****

**CS ANTIQUE SHOP MARKETPLACE**

**Diploma in Software Engineering 22.1F**

**Final Project**

**CODSE22.1F-004 - Dileeka**

**CODSE22.1F-043 - Steve**

**CODSE22.1F-059 - Dilshan**

**CODSE22.1F-094 - Chehan**

**Contents**

Introduction …………………………………………………………………………………………………… pg 2

Use Case Diagram …………………………………………………………………………………………… pg 4

Class Diagram ………………………………………………………………………………………………… pg 10

Sequence Diagram… ……………………………………………………………………………………….. pg 20

ER Diagram……………..………………………………………………………………………………………. pg 28

File Diagram ………………………………………………………………………………………………….. pg 38

Interface Design …………………………………………………………………………………………...… pg 42

Report Layouts ……………………………………………………………………………………………..… pg 60

Conclusion ……………………………………………………………………………………………………… pg 67

References ……………………………………………………………………………………………………… pg 70

**Introduction**

purpose and scope of the project:

Our project's goal is to design CS Antiques Shop a polished and user-friendly website. The major objective is to build the shop's online presence so that clients can quickly browse and buy a variety of antiques and reproduction items from the comfort of their homes. By creating this website, we hope to give the store an effective and contemporary platform to present its distinctive inventory, draw in a bigger consumer base, and simplify the overall operation of the business.

The project's scope includes creating and implementing a user-friendly, aesthetically pleasing online interface that captures the spirit of CS Antiques Shop. Customers will be able to browse and choose products depending on their tastes thanks to the website's extensive product catalog, which will include in-depth product descriptions and high-quality photographs. The integration of safe online payment methods, a productive order management system, and user-friendly features like search filters, wish lists, and customer reviews are also included in the scope. The website will also feature an intuitive content management system that will make it simple to update and maintain product listings, price, and inventories.

an overview of the CS Antiques shop and its requirements:

A prominent store that specializes in the sale of antique and replica objects is called CS Antiques Shop. The business serves a varied clientele of antique aficionados, collectors, and interior designers and has a great collection of rare and priceless objects. Although clients from other countries visit the physical showroom, expanding into the online market is necessary to reach a larger audience and take advantage of the developing trend of online purchasing.

The criteria for the website project include developing an aesthetically pleasing and brand-consistent website interface that effectively displays the store's sizable inventory. Customers should be able to view individual product details, explore through various categories, and make safe online transactions. The website ought to include concise and accurate product descriptions that include background information and any noteworthy features. Customers will be able to find particular items by using filters like time period, style, or material thanks to a powerful search functionality. A user-friendly interface for managing client inquiries, processing orders, tracking shipments, and managing inventory should also be included on the website.

objectives and goals of the web page project:

The following are the goals of the CS Antiques Shop website project:

Create an online platform that reflects the shop's distinct personality and offerings in order to (a) establish an online presence, extend the shop's reach beyond its physical showroom, and draw in more customers.

b) Improve consumer experience: The website will offer easy-to-use navigation, thorough product descriptions, top-notch photos, and practical features to make browsing seamless and delightful for customers. Customers will be more satisfied as a result, and there will be more chances for them to make additional purchases.

c) Facilitate online sales: By enabling safe online transactions, the website will enable clients to conveniently buy desired items from the store's inventory. A quick and safe checkout process will be ensured through integration with reputable payment gateways.

d) Simplify business operations: The website will have a reliable inventory management system, making it possible to easily track available items, receive alerts on new arrivals, and manage stock in real-time. The shop's operations will be more organized and managed as a result of this.

e) Improve brand awareness and visibility: By creating a visually appealing website, the store hopes to improve its brand presence and position itself as a reliable supplier of high-quality antiques and replicas. As a result, there will be an increase in sales and growth as well as greater consumer trust and loyalty.

By attaining these targets, the CS Antiques Shop website project hopes to establish the store as a premier online stop for antique lovers, providing a pleasant and enjoyable buying experience.

**Use Case Diagram**

A use case diagram is a graphic representation of how users or external systems interact with a system or piece of software. It illustrates the numerous use cases or functionality that the system offers and demonstrates how various actors engage with those functionalities. Use case diagrams are a crucial tool in software development because they make it easier to comprehend, evaluate, and convey the needs and behavior of the system.

The following points help to clarify the significance of use case diagrams in software development:

Use case diagrams, which represent the functional needs of the system from the viewpoint of the user, are essential throughout the requirement analysis process. They give a clear knowledge of the system's objectives, the players' activities, and the anticipated results. Developers may make sure that the system satisfies users' needs and expectations by identifying and documenting the use cases.

Use case diagrams serve as a communication and collaboration tool for all parties involved, including developers, designers, project managers, and clients. They offer a common vocabulary to talk about the interconnections and functionalities of systems. By enabling stakeholders to visualize and align their understanding of the system, use case diagrams promote collaboration and promote more efficient communication and decision-making.

Use case diagrams are helpful in the design of the user interfaces and system architecture. They assist in determining the key elements or modules of the system and their connections. The necessary interfaces, inputs, and outputs for each use case are also determined with the use of use case diagrams. The structure and behavior of the system are designed using this information as a foundation.

Use case diagrams are essential for test planning and the creation of test cases. They give a thorough overview of the system's features, which aids in determining which scenarios need to be tested. By explicitly deriving test cases from use cases, it is possible to guarantee that all crucial system interactions are covered during testing. Better test coverage is achieved as a result, and the system's behavior is checked against the given criteria.

Use case diagrams help with system upkeep and potential future improvements. They act as a point of reference for comprehending the operation of the current system and can direct developers when adding new features or changing existing ones. Use case diagrams make it simpler to determine how changes will affect other system components and make sure that the changes are consistent with the system's general behavior.

Use case diagrams are crucial for software development because they show how a system functions and interacts visually. They support requirement analysis, dialogue, system design, test preparation, and system upkeep. Software development teams can make sure that the final system satisfies user expectations and accomplishes the necessary goals by efficiently utilizing use case diagrams.

**Class Diagram**

The structure and interrelationship of the classes, interfaces, and their interactions inside a software system are shown visually in class diagrams. It is a key tool in object-oriented modeling and offers a high-level view of the architecture of the system, focusing more on the static structure than the dynamic behavior.

Our project's class diagram has multiple functions.

Modeling Classes and Objects: The system's primary classes and objects can be found using the class diagram. It represents each class's blueprint, including its methods (operations or behaviors) and attributes (data fields). We are able to comprehend how the system is organized and how various entities interact with one another by expressing classes and objects in the diagram.

Defining Relationships: The class diagram highlights the affiliations, dependencies, and inheritances that exist between classes to show how they are related to one another. While dependencies show that one class depends on another, associations depict the connections or links between classes. The hierarchical links between subclasses and superclasses, represented by inheritances, are illustrated. We can clearly grasp the structure and interdependence of the system by specifying these interactions.

Data and behavior encapsulation: Using the class diagram, we may define classes that include data and behavior. The state or information related to objects of a class is stored in data fields, which represent the characteristics of a class. The actions or behaviors that can be taken on objects are specified by methods. We achieve the principles of abstraction, modularity, and information hiding by encapsulating data and behavior within classes, resulting in a system that is more reliable, maintainable, and reusable.

System Architecture Illustration: By showing how classes are organized and interact, the class diagram illustrates the system's architecture. Understanding the various system layers, parts, and modules is aided by this. By making the links and duties of different classes more clear, this representation helps with system design and implementation.

Implementation Guidance: The class diagram acts as a reference during the software development process' implementation phase. It offers a guide for programmers to build classes, specify their properties and methods, and construct connections between them. By using a class diagram, it is possible to confirm that the implementation follows the planned system behavior and design.

Supporting System Documentation: The class diagram is an important piece of documentation that aids in explaining the structure of the system to all relevant parties, such as developers, designers, and project managers. It offers a clear and visual picture of the system, making it simpler to comprehend and go through its constituent parts and relationships.

A strong tool for simulating the architecture of a software system is the class diagram. It records the classes, objects, characteristics, methods, and their connections, making it easier to comprehend, develop, put into practice, and document the system. We may guarantee a well-structured and maintainable software system in our project by successfully applying the class diagram.

**Sequence Diagram**

In UML (Unified Modeling Language), sequence diagrams are a sort of behavioral diagram that show the communications and interactions between objects or components in a software system. By displaying the sequence of actions and the movement of control during a certain use case or scenario, they demonstrate the dynamic behavior of the system.

In our project, sequence diagrams are used to record and depict the interactions between the items or components used to carry out particular use cases. They are essential for comprehending the progression of events, the order in which objects work together, and the signals that are passed back and forth. The function of sequence diagrams in our project is as follows:

Sequence diagrams give a visual representation of how items interact with one another through time, making it easier to visualize object interactions. They show the flow of control and the hierarchy of processes by illuminating the order of messages that are passed between objects. Sequence diagrams aid in understanding the dynamic behavior of the system and how items work together to complete a certain goal by showing these interactions.

Sequence diagrams are useful for examining and comprehending the system's behavior when a use case is being executed. They document the various processes, circumstances, and interactions that take place within the system. Developers and stakeholders can acquire insights into the logic and behavior of the system, detecting potential problems or changes, by looking at the sequence of messages and the participation of objects.

Sequence diagrams are used to identify partnerships and dependencies between objects or components. They draw attention to which items are in charge of carrying out particular procedures or tasks and which things rely on other objects to function. This knowledge makes it easier to comprehend how the system's relationships and dependencies work, resulting in more effective design, implementation, and maintenance.

Verifying Use Case Flows: Sequence diagrams offer a way to check and confirm that a use case will behave as expected. The required interactions and processes can be appropriately portrayed by developers by comparing the sequence diagram with the use case description. By doing so, it is possible to spot any incomplete or inaccurate phases and improve the use case flow so that it more closely matches the expected system behavior.

Designing and Developing the System Architecture: Sequence diagrams help to design and develop the system architecture. They help in defining the items or elements involved in a particular use case, their functions, and their relationships. This information can help architects make decisions about how to define interfaces, specify the roles of various components, and maximize the effectiveness and performance of the system as a whole.

Supporting Communication and cooperation: Sequence diagrams serve as a tool for cooperation and communication among project stakeholders. They offer a visual representation that programmers, designers, and other stakeholders can easily comprehend. Sequence diagrams make it easier for people to talk about, clarify, and agree upon how the system behaves. As a result, everyone involved can see how the system works together.

In order to accurately capture and depict the interactions between items or components in a software system, sequence diagrams are crucial. They facilitate efficient communication and cooperation among project stakeholders by aiding in the visualization of object collaborations, analysis of system behavior, identification of dependencies, confirmation of use case flows, and creation of the system architecture. In our project, sequence diagrams will help us better understand the dynamic behavior of the system and ensure that use cases are carried out successfully.