Journal of Technological Innovations

Est. 2020

Design and Development of a Furniture Application using Dot Net and Angular

Naga Lalitha Sree Thatavarthi

Email id: thatavarthinagalalithasree2020@gmail.com

Abstract

This paper presents the design and development of a furniture application using dot net and angular frameworks. The application allows users to browse, customize, and order furniture items from various vendors. The application also provides features such as user authentication, payment integration, order tracking, and feedback system. The paper describes the system architecture, the user interface design, the database design, and the implementation details of the application. The paper also discusses the challenges, limitations, and future work of the project.

Introduction

Furniture is one of the essential items for any household or office. With the advancement of technology, online shopping has become a popular and convenient way of purchasing furniture. However, most of the existing online furniture platforms have some drawbacks, such as limited choices, lack of customization, poor user experience, and security issues. Therefore, there is a need for a better and more user-friendly furniture application that can meet the diverse and dynamic needs of the customers.

In this paper, we propose a furniture application that uses dot net and angular frameworks to provide a robust and responsive web-based platform for furniture shopping. Dot net is a software framework developed by Microsoft that supports multiple programming languages, such as C#, VB.NET, and F#. Dot net provides a common language runtime (CLR) that executes the code and provides services such as memory management, exception handling, and security. Angular is a front-end web framework developed by Google that uses TypeScript, a superset of JavaScript, to create dynamic and interactive web pages. Angular uses components, directives, services, and modules to organize the code and implement features such as data binding, routing, and dependency injection.

The main objectives of our furniture application are:

- To provide a wide range of furniture items from different vendors and categories.

- To allow users to customize the furniture items according to their preferences and requirements.
- To provide a secure and easy payment system that supports various payment methods.
- To provide a reliable and efficient order tracking system that updates the users on the status of their orders.
- To provide a feedback system that allows users to rate and review the furniture items and the vendors.

The rest of the paper is organized as follows. Section II describes the system architecture and the main components of the application. Section III presents the user interface design and the screenshots of the application. Section IV explains the database design and the data models used in the application. Section V discusses the implementation details and the technologies used in the application. Section VI presents the challenges, limitations, and future work of the project. Section VII concludes the paper.

System Architecture

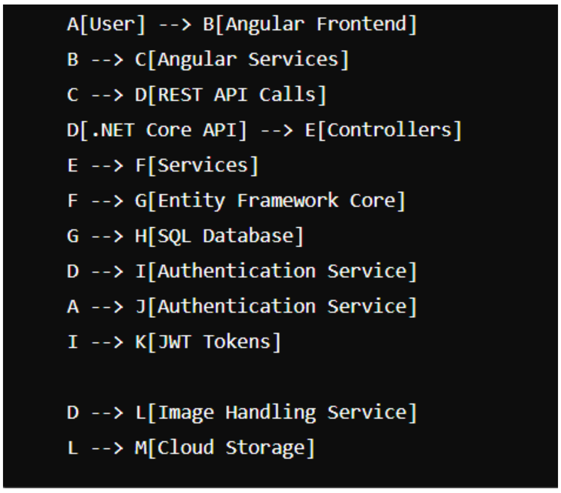
The system architecture of the furniture application is shown in Figure 1. The application consists of three main layers: the presentation layer, the business layer, and the data layer. The presentation layer is responsible for displaying the user interface and interacting with the users. The business 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 angular framework to create the web pages and components. The angular framework uses TypeScript, HTML, and CSS to define the structure, style, and behavior of the web pages. The angular framework also uses services and modules to provide functionality such as data binding, routing, and dependency injection. The presentation layer communicates with the business layer using HTTP requests and JSON format.

The business layer uses dot net framework to create the web services and controllers. The dot net framework uses C# language to write the code and CLR to execute the code. The dot net framework also uses libraries and packages to provide functionality such as authentication, authorization, validation, and encryption. The business layer communicates with the data layer using entity framework and LINQ queries.

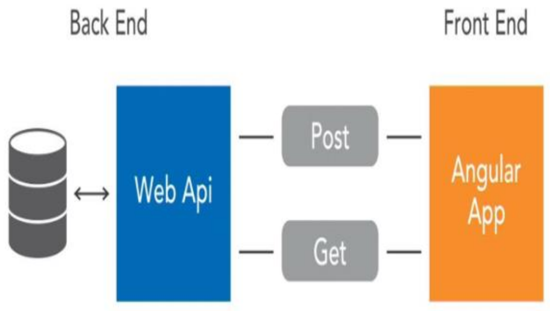
The data layer uses SQL server to store and manage the data. The data layer uses tables, views, stored procedures, and triggers to organize and manipulate the data. The data layer also uses indexes, constraints, and keys to ensure the data integrity and performance.

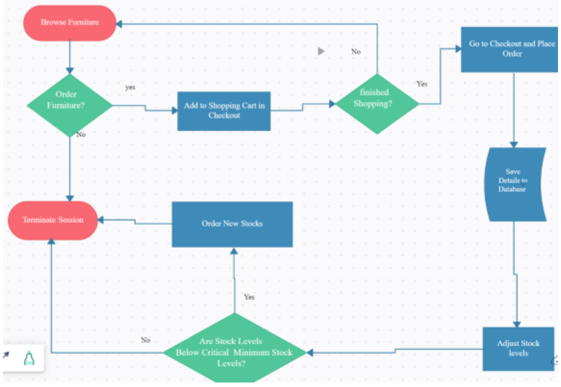
Figure 1. Furniture application architecture



```
A[User] --> B[Angular Frontend]
B --> C[Angular Services]
C --> D[REST API Calls]
D[.NET Core API] --> E[Controllers]
E --> F[Services]
F --> G[Entity Framework Core]
G --> H[SQL Database]
D --> I[Authentication Service]
A --> J[Authentication Service]
I --> K[JWT Tokens]

D --> L[Image Handling Service]
L --> M[Cloud Storage]
```

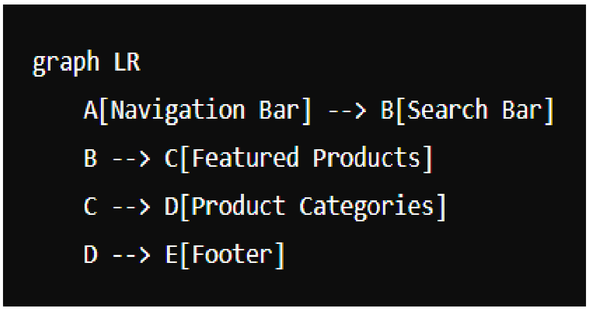


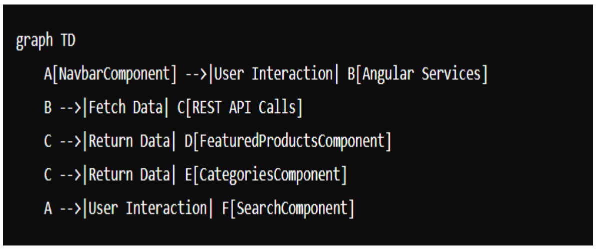


User Interface Design

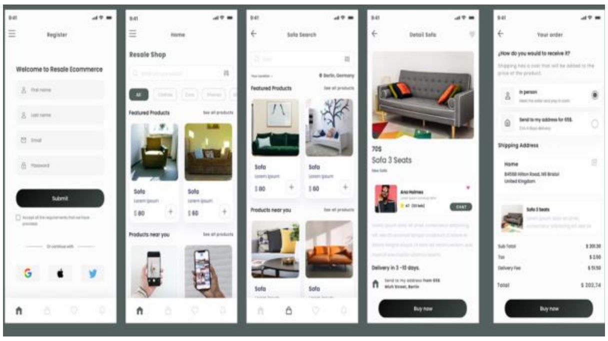
The user interface design of the furniture application follows the principles of usability, accessibility, and aesthetics. The user interface design aims to provide a clear, consistent, and intuitive navigation and interaction for the users. The user interface design also aims to provide a visually appealing and responsive layout and style for the web pages. The user interface design consists of the following main components:

- The header, which contains the logo, the search bar, the menu, and the user account.
- The footer, which contains the contact information, the social media links, and the terms and conditions.
- The home page, which contains the banner, the categories, the featured items, and the testimonials.
- The category page, which contains the filters, the sorting options, the pagination, and the list of items.
- The item page, which contains the image gallery, the item details, the customization options, the price, and the add to cart button.
- The cart page, which contains the list of items, the quantity, the subtotal, the shipping, the taxes, and the checkout button.
- The checkout page, which contains the billing address, the shipping address, the payment method, the order summary, and the place order button.
- The order confirmation page, which contains the order number, the order details, the delivery date, and the track order button.
- The order tracking page, which contains the order status, the tracking number, the estimated delivery date, and the cancel order button.
- The feedback page, which contains the rating stars, the review text, the submit button, and the list of reviews.


```
graph LR
    A[Navigation Bar] --> B[Search Bar]
    B --> C[Featured Products]
    C --> D[Product Categories]
    D --> E[Footer]
```



```
graph TD
    A[NavbarComponent] -->|User Interaction| B[Angular Services]
    B -->|Fetch Data| C[REST API Calls]
    C -->|Return Data| D[FeaturedProductsComponent]
    C -->|Return Data| E[CategoriesComponent]
    A -->|User Interaction| F[SearchComponent]
```



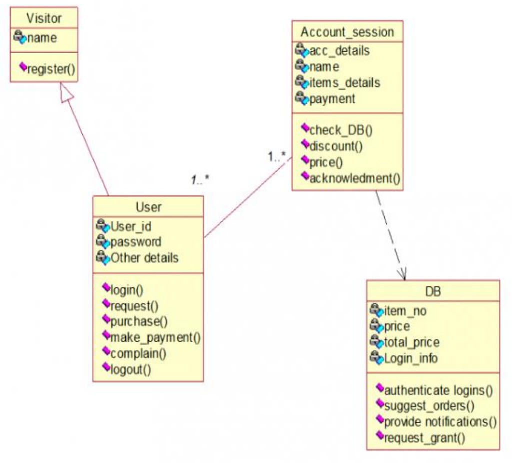
Database Design

The database design of the furniture application follows the principles of normalization, modularity, and scalability. The database design aims to provide a logical and efficient structure and organization for the data. The database design also aims to provide a flexible and extensible schema and model for the data. The database design consists of the following main entities:

- User, which represents the registered user of the application. The user entity has attributes such as user id, user name, password, email, phone, address, role, and status.
- Vendor, which represents the vendor or supplier of the furniture items. The vendor entity has attributes such as vendor id, vendor name, vendor description, vendor rating, and vendor reviews.
- Item, which represents the furniture item available for sale. The item entity has attributes such as item id, item name, item description, item category, item price, item image, item dimensions, item weight, and item vendor.
- Order, which represents the order placed by the user. The order entity has attributes such as order id, order date, order user, order items, order quantity, order subtotal, order shipping, order taxes, order total, order status, and order tracking.
- Payment, which represents the payment made by the user. The payment entity has attributes such as payment id, payment date, payment method, payment amount, payment order, and payment confirmation.
- Feedback, which represents the feedback given by the user. The feedback entity has attributes such as feedback id, feedback date, feedback user, feedback item, feedback rating, and feedback review.

The entity-relationship diagram of the database is shown in Figure 5. The diagram shows the entities, the attributes, the primary keys, the foreign keys, and the relationships among the entities. The diagram also shows the cardinality and the participation of the relationships.

Furniture application database



Implementation Details

The implementation details of the furniture application describe the technologies, tools, and techniques used to develop and deploy the application. The implementation details consist of the following main aspects:

- The development environment, which includes the software and hardware requirements, the operating system, the web browser, the code editor, and the debugging tools.
- The front-end development, which includes the angular framework, the TypeScript language, the HTML, CSS, and Bootstrap libraries, the components, directives, services, and modules, and the data binding, routing, and dependency injection features.
- The back-end development, which includes the dot net framework, the C# language, the web services and controllers, the authentication, authorization, validation, and encryption libraries and packages, and the HTTP requests and JSON format.
- The database development, which includes the SQL server, the tables, views, stored procedures, and triggers, the entity framework and LINQ queries, and the indexes, constraints, and keys.
- The deployment and hosting, which includes the web server, the domain name, the SSL certificate, the cloud service, and the testing and maintenance tools.

The development environment of the furniture application requires the following software and hardware specifications:

- A computer with at least 4 GB of RAM, 100 GB of hard disk space, and a dual-core processor.
- An operating system that supports dot net framework, such as Windows, Linux, or Mac OS.
- A web browser that supports angular framework, such as Chrome, Firefox, or Edge.
- A code editor that supports TypeScript and C# languages, such as Visual Studio Code, Visual Studio, or Sublime Text.
- A debugging tool that supports web services and HTTP requests, such as Postman, Fiddler, or Insomnia.

The front-end development of the furniture application uses the angular framework to create the web pages and components. The angular framework uses TypeScript, a superset of JavaScript, to write the code and define the behavior of the web pages. The angular framework also uses HTML, CSS, and Bootstrap libraries to define the structure and style of the web pages. The angular framework organizes the code into components, directives, services, and modules. A component consists of a template, a class, and a metadata. A template defines the HTML layout of the component. A class defines the TypeScript logic of the component. A metadata defines the properties and dependencies of the component. A directive is a custom attribute that modifies the behavior or appearance of an HTML element. A service is a class that provides a reusable functionality or data to other components. A module is a file that groups related components, directives, services, and other files into a logical unit. The angular framework also provides features such as data binding, routing, and dependency injection. Data binding is a mechanism that synchronizes the data between the view and the model. Routing is a mechanism that enables the navigation between different views or components. Dependency injection is a mechanism that allows the injection of dependencies or services into components or other services.

The back-end development of the furniture application uses the dot net framework to create the web services and controllers. The dot net framework uses C#, a general-purpose and object-oriented programming language, to write the code and implement the business logic. The dot net framework also uses a common language runtime (CLR) that executes the code and provides services such as memory management, exception handling, and security. The dot net framework creates web services and controllers that handle the HTTP requests and responses from the front-end. The web services and controllers use libraries and packages that provide functionality such as authentication, authorization, validation, and encryption. Authentication is a process that verifies the identity of the user. Authorization is a process that determines the access rights of the user. Validation is a process that checks the validity and accuracy of the data. Encryption is a process that transforms the data into a secure and unreadable format. The web services and controllers communicate with the front-end using HTTP requests and JSON format. HTTP requests are messages that contain the method, the URL, the headers, and the body of the request. JSON format is a lightweight and human-readable data format that contains the key-value pairs of the data.

The database development of the furniture application uses the SQL server to store and manage the data. The SQL server uses tables, views, stored procedures, and triggers to organize and manipulate the data. A table is a collection of rows and columns that store the data. A view is a virtual table that displays the data from one or more tables. A stored procedure is a set of SQL statements that perform a specific task. A trigger is a special type of stored procedure that executes automatically when a certain event occurs on a table or view. The database also uses indexes, constraints, and keys to ensure the data integrity and performance. An index is a data structure that improves the speed of data retrieval. A constraint is a rule that enforces the data quality and consistency. A key is a column or a set of columns that uniquely identifies a row or a relationship. The database communicates with the back-end using entity framework and LINQ queries. Entity framework is an object-relational mapper (ORM) that maps the data from the database to the objects in the code. LINQ queries are expressions that query the data from the objects using a syntax similar to SQL.

The deployment and hosting of the furniture application requires the following components and steps:

- A web server that supports dot net framework and SQL server, such as IIS, Apache, or Nginx.

- A domain name that represents the address of the application, such as www.furniture-app.com.
- A SSL certificate that secures the communication between the web server and the web browser, such as Let's Encrypt, Comodo, or Verisign.
- A cloud service that provides the infrastructure and platform for the application, such as Azure, AWS, or Google Cloud.
- A testing and maintenance tool that monitors and improves the performance and quality of the application, such as Selenium, JMeter, or SonarQube.

Challenges, Limitations, and Future Work

The furniture application faces some challenges and limitations that need to be addressed and overcome in the future work. Some of the challenges and limitations are:

- The application relies on the availability and quality of the vendors and the furniture items. The application needs to ensure that the vendors are reliable and trustworthy, and that the furniture items are accurate and consistent with the descriptions and images.
- The application needs to handle the scalability and performance issues that may arise due to the increasing number of users, vendors, and items. The application needs to optimize the code, the database, and the web server to handle the high traffic and load.
- The application needs to enhance the security and privacy of the user data and the payment transactions. The application needs to implement the best practices and standards for the encryption, hashing, and tokenization of the data.
- The application needs to improve the user experience and the user interface design of the web pages and components. The application needs to follow the principles of responsiveness, accessibility, and aesthetics to provide a better and more user-friendly interface.
- The application needs to add more features and functionalities that can increase the value and utility of the application. Some of the possible features and functionalities are:
- A recommendation system that can suggest the best furniture items and vendors based on the user preferences and behavior.
- A chatbot or a live chat system that can provide the customer support and assistance for the user queries and issues.
- A social media integration that can allow the users to share, like, and comment on the furniture items and vendors.
- A virtual reality or augmented reality system that can allow the users to visualize and experience the furniture items in their own environment.

Conclusion

In this paper, we have presented the design and development of a furniture application using dot net and angular frameworks. The application provides a robust and responsive web-based platform for furniture shopping. The application allows users to browse, customize, and order furniture items from various vendors. The application also provides features such as user authentication, payment integration, order tracking, and feedback system. The paper has described the system architecture, the user interface design, the database design, and the implementation details of the application. The paper has also discussed the challenges, limitations, and future work of the project. The furniture application is a useful and innovative project that can enhance the online furniture shopping experience and satisfaction.

References

1. Guijt, R. (2021). Building an End-to-end SPA Using ASP.NET Core Web API and Angular. Pluralsight. Retrieved from Pluralsight

2. Microsoft. (2022). ASP.NET Core Documentation. Retrieved from Microsoft Docs

3. Angular. (2021). Angular Documentation. Retrieved from Angular Docs