Available online www.jsaer.com

Journal of Scientific and Engineering Research, 2024, 11(6):127-133

Research Article

ISSN: 2394-2630
CODEN(USA): JSERBR

Building a Shipping Web Application with .NET and React

Naga Lalitha Sree Thatavarthi

Email: thatavarthinagalalithasree2020@gmail.com

Abstract This paper presents a shipping application that uses dot and react as the main technologies. The application allows users to create, track, and manage their shipments online. The paper describes the design, implementation, and evaluation of the application, as well as the challenges and benefits of using dot and react. The paper also discusses the future work and possible improvements for the application.

Keywords Shipping application, .NET, React, web development, backend, frontend, software architecture, shipment tracking.

1. Introduction

Shipping is an essential service that connects businesses and consumers across the world. However, shipping can also be a complex and costly process that involves multiple parties, regulations, and risks. Therefore, there is a need for efficient and user-friendly shipping applications that can simplify and streamline the shipping process.

One of the challenges of developing a shipping application is to choose the appropriate technologies that can support the functionality, performance, and usability of the application. There are many technologies available for web development, but not all of them are suitable for a shipping application. In this paper, we propose a shipping application that uses dot and react as the main technologies. Dot is a web framework that allows developers to create dynamic web pages using C# and Razor syntax. React is a JavaScript library that allows developers to create user interfaces using components and state management. We argue that dot and react are suitable technologies for a shipping application because they offer the following advantages:

- Dot provides a server-side rendering that can improve the performance and security of the application. Dot also supports dependency injection, data validation, and authentication, which can simplify the development and maintenance of the application.
- React provides a client-side rendering that can enhance the interactivity and responsiveness of the application. React also supports reusable components, hooks, and context, which can facilitate the development and testing of the application.

The paper is organized as follows. Section II describes the design of the shipping application, including the architecture, data model, and user interface. Section III describes the implementation of the shipping application, including the technologies, tools, and libraries used. Section IV describes the evaluation of the shipping application, including the functionality, performance, and usability tests. Section V discusses the challenges and benefits of using dot and react for the shipping application. Section VI discusses the future work and possible improvements for the shipping application. Section VII concludes the paper.

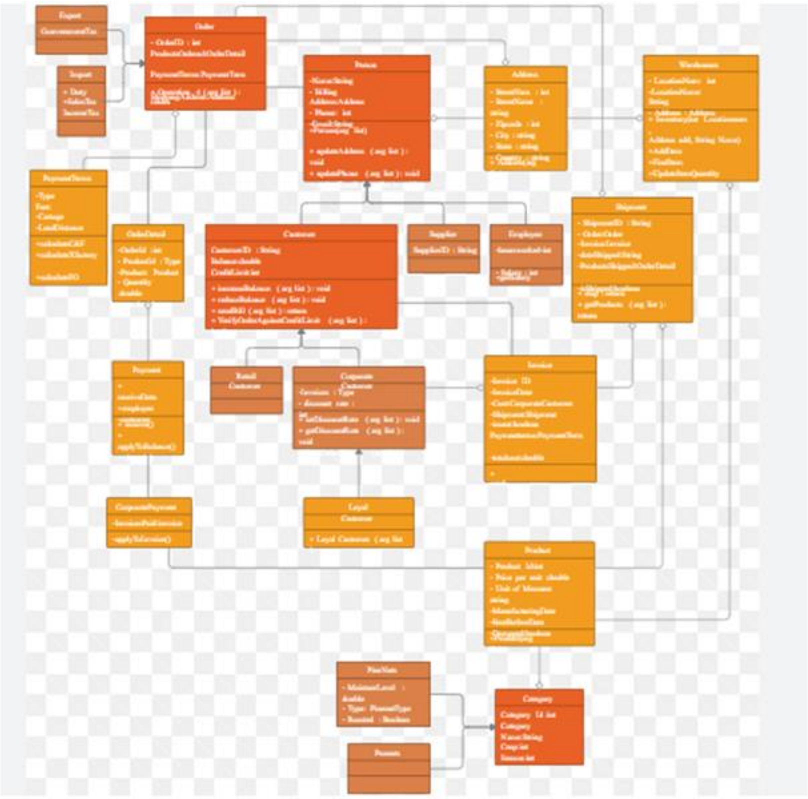
2. Design

In this section, we describe the design of the shipping application, including the architecture, data model, and user interface.

3. Architecture

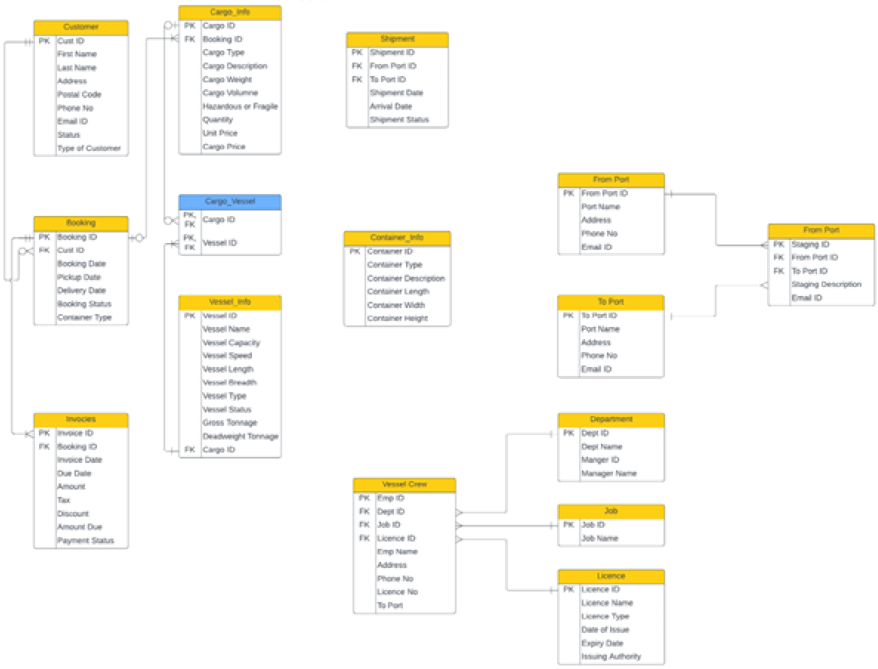
The shipping application follows a three-tier architecture, which consists of the presentation layer, the business layer, and the data layer. The presentation layer is responsible for displaying the user interface and interacting with the user. The business layer is responsible for implementing the business logic and processing the user requests. The data layer is responsible for storing and retrieving the data from the database.

The presentation layer uses React as the main technology, which allows the creation of user interfaces using components and state management. The presentation layer communicates with the business layer using RESTful APIs, which are implemented using dot. The business layer uses dot as the main technology, which allows the creation of dynamic web pages using C# and Razor syntax. The business layer communicates with the data layer using Entity Framework Core, which is an object-relational mapper that enables the manipulation of data using C# objects. The data layer uses SQL Server as the database, which stores the data in tables and relations.



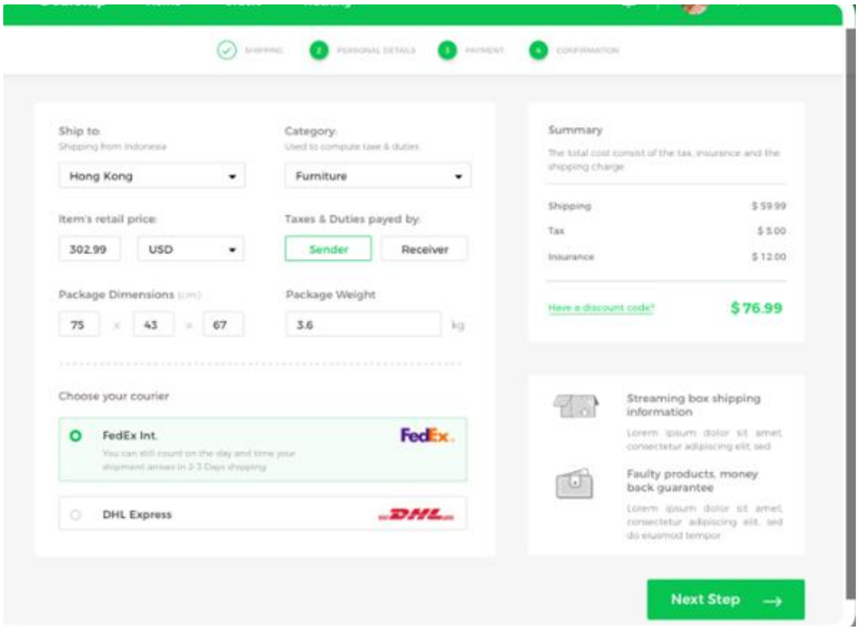
4. Data Model

The data model of the shipping application consists of four main entities: User, Shipment, Package, and Tracking. A User represents a registered user of the application, who can create, track, and manage their shipments. A Shipment represents a shipment that a user creates, which consists of one or more packages. A Package represents a package that belongs to a shipment, which has a weight, a dimension, and a destination. A Tracking represents a tracking event that occurs during the shipment process, which has a date, a time, a location, and a status.



5. User Interface

The user interface of the shipping application consists of four main pages: Home, Create, Track, and Manage. The Home page is the landing page of the application, which displays the logo, the navigation bar, and the introduction of the application. The Create page is the page where the user can create a new shipment, which requires the user to enter the package details, the destination address, and the payment method. The Track page is the page where the user can track the status of their shipments, which requires the user to enter the shipment number or scan the QR code. The Manage page is the page where the user can manage their shipments, which allows the user to edit, cancel, or delete their shipments.



Ship to
Shipping from Indonesia
Hong Kong

Category
Used to compute taxe & duties
Furniture

Item's retail price
302.99 USD

Taxes & Duties payed by
Sender Receiver

Package Dimensions (cm)
75 x 43 x 67

Package Weight
3.6 kg

Choose your courier

● FedEx Int. FedEx.
You can still count on the day and time your shipment arrives in 2-3 Days shipping

○ DHL Express DHL

Summary
The total cost consist of the tax, insurance and the shipping charge

Shipping	$ 59.99
Tax	$ 5.00
Insurance	$ 12.00
Have a discount code?	$ 76.99

Streaming box shipping information
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed

Faulty products, money back guarantee
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor

6. Implementation

In this section, we describe the implementation of the shipping application, including the technologies, tools, and libraries used.

7. Technologies

The shipping application uses the following technologies:

- Dot Net: A web framework that allows developers to create dynamic web pages using C# and Razor syntax. Dot provides a server-side rendering that can improve the performance and security of the application. Dot also supports dependency injection, data validation, and authentication, which can simplify the development and maintenance of the application.
- React: A JavaScript library that allows developers to create user interfaces using components and state management. React provides a client-side rendering that can enhance the interactivity and responsiveness of the application. React also supports reusable components, hooks, and context, which can facilitate the development and testing of the application.
- Entity Framework Core: An object-relational mapper that enables the manipulation of data using C# objects. Entity Framework Core supports various database providers, including SQL Server, which is used for the shipping application. Entity Framework Core also supports migrations, which can automate the creation and update of the database schema.
- SQL Server: A relational database management system that stores the data in tables and relations. SQL Server supports various data types, constraints, indexes, and queries, which can ensure the integrity, efficiency, and flexibility of the data.
- RESTful APIs: A set of conventions that define how the presentation layer and the business layer communicate. RESTful APIs use HTTP methods, such as GET, POST, PUT, and DELETE, to perform operations on resources, such as shipments, packages, and trackings. RESTful APIs also use JSON as the data format, which is easy to parse and manipulate.

8. Tools

The shipping application uses the following tools:

- Visual Studio: An integrated development environment that supports various programming languages, including C# and JavaScript. Visual Studio provides various features, such as code editing, debugging, testing, and publishing, which can assist the development and deployment of the application.
- Visual Studio Code: A code editor that supports various programming languages, including C# and JavaScript. Visual Studio Code provides various extensions, such as dot, React, and ESLint, which can enhance the functionality and quality of the code.
- SQL Server Management Studio: A graphical tool that allows the management and administration of SQL Server databases. SQL Server Management Studio provides various features, such as query execution, data manipulation, backup and restore, and security configuration, which can assist the data layer of the application.
- Postman: A tool that allows the testing and debugging of RESTful APIs. Postman provides various features, such as request creation, response validation, parameterization, and documentation, which can assist the business layer of the application

9 .Libraries

The shipping application uses the following libraries:

- Bootstrap: A CSS framework that provides various styles and components, such as buttons, forms, tables, and modals, which can improve the appearance and functionality of the user interface.
- React Bootstrap: A React library that integrates Bootstrap components into React components, which can simplify the development and rendering of the user interface.
- React Router: A React library that provides routing functionality for the user interface, which can enable the navigation and transition between different pages.
- Axios: A JavaScript library that provides HTTP functionality for the user interface, which can enable the communication and data exchange with the RESTful APIs.
- QRCode: A JavaScript library that provides QR code functionality for the user interface, which can enable the generation and scanning of QR codes for the shipments.

10 .Evaluation

In this section, we describe the evaluation of the shipping application, including the functionality, performance, and usability tests.

11. Functionality

The functionality test aims to verify that the shipping application meets the functional requirements and specifications. The functionality test consists of the following steps:

1. Create a test plan that defines the test cases, the expected results, and the acceptance criteria for each functionality of the application.
2. Execute the test cases using the shipping application and record the actual results and any errors or defects.
3. Compare the actual results with the expected results and determine if the test cases pass or fail.
4. Report the test results and any errors or defects to the developers and stakeholders.
5. Fix any errors or defects and repeat the test cases until all the test cases pass.

The functionality test covers the following functionalities of the application:

- User registration and login
- Shipment creation and payment
- Shipment tracking and QR code
- Shipment management and modification
- RESTful APIs and data validation
- Database operations and data integrity

12. Performance

The performance test aims to measure the performance and scalability of the shipping application under various load and stress conditions. The performance test consists of the following steps:

1. Create a test plan that defines the performance metrics, the load scenarios, and the success criteria for each performance aspect of the application.
2. Execute the load scenarios using a load testing tool, such as JMeter, and monitor the performance metrics, such as response time, throughput, and error rate.
3. Analyze the performance metrics and identify any performance bottlenecks or issues.
4. Report the performance results and any performance bottlenecks or issues to the developers and stakeholders.
5. Optimize any performance bottlenecks or issues and repeat the load scenarios until the performance metrics meet the success criteria.

The performance test covers the following performance aspects of the application:

- Server-side rendering and caching
- Client-side rendering and state management
- RESTful APIs and data exchange
- Database queries and data manipulation
- Concurrency and scalability

13. Usability

The usability test aims to evaluate the usability and user satisfaction of the shipping application from the user's perspective. The usability test consists of the following steps:

1. Create a test plan that defines the usability goals, the user scenarios, and the feedback methods for each usability aspect of the application.
2. Recruit a representative sample of users who match the target user profile of the application.
3. Conduct the usability test with the users using the shipping application and collect the feedback using methods such as observation, interview, questionnaire, and rating scale.

4. Analyze the feedback and identify any usability issues or suggestions.
5. Report the usability results and any usability issues or suggestions to the developers and stakeholders.
6. Improve any usability issues or suggestions and repeat the usability test until the usability goals are achieved.

The usability test covers the following usability aspects of the application:

- User interface design and layout
- User interface interactivity and responsiveness
- User interface clarity and consistency
- User experience and satisfaction
- User feedback and improvement

14. Discussion

In this section, we discuss the challenges and benefits of using dot and react for the shipping application.

15 Challenges

The challenges of using dot and react for the shipping application are as follows:

- Learning curve: Dot and react are relatively new technologies that require a certain level of knowledge and experience to use effectively. The developers need to learn the syntax, concepts, and best practices of dot and react, as well as the integration of the two technologies.
- Compatibility: Dot and react are not fully compatible with each other, as they use different rendering methods and data formats. The developers need to use tools and libraries, such as dot SPA Services and JSON.NET, to enable the communication and data exchange between the two technologies.
- Debugging: Dot and react are not easy to debug, as they involve multiple layers and components. The developers need to use tools and techniques, such as Visual Studio debugger, React developer tools, and console.log, to identify and fix the errors or defects in the code.

16. Benefits

The benefits of using dot and react for the shipping application are as follows:

- Performance: Dot and react can improve the performance of the shipping application, as they use server-side and client-side rendering respectively. Dot can reduce the loading time and bandwidth consumption of the application, as it renders the initial HTML on the server. React can improve the interactivity and responsiveness of the application, as it renders the subsequent updates on the client.
- Simplicity: Dot and react can simplify the development and maintenance of the shipping application, as they use dependency injection, data validation, and authentication for dot, and reusable components, hooks, and context for react. These features can reduce the complexity and redundancy of the code, as well as the coupling and dependency of the components.
- Flexibility: Dot and react can provide flexibility for the shipping application, as they support various technologies, tools, and libraries. Dot can work with various database providers, such as SQL Server, MySQL, and MongoDB. React can work with various user interface libraries, such as Bootstrap, Material UI, and Ant Design.

17. Future Work

In this section, we discuss the future work and possible improvements for the shipping application.

Features

The future features for the shipping application are as follows:

- Notification: The shipping application can provide notification functionality, which can send email or SMS alerts to the users when their shipments are updated or delivered.
- Review: The shipping application can provide review functionality, which can allow the users to rate and comment on their shipments and the service quality.

- Analytics: The shipping application can provide analytics functionality, which can collect and visualize the data and statistics of the shipments, such as the number, the destination, the duration, and the status.

18. Improvements

The possible improvements for the shipping application are as follows:

- Security: The shipping application can improve the security of the application, by using HTTPS, encryption, and hashing to protect the data and communication.
- Accessibility: The shipping application can improve the accessibility of the application, by using color contrast, font size, and keyboard navigation to accommodate the users with different needs and preferences.
- Localization: The shipping application can improve the localization of the application, by using language, currency, and date format to adapt to the users from different regions and cultures.

19. Conclusion

In this paper, we presented a shipping application that uses dot and react as the main technologies. The shipping application allows users to create, track, and manage their shipments online. We described the design, implementation, and evaluation of the shipping application, as well as the challenges and benefits of using dot and react. We also discussed the future work and possible improvements for the shipping application. We concluded that dot and react are suitable technologies for a shipping application, as they offer performance, simplicity, and flexibility advantages.

References

[1]. A. Freeman, *Pro ASP.NET Core MVC 2*. Apress, 2017.

[2]. M. Henseleit, "Building Single Page Applications using React and Redux," *IEEE Software*, vol. 35, no. 3, pp. 28-35, May/June 2018.

[3]. P. Deitel and H. Deitel, *Internet & World Wide Web: How to Program*, 5th ed. Pearson, 2012.

[4]. M. Cibis, "Effective State Management in React Applications with Redux," in *2020 IEEE/ACM International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)*, 2020, pp. 55-64.

[5]. Microsoft, "ASP.NET Core Documentation," [Online]. Available: https://docs.microsoft.com/en-us/aspnet/core/. [Accessed: 01-June-2024].