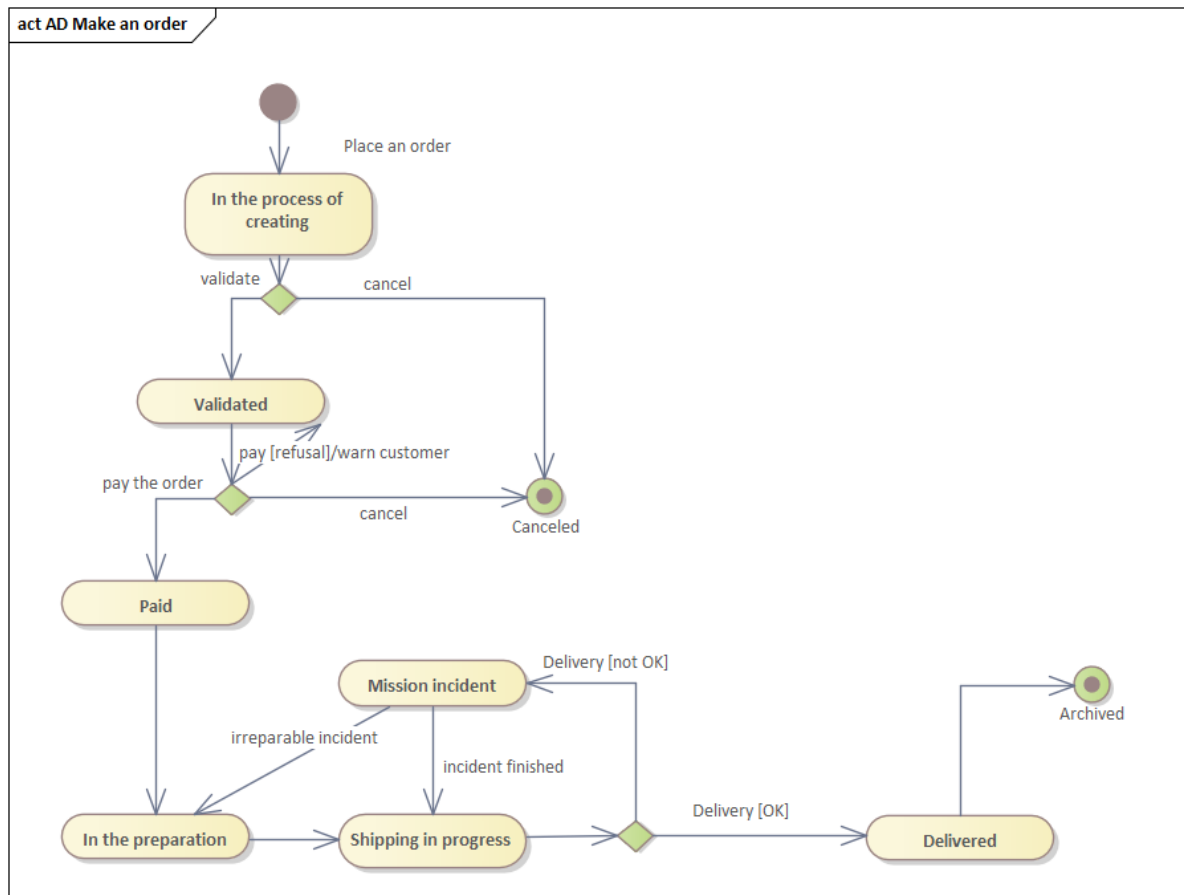


Figure 11: Activity diagram - Make an order

1.7 Global Interaction Diagram

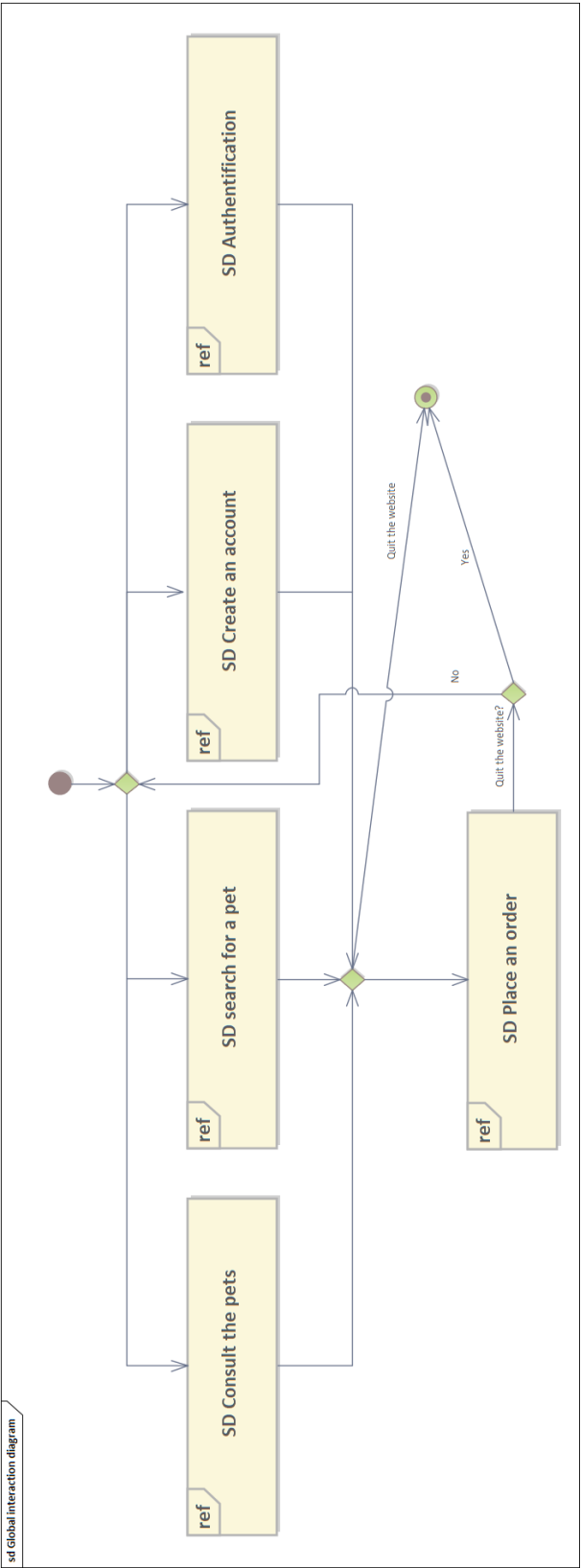


Figure 12: Global interaction diagram



## Chapter 2: Implementation

The implementation of the YAP Shop was a crucial part of the project, as it involved the development of the front-end, back-end, and database components of the system. In this section, the focus will be on showcasing how the different technologies were used to build the system and how the code was structured and organized. The implementation of the project was done keeping in mind the requirements and design that were previously defined. The use of React JS, Express JS, Node JS, and PostgreSQL allowed for the creation of a user-friendly, scalable, and secure e-commerce platform. The following sub-sections will provide a detailed explanation of the different components of the system and how they interact with each other to provide a seamless user experience.

### 2.1 Front-end development with React JS

#### 2.1.1 What is React?

React.js is an open-source JavaScript package that is used to make user interfaces for single-page apps. It takes care of the view layer for apps that run online and on phones. React also lets us make user interface parts that can be used more than once. React was first made by a software engineer at Facebook named Jordan Walke. React was first used in 2011 on Facebook's news feed and in 2012 on Instagram.com.



With React, developers can make large web apps that can change data without having to reload the page. React's main goal is to be fast, easy to use, and scalable. It only works on the user interfaces of the app. The view in the MVC template is the same as this. It can work with other libraries or frameworks written in JavaScript, like Angular JS in MVC. React is the shorter form of React JS.

### 2.1.2 React Features

These are the features that make the React library so good and strong for building modern apps, as well as what distinguishes React from competing frameworks and libraries:

#### Virtual DOM

DOM (Document Object Model) is the most essential component of web applications; it describes the HTML-based structure of a web page document. DOM manipulation is frequently utilized nowadays since contemporary applications demand a lot of state changes, animations, effects, and so on. Consider the following scenario: you have an application, and you want to update just the areas of your DOM tree that are affected by state changes. You don't want to re-render your whole UI from start because it would be expensive in terms of performance and user experience. React employs a feature known as virtual DOM, which is a virtual version of the real DOM tree. It's simply a tree data structure of basic JavaScript objects that is maintained in memory synchronized. Because the virtual DOM will never be presented to the user, it will only exist in memory, rendering it is quicker.

React builds a duplicate of your real DOM tree when your React app loads. Instead of re-rendering the whole real DOM tree whenever a state change occurs in your application, React first updates its virtual DOM with the changed state. React then compares its virtual DOM to your actual DOM tree to determine what needs to be modified. Then it modifies your actual DOM tree, but it just changes 7 the components that need to be altered. The virtual DOM is one of the aspects that contribute to the React framework's speed and dependability.

## **JSX**

JSX is an extension of JavaScript used by React. We utilize it to construct "elements" in React. Preprocessors (such as Babel) employ JSX to turn HTML like content in JavaScript files into parsable JavaScript objects. React does not mandate the use of JSX, however many developers find it beneficial for working with the UI within the JavaScript code. JSX is an essential component of React since it is used to generate React components.

## **Components**

React components are responsible for the reusability of our code and the segmentation of our UI into distinct parts. React components operate in the same method as JavaScript functions. A React component takes arbitrary inputs, called props, and must always return a React element indicating what should be presented to the user. If a React component fails to return a React element, it will issue an error. React components adhere to the separation of concerns design concept, which dictates that a computer program (in this example, our application) should be divided into distinct pieces, with each section addressing distinct issues.

React components are extremely effective because they enable us to write code that is clearer, more resilient, and more succinct throughout the whole application. We can have an unlimited number of React components.

## 2.2 Back-end development with Express JS and Node JS

Express JS and Node JS are popular JavaScript technologies that are widely used for building back-end web applications.

### 2.2.1 Express JS

Express JS is a minimal and flexible web application framework for Node.js. It provides a robust set of features for web and mobile applications, such as routing, middleware, and request handling. Express JS makes it easy to build API endpoints, which are necessary for a back-end system to communicate with the front-end.



### 2.2.2 Node JS

Node JS is a runtime environment that allows developers to run JavaScript code on the server-side. It is built on Chrome's V8 JavaScript engine and uses an event-driven, non-blocking I/O model. Node JS provides a fast and efficient environment for building scalable and high-performance applications, making it ideal for back-end development.



### 2.2.3 Why use Express JS & Node JS?

The use of Express JS and Node JS together makes it possible to create a full-stack JavaScript application. This allows for a seamless development experience, as the same language can be used on both the front-end and back-end. In addition, the large and active community around Node JS provides a wealth of resources, such as packages and modules, that can be used to add functionality and speed up development.

#### 2.2.1 PostgreSQL

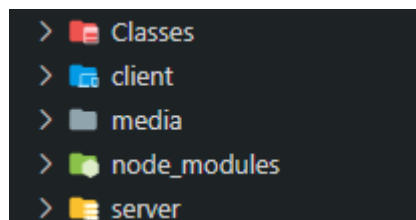
PostgreSQL is a powerful and open-source relational database management system (RDBMS) that is used as the database for the YAP Shop. It is well-suited for use in web applications due to its reliability, scalability, and support for complex data structures and transactions.

**In conclusion,** the choice of Express JS, Node JS, and PostgreSQL for the YAP Shop was driven by their suitability for building modern, scalable, and secure web applications.

## 2.3 Project Code & Structure

### 2.3.1 Project Code structure

In a typical web application development project, it is common to organize the code into different folders to keep it organized and maintain a clear separation of concerns. The four folders you mentioned, Classes, Client, Server, and node\_modules, serve specific purposes in the code structure.



**Figure 13: Project code structure**

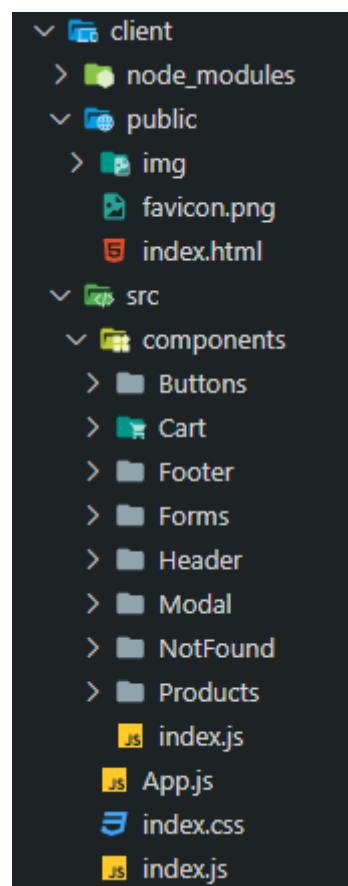
1. **Classes:** This folder contains class definitions and objects used in the back-end code. It may also contain helper functions and utility classes that are used across different parts of the application.
2. **Client:** This folder contains the front-end code of the application, built using React JS. This folder may contain subfolders for components, styles, and other assets necessary for the front-end.
3. **Server:** This folder contains the back-end code of the application, built using Express JS and Node JS. This folder may contain subfolders for controllers, models, and other components necessary for the back-end.

4. **node\_modules:** This folder contains the dependencies and packages that are used in the project. It is automatically generated when using the Node Package Manager (npm) to install packages. The packages in this folder are necessary for the application to run, and they can be updated or removed as needed.

These folders help to keep the code organized and maintain a clear separation of concerns. This makes the code more readable and maintainable, and makes it easier to add new features or make changes to the application in the future.

### 2.3.1 Front-end Code

The figure below shows how the frontend, represented by the client folder, is structured:



**Figure 14: Front-end code structure**

The client side of the YAP Shop is built using React JS, a JavaScript library for building user interfaces. React uses a component-based architecture, where the UI is broken down into reusable components that can be combined to form the complete UI. Each component has its own state, which represents the data that the component needs to render the UI. The state can be updated based on the user interactions or API responses.

The client side code communicates with the back-end through the API. For example, when the user wants to view the list of pets available in the YAP Shop, the client side code will make a GET request to the API endpoint `"/pets"`. The back-end will then respond with the data for the list of pets, which the client side code can then use to update the state of the component and render the UI.

The client side code also handles the user interactions, such as submitting a form to add a new pet to the YAP Shop. When the user submits the form, the client side code will make a POST request to the API endpoint `"/pets"` with the form data. The back-end will then process the request and store the new pet in the database. The client side code will then update the state of the component to show the updated list of pets, and the UI will be re-rendered to reflect the changes.

```
componentDidMount() {  
  Axios.get("/api/v1/products").then(({ data: { products } }) => {  
    this.setState({ products: products });  
    const max = Math.max.apply(  
      Math,  
      products.map(({ price }) => price)  
    );  
    this.setState({ priceRange: [0, max] });  
  });  
}
```

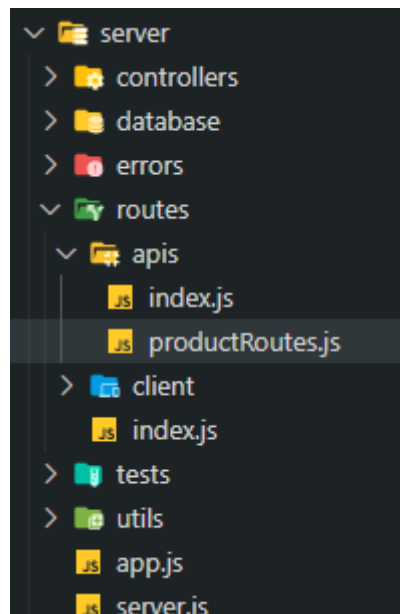
**Figure 15: GET pets request to the API**



**In conclusion**, the client side code in the YAP Shop is responsible for rendering the UI, handling user interactions, and making API requests to the back-end. The component-based architecture of React allows for a modular and maintainable code structure, while the communication with the back-end through the API allows for a dynamic and responsive user experience.

### 2.3.1 Back-end Code

The figure below shows how the frontend, represented by the client folder, is structured:



**Figure 16: Back-end code structure**

The back-end code provides the API endpoints for the client side to communicate with. For example, the endpoint `"/pets"` allows the client side to retrieve the list of pets available in the YAP Shop, or add a new pet to the YAP Shop. The back-end code also interacts with the PostgreSQL database to store and retrieve the data for the pets.

```
const router = require('express').Router();
const {
  getAllProducts,
  getSingleProduct,
  createProduct,
  deleteProduct,
  updateProduct,
} = require('../controllers');

router.get('/', getAllProducts);
router.post('/', createProduct);
router.get('/product/:id', getSingleProduct);
router.delete('/product/:id', deleteProduct);
router.patch('/product/:id', updateProduct);

module.exports = router;
```

**Figure 17: Pet data routes**

Express JS provides middleware functions, which can be used to process the incoming API requests and handle various tasks such as parsing the request body, validating the request data, or applying authentication and authorization. The back-end code can also define error-handling middleware functions to handle any errors that may occur during the processing of the API requests.

Node JS provides a number of built-in modules and packages, such as the HTTP module and the Sequelize ORM, which can be used to implement the back-end functionality. The HTTP module can be used to create the API server and handle incoming API requests, while the Sequelize ORM can be used to interact with the PostgreSQL database and perform operations such as creating tables, inserting data, or querying data.

**In conclusion,** the back-end code in the YAP Shop provides the API endpoints for the client side to communicate with and handles the interactions with the database. The combination of Express JS and Node JS provides a robust and flexible back-end platform for building web applications, with a rich set of features for processing API requests and interacting with databases.

## Chapter 3: Web Application Preview

### 3.1 Home Page

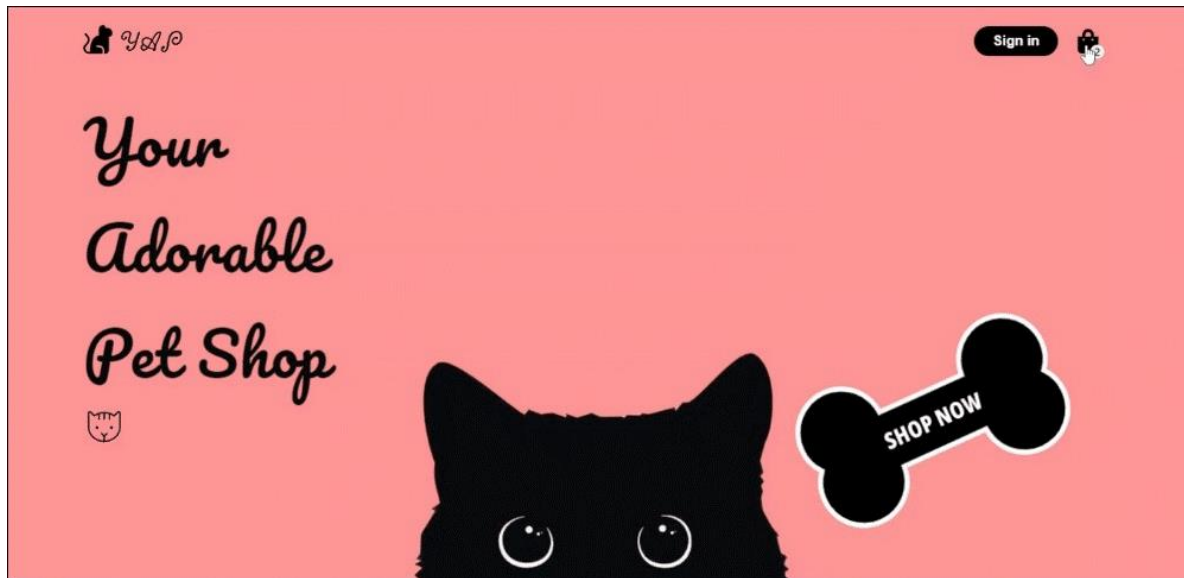


Figure 18: Home Page preview

### 3.2 Pet Categories

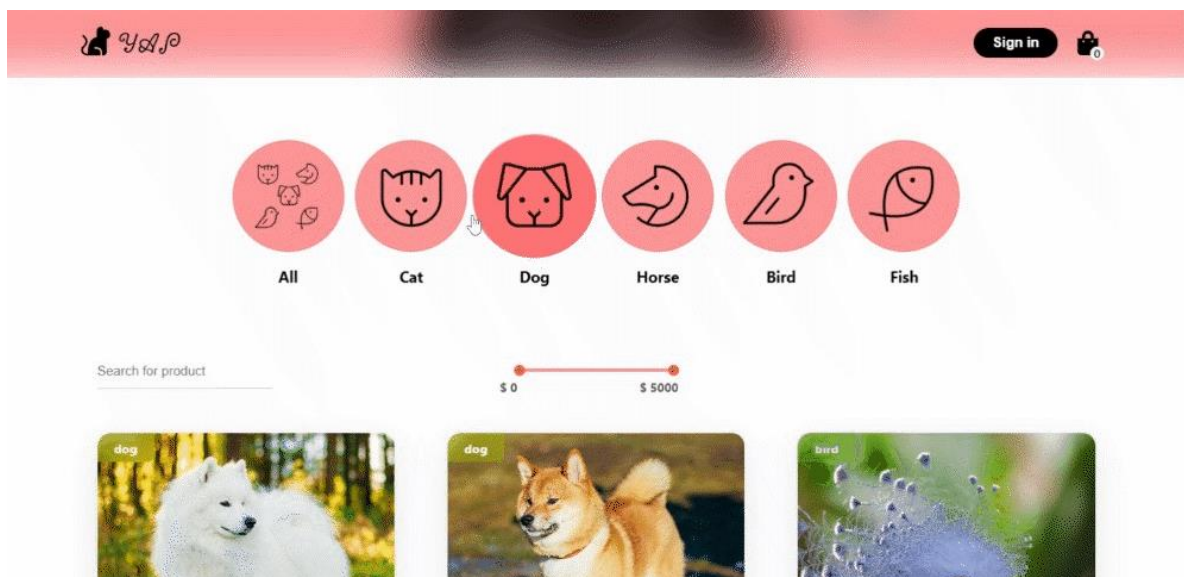


Figure 19: Pet Categories preview

### 3.3 Pet Catalog

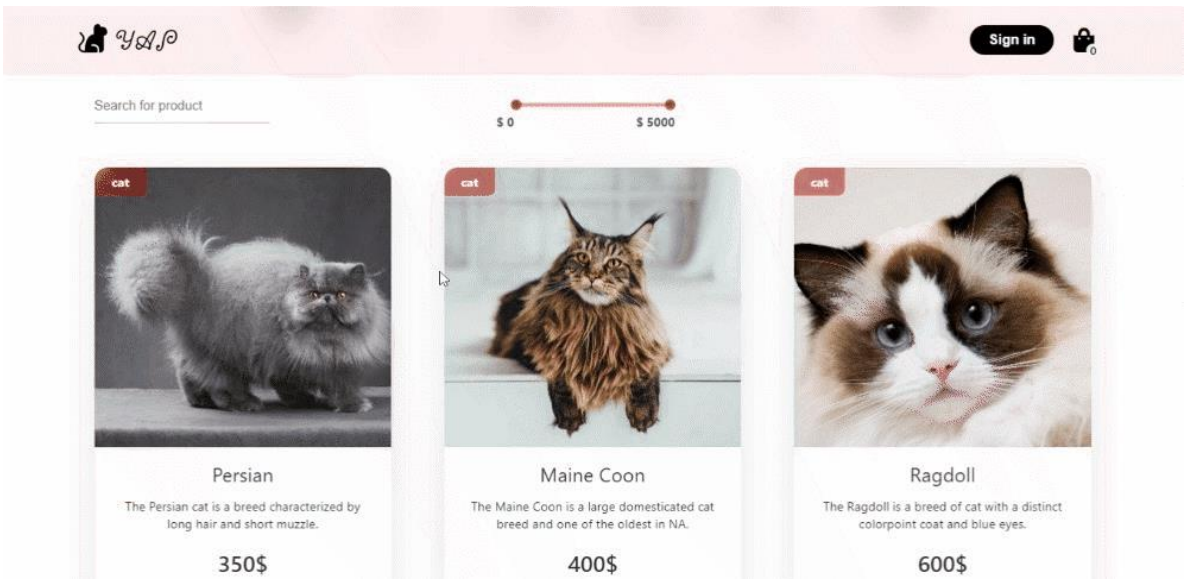


Figure 20: Pet Catalog preview

### 3.4 Pet Search

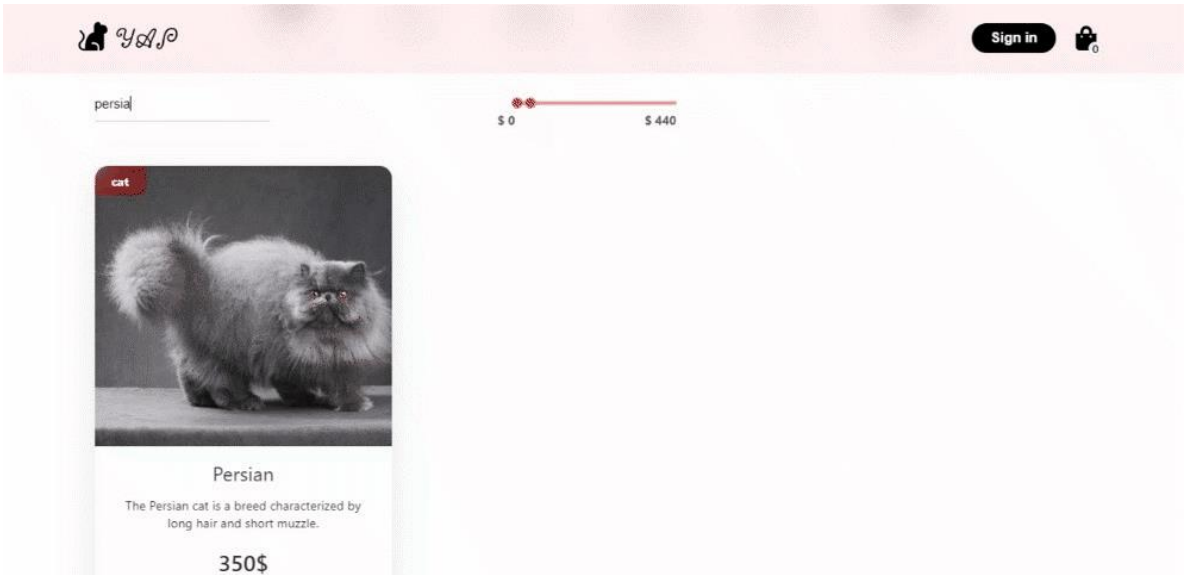


Figure 21: Pet Search preview

## 3.5 Pet Details Page

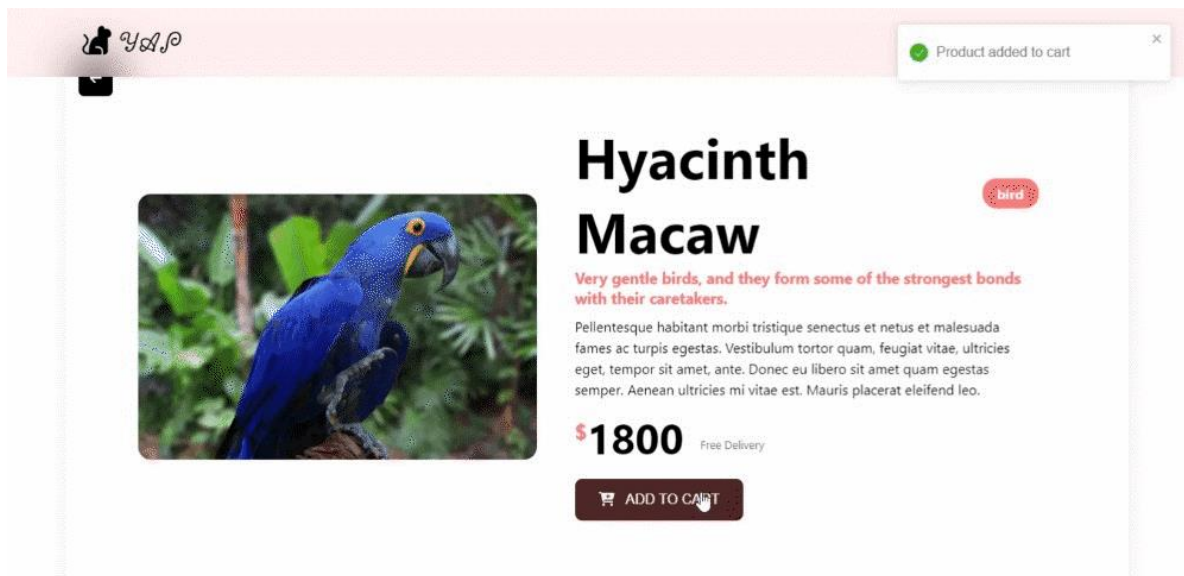


Figure 22: Pet Details Page preview

## 3.6 Shopping Cart

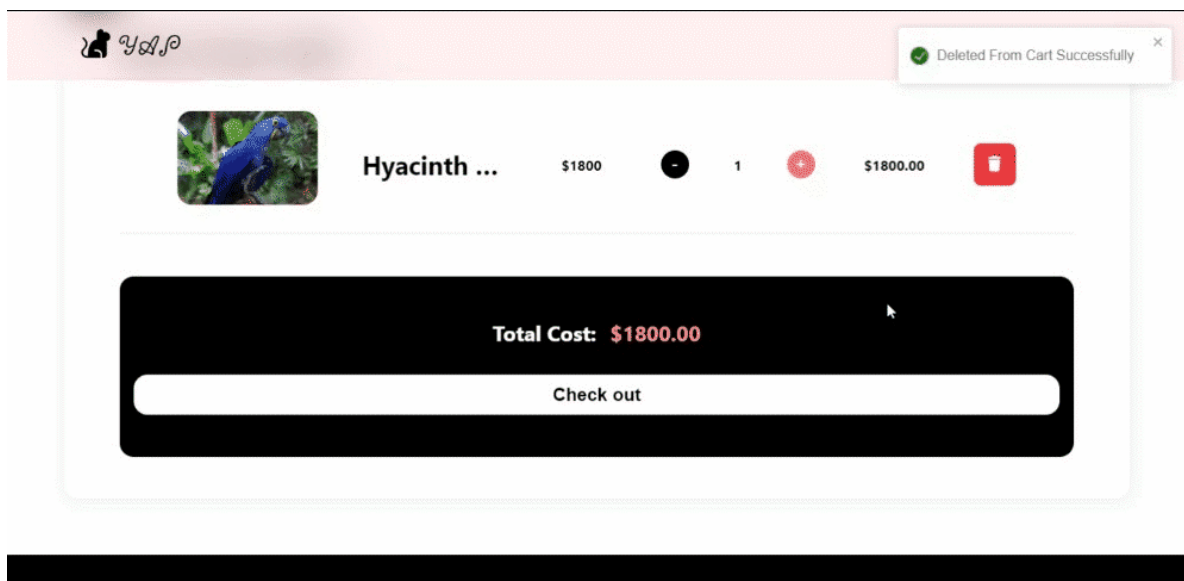


Figure 23: Shopping Cart preview

## 3.7 Login

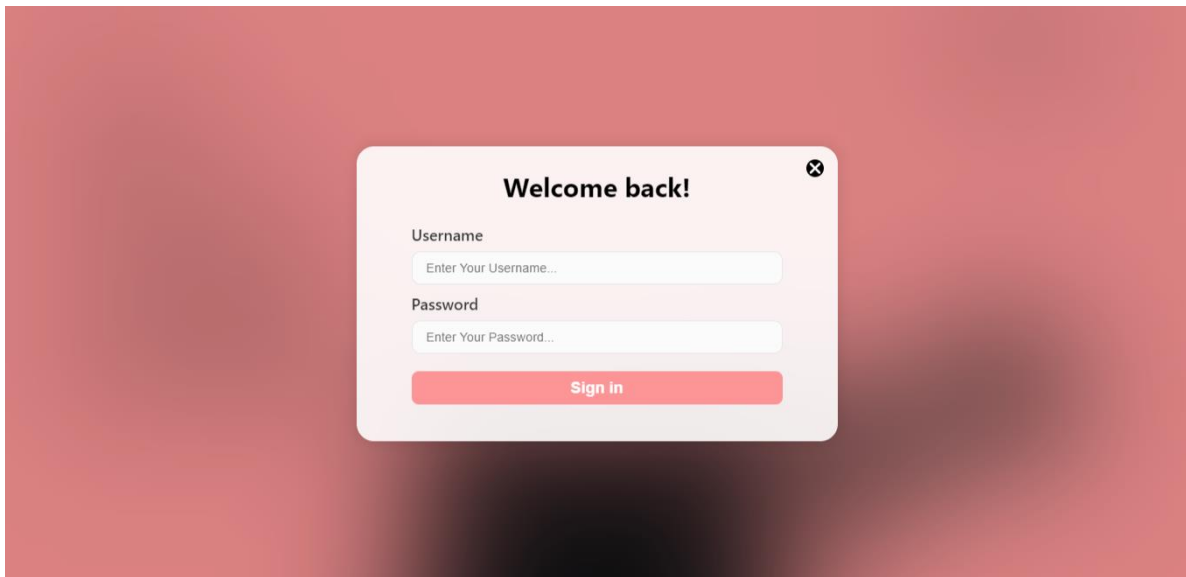


Figure 24: Login preview

## 3.8 Add New Pet

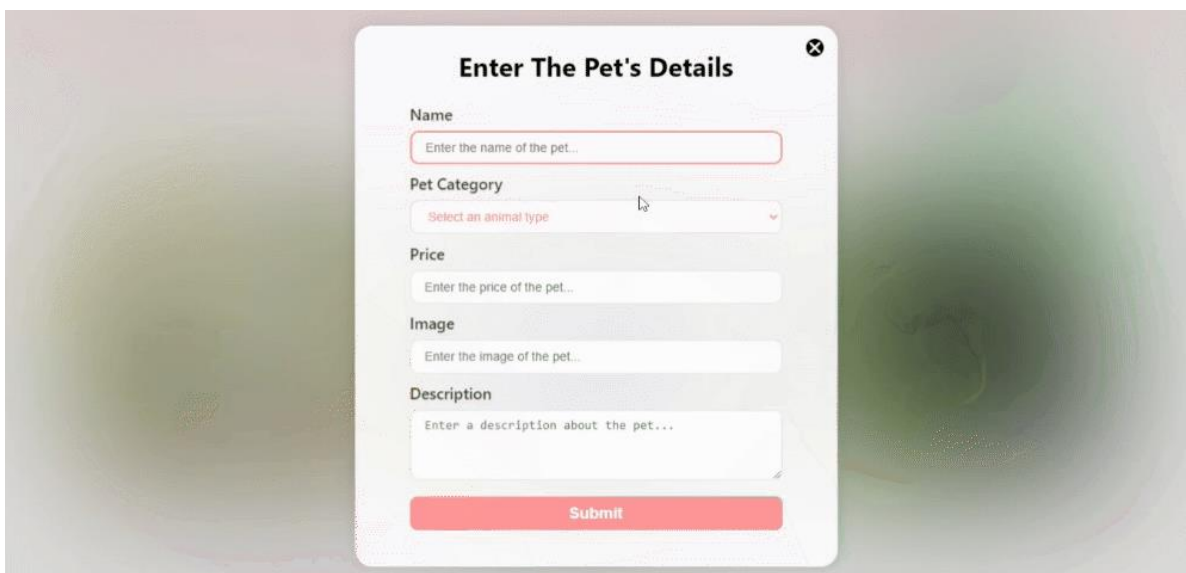


Figure 25: Add New Pet preview

## 3.9 Update Pet Details

**Update Pet Details**

Name  
Black Palm Cockatoo

Pet Category  
bird

Price  
100

Image  
<https://www.thehappychickencoop.com/wp-content/uploads/2/>

Description  
These birds have the ability to use tools like large sticks in drumming against a hollow tree.

Update

Please fill all fields Correctly

Figure 26: Update Pet Details preview

## Conclusions & Perspectives

The YAP Shop project has been successfully implemented as an online pet shop that sells various pets. The project showcases the capabilities of React JS, Express JS, Node JS, and PostgreSQL in building a modern, scalable, and secure e-commerce platform. The project has been designed with a user-centered approach, providing a rich and intuitive user experience that is both attractive and easy to use.

The implementation of the project demonstrates a clear understanding of the technologies used and the architecture of a modern web application. The use of Express JS as the web framework and Node JS as the runtime environment provides a flexible and scalable back-end functionality. The integration of PostgreSQL as the database management system provides a robust and secure data management solution, capable of handling complex data structures and transactions.

The project also showcases the advantages of using modern front-end technologies such as React JS, which provides a dynamic and responsive user interface, and ensures a seamless and engaging user experience. The code structure of the project is well-organized, making it easy to maintain and expand upon in the future.

In conclusion, the YAP Shop project is a successful implementation of a modern e-commerce platform that meets the needs of a pet shop. The use of the latest web technologies has provided a robust and scalable solution that is both secure and user-friendly. The project demonstrates the capability of the team to develop a high-quality and well-designed web application, and it has the potential for future growth and expansion.