A critical assessment of the design issues in e-commerce systems development

Sylvanus A. Ehikioya1 | Eric Guillemot2

1Department of Computer Science, Baze University, Abuja, Nigeria
2Department of Computer Science, University of Manitoba, Winnipeg, Manitoba, Canada

Abstract
Designers of electronic commerce (e-commerce) transaction system applications face many challenges; some are old while others are emerging because of the emerging nature of the e-commerce domain itself. This article examines the critical design challenges at different stages of the design process of e-commerce applications. The high-level designs have a large number of new issues specific to Web applications. Among them, the security and networking problems are the most determinant for the success of the applications. At the detailed design level, the languages and the user interface are major concerns. The user interface, specific to the Web, forces designers to redefine the traditional concept of the end user for the more appropriate customer-centric concept. This is necessary to address the needs of consumers and for the survival of e-commerce businesses. In addition, we also examine issues bothering the development team and the e-commerce environment. Designers experienced in non-Web applications can be more successful at developing e-commerce applications if they address the issues we identified in this article in a creative way.

KEYWORDS
design issues, e-commerce, e-commerce systems/application

1 INTRODUCTION

The concept of electronic commerce (e-commerce) has crystallized into a fairly well understood concept1-7; its popularity and fast paced acceptance as an alternative to the brick and mortar market economic model3,8 tend to undermine a clear understanding, and clear exposition of the characteristics, of e-commerce. Most adherents of e-commerce often focus only on the exchange of value end product6,7 without paying close attention to, or knowing the intricacies of, the underlying transaction itself.1,4,9-13 Therefore, a proper characterization of e-commerce applications is fundamental. We focus on the design challenges in e-commerce transactions in a typical business-to-consumer (B2C) e-commerce domain, which is more popular, more widely patronized by users, and attracts a higher volume of e-commerce transactions. The B2C e-commerce includes retail transactions of products or services from businesses to individual shoppers (ie, businesses sell products or services to individual consumers). Thus, the seller is a business organization whereas the buyer is a consumer.
The platforms of Amazon.com (www.amazon.com), one of the most well-known e-commerce web sites, and Walmart.com (which sells merchandise to consumers through its web site) are examples of this type of e-commerce. Since the sellers are usually retailers, B2C can sometimes be referred to as e-tailing because this model emulates the situation of physical retailing.

A critical element in the development of e-commerce applications is the design phase, but it is often characterized by a lot of challenges, that it deserves special attention. Design, the first building and most creative step of the software development life cycle (SDLC), is the phase where software engineers decide how the system will be constructed. Davis summarizes the design phase as follows:

*The design should minimize the intellectual distance between the software and the problem as it exists in the real world.*

The design phase is usually divided into two parts: high-level design and the detailed design. The high-level design, as its name implies, is concerned with high-level decisions such as the product architecture (ie, the breakdown of the application into subsystems), the type and structure of the database, the languages, and the tools to use. The output of this phase is a set of documents and diagrams, such as the architectural diagrams, that are used to report high-level decisions made by software engineers. The decisions taken at this level are critical as any change later in the SDLC is either impossible or difficult to effect without significantly impacting on a large part of the already created product. In the detailed design phase, design engineers focus on the user interface and the algorithms. This phase is a preparation for the coding activity and, therefore, engineers must be competent in all the languages and tools chosen to build the application.

A characterization of e-commerce application that provides key contextual environment for the understanding of e-commerce transactions is available in the literature but will not be discussed in this article, so interested readers should see the referenced resources. However, it suffices to mention these characteristics here: concurrent, distributed, dynamic, complex data access patterns, and real-time behavior of e-commerce transactions. Other properties are fail-safe, dependable, guarantee correctness of operations, secure environment, and trustworthy and minimize risks. The characterization enables us to properly dimension the various aspects of e-commerce applications and hence their design requirements. The design of e-commerce applications incorporates a wide range of technologies (eg, networking, multimedia, publishing, etc.) and is, therefore, one of the most challenging. Designers must not only be competent in all these areas but must also be able to anticipate new technologies and design accordingly.

e-Commerce is a relatively new domain. It is tempting to pretend that conventional software engineering methods cannot be used and, therefore, tend to begin in level one of the five levels of the Capability Maturity Model defined by the Software Engineering Institute (SEI) as shown in Figure 1.

![FIGURE 1 Capability maturity model](image-url)
for this are many but we highlight only a few below. Quality management in agile development is informal rather than document-based and the review process is usually informal. Also, the agile community sees as bureaucratic overheads the use of standards-based approaches and quality processes as embodied in ISO 9001. Developing dependable systems (including e-commerce applications) requires the adoption of, and compliance to, both process and product standards. Product standards apply to the software product being developed. They include document standards, such as the structure of requirements documents, documentation standards, such as a standard comment header for an object class definition, and coding standards, which define how a programming language should be used. Process standards define the processes that should be followed during software development, which may include definitions of specification, design and validation processes, process support tools, and a description of the documents that should be written during these processes. Reviews and inspections of the system being developed are critical; they examine part or all of a process or system and its documentation to find potential problems. Unfortunately, agile methods do not meet this standard, and often can suffer from incomplete software testing that does not include tests with different (method) parameters. While some merchants outsource the development of their e-commerce platforms to professional developers, most e-commerce applications developments are done in-house, using automated tools/assistants such as Adobe Dreamweaver CC and Microsoft Visual Studio or using packaged web site building tools such as IBM WebSphere and Sitecore Commerce Server 11, by designers who may not have been trained or retrained specifically for Internet applications. This leads to a “culture” where poorly designed applications are the norm, and standardization is not encouraged or emphasized.

As Web applications moved from purely static applications to fully dynamic applications, a new or modified reference model became necessary for designing applications such as e-commerce applications. An example of such a model is the reference model proposed in Reference 19. Others propose to use the Unified Modeling Language to model e-commerce applications. A good example of the efforts to develop a formal specifications strategy specifically for e-commerce is documented in References 1 and 9. These resources and others examine, in particular, the correctness of e-commerce transactions.

This article examines the design challenges in the development of e-commerce applications, the tools, and the techniques that are used to resolve the design challenges, and provides a basis for application design and implementation that is robust, secure, and fail-safe, and offer better quality of user experience to e-commerce applications users. We categorize the design challenges and then examine each category in detail by isolating the design elements in each group. Our approach offers the following key benefits:

• Our analysis is simple and easy to understand by potential users and domain experts, thereby providing opportunities for detailed discussions, which can identify ambiguous, incomplete, or inconsistent designs and provides the basis for a detailed system description that can be used to generate a system implementation; thus, models have to be both correct and complete.

• The incremental and categorical analysis of the key design attributes’ challenges of e-commerce applications, which ensure correctness and completeness, in turn supports collaborative development among the development team members.

The contributions of this article are:

• We provide an extensible infrastructure input to the e-commerce model that offers opportunities to reason about designing e-commerce applications systematically, thereby intuitively and inherently providing and enforcing a correct, fail-safe, unambiguous, and robust transaction processing environment for e-commerce.

• Identifies the key challenges in the design of e-commerce applications and juxtaposes these with e-commerce applications functionality to ensure good customer quality of user experience.

• Provides a roadmap for challenges in the design of e-commerce applications/systems to accommodate the increasing sophistication of e-commerce applications and the increasing demands of users for higher quality of user experience.

• It provides a practical case study for gaining knowledge and skills acquisition on designing an e-commerce system. The issues/challenges identified in designing e-commerce applications provide areas of realistic concern that demand appropriate attention and requiring use of standardized design methods and tools. The basic principles and thrust on which we anchored our examination of the issues is applicable to solving real-world problems and provides a stimulating learning experience to other system developers.
• The implementation of any design that addresses the design challenges we examined in this article provides a test-bed platform/environment for learning and gaining practical experience in Web applications architecture (software) design and implementation.

This article is significant for the following reasons:

• First, by identifying the design issues at different levels of abstractions makes the understanding of the relevant elements in e-commerce application development clearer.
• It helps e-commerce applications developers to focus on the identified design challenges in implementing e-commerce solutions that are robust, secure, and fail-safe.
• It provides the basis for reengineering Web application functionalities that support e-commerce applications transactions to offer better quality of user experience to e-commerce applications users.
• While not reinventing the wheel, our study helps e-commerce application developers to identify and use appropriate system design methods and tools to provide support for e-commerce functionalities. This approach helps speed up e-commerce application development by focusing on the proven design issues during implementation.

This article is helpful for designers and developers of e-commerce applications because it highlights some key fundamental or pragmatic challenges in the development of e-commerce applications. Therefore, examining the design process, operational efficiency, and effectiveness is clearly important because we are able to understand, apply, and promote notable design principles, processes, and practices to overcome the identified challenges.

The remainder of this article is organized as follows: Section 2 provides a general review of some of the research in the acceptance of e-commerce and some of the problems in software design in general, while in Section 3 we briefly introduced the design goals of software development to provide appropriate context. Section 4 describes in detail the critical issues in the design of e-commerce applications/systems. The issues/challenges are classified into five categories: high-level design issues, team issues, user interface issues, detailed design, and environment issues. In Section 5, we show in a tabular format the essential elements of, and their relationships with, design goals, e-commerce functionality, and design features. We provide a discussion of our results/experience in Section 6. Finally, we conclude the article and outline some future directions for this research in Section 7.

2 | BACKGROUND AND RELATED WORK

Software design is a fundamental concern during the software development process. Design issues are frequent concerns among experiences shared by developers. This is fundamental because early design decisions by developers may influence some key attributes of a system, such as maintainability, robustness, comprehensibility, and performance.

Bourque and Fairley discussed some key issues that must be dealt with in software design, namely: concurrency, control and handling of events, data persistence, distribution of components, errors and exception handling and fault tolerance, interaction and presentation, and security. While some of these key issues are quality concerns (e.g., performance, security, reliability, usability, etc.) which all software must address, other issues deal with crosscutting concerns such as how to decompose, organize, and package software components. Also, Hummel et al. provided an overview of some key software engineering challenges and allocated them to the common software development life-cycle activities while focusing on crosscutting concerns.

According to Sousa et al., a software design problem occurs when a software design decision has a negative impact on one or more quality attributes of the software. A design problem impacts on some of the key properties (such as extensibility and maintainability, robustness, comprehensibility, and performance) of a software system. They opined that it is difficult to identify a design problem in a system, and in particular when the source code is the only available artifact, it is more challenging to identify the design problems. They concluded that developers/software designers often use multiple indicators during the identification of each design problem.

The harmful effects of design problems have led to the discontinuance or reengineering of some software systems. Silva et al., Xiao et al., and Sousa et al. show that design issues are one of the most common categories of technical issues that lead to the failure of software systems. Therefore, given the negative effects of design problems, it is
recommended that developers should remove them from software systems as early as possible\textsuperscript{32,33} in the software development process. However, identifying the design problems is nontrivial.\textsuperscript{34,35}

Tago et al\textsuperscript{36} identified interface design and experience, among other factors, as crucial and critical factors that help to enlarge our comprehension of the application and use of [mobile] electronic commerce applications. This also validates the original technology acceptance model (TAM) constructs.\textsuperscript{37-39} Aniche et al\textsuperscript{40} examined the practical challenges of e-commerce applications design in modern software development. According to Aniche et al,\textsuperscript{40}

“good software designs are, among other characteristics, lowly coupled, highly cohesive, extensible, comprehensible, and not fragile.”\textsuperscript{40}

Besides the above, additional requirements are, however, necessary due to the increased complexity and heterogeneity of contemporary software.

Koussouris and Di Nitto\textsuperscript{41} state that

“high quality software requires specific skills and the adoption of good and controlled development and operation practices.”\textsuperscript{41}

They assert that the challenges in software engineering, which encompass both the process and technological aspects, should be resolved in order to be able to develop high quality software. Koussouris and Di Nitto\textsuperscript{41} and Casale et al\textsuperscript{42} identified the challenges as: (a) process, methodologies, and productivity issues that seek to redefine software measurement in terms of its other qualities such as usability, reliability, scalability, etc. instead of the obsolete and deprecated “lines of code” as the right measure of productivity; (b) application contexts; (c) design patterns development at the architectural level that describes the obligations/constraints of the software system environment, while elevating the need for their validation and standardization, and development of methods for their application in a dynamic environment; (d) quality guarantees to cope with the emerging trends in software development; (e) requirements engineering issues resulting from the increased complexity of knowledge capturing and representation because of the peculiarities of connected-world concept; (f) privacy and security by design; (g) new system architectures and software infrastructures able to cope with big data; and (h) the continuous need toward accelerating open source software innovation.

The position held by Koussouris and Di Nitto\textsuperscript{41} supports the various views expressed in the literature\textsuperscript{1,2,9,21,43} canvassing for the use of formal methods in the modeling of e-commerce applications. The main challenges in the new web environment include the development of models, methods and design tools for web-enabled applications, going beyond mere formal methods research, to creating abstractions and formalisms to construct and reason about systems with diverse and more difficult-to-characterize components as enunciated in Reference 17. Web applications also need novel methods for software adaptability, scalability, and maintainability because they need to be continuously modified and maintained to meet changing requirements during runtime, thereby requiring run-time adaptation of software quality assurance and capability to deal with uncertainty and variability at the same time, software-awareness of hardware to ensure Web-scale performance, flexibility, and agility.\textsuperscript{41}

Basha et al\textsuperscript{44} examined the domain specificity, meta-modeling, and iterative transformation challenges in model-based software development (MBSD). Although the idea of using models to alleviate software complexity is by no means new, however, their applicability have been restricted largely to selected aspects of the development process, particularly structural and compositional aspects in the design phase, and model checking and verification in the testing phase. These two phases are insufficiently interconnected\textsuperscript{44} due to the lack of a unified theory to express relevant concepts at an appropriate abstraction level.\textsuperscript{9} However, the development of complex software, such as e-commerce applications, requires that the developers provide models of the system from multiple perspectives, such as structure, behavior, interaction, and architecture and nonfunctional properties.

MBSD demands an enhanced requirement on software models for completeness and precision, and an efficient mechanism for mapping between design model and implementation. These requirements (completeness and precision, and efficient model—implementation mapping mechanism) extend beyond generating model-prescribed code to also include management of the consistency between model and code as the software evolves. The cost of mapping between design model and implementation often significantly exceeds that of working directly on code; therefore, MBSD is often not pervasive in practice. However, “software models in MBSD not only have to contain enough details to generate relatively complete code, but also need to be, and stay, simpler than the software programs created during this process.”\textsuperscript{44}
van der Straeten et al\textsuperscript{45} identified scalability, consistency and coevolution, formal foundations, and industrial adoption as key challenges in model driven software engineering. Similarly, the participants at the International Workshop on Challenges in Model-Driven Software Engineering\textsuperscript{46} identified the following major categories of challenges:

- **Model quality**: The need to deal with quality aspects in modeling and model-driven engineering.
- **Run-time models**: The need for shift of focus from primarily on using models at analysis, design, implementation, and deployment stages of development to the use of models during run-time which extends the use of modeling techniques beyond the design and implementation phases of development. This paradigm shift introduces a number of challenges, such as dynamic behavior representation.
- **Requirements modeling**: The key challenge is the need for appropriate modeling framework and how to achieve traceability between requirement specifications and design models and how to bridge the gap between the often predominantly informal (textual) requirement specifications and formal requirement models.
- **Standards and benchmarks**: Benchmarks and standards provide an excellent resource to compare different tools and approaches. However, there is the absence of widely accepted benchmarks. In this respect, several key challenges exist; such as how to design and develop benchmarks that facilitate comparison between tools and approaches, the determination of the standards that are needed to facilitate interoperability between tools, and determining how to obtain and share common data models and model transformations.
- **Modeling languages**: Appropriate models cannot be developed without precise modeling languages. The key challenge is determining which languages, methods, principles, and tools are necessary to design precise meta-models.
- **Domain-specific modeling**: Models for complex systems encompass several domains. Capturing all important aspects of such a complex system requires developing multiple models using different domain-specific modeling languages (DSMLs) and introduces many challenges such as how to develop and integrate models using different DSMLs.
- **Model verification and validation**: Verification and validation is an important aspect of model driven design. A critical challenge is determining how to verify, validate, debug, and test the models and the code generated from those models.
- **Formal foundations**: A formal foundation is a necessity for the verification and validation of models and their transformations. The key challenge is how to integrate formal verification tools into modeling environments. We need intuitive, simple and yet expressive tools that do not require the user of the modeling environment to have any expertise in the different formalisms and techniques used for verification.
- **Model consistency and coevolution**: Models are built using a variety of DSMLs. The question arises how to develop DSMLs that can be efficiently extended, adapted, or customized.
- **Other notable challenges include empirical analysis, process support, industrial adoption, and scalability issues.**

The specification and testing of model transformations are not easy tasks and they present numerous challenges. Vallecillo et al\textsuperscript{47} present some of the key issues in model transformation specification and testing. They introduced the concept of tract, which is a collection of source models satisfying the source constraints, that generalizes model transformation contracts and show how they can be used for model transformation specification and black-box testing, and the kinds of acceptable analyses.

Otero\textsuperscript{48} identified requirements volatility, the design process, technology, ethical and professional practices, and managing design influences as key design challenges. In addition, other major challenges include quality issues (eg, performance, usability, security, etc.), distributed software development, efficient allocation of human resources to development tasks, limited budgets, unreasonable expectations and schedules, fast-changing technology, and accurate transformation from software requirement to a software product.

A key characteristic of requirements volatility is its potential to impact future or ongoing development efforts. This feature, according to Otero,\textsuperscript{49} “forces designers to create designs that provide solutions to problems at a given state while also anticipating changes and accommodating them with minimal effort. This requires designers to have a strong understanding of the principles of software design and develop skills to manage complexity and change in software development.”\textsuperscript{49}
Privacy and security at both design time and runtime of e-commerce application software is crucial to comply with software development methodologies evolution as the landscape of threats and vulnerabilities is continuously expanding. In particular, complex distributed systems, which in many cases have to handle large data volumes in a distributed way, need special attention in handling privacy and security.

Information systems and technology have become essential tools for organizations to achieve their goals. Al-Mamary et al.\(^6\) examined models and theories on individual acceptance of technology and x-rayed the factors which make information systems successful based on the theoretical models rooted in the literature.\(^{37,38,51-55}\) Munir et al.\(^5\) examined customer acceptance of, and the factors that influence the use of, mobile banking services in Makassar using the TAM framework. They collected data through structured interviews with a questionnaire. The results showed that (a) perceived ease of use significantly influence perceived usefulness and intention to use, and (b) perceived usefulness significantly influence intention to use. They concluded that to increase customer acceptance or adoption of mobile banking services, the banks should pay adequate attention to, and highlight factors such as, perceived usefulness and perceived ease of use.

Islam et al.\(^5\) also assessed the adoption of mobile banking in Bangladesh using the unified theory of acceptance and use of technology (UTAUT) model.\(^{52,57}\) Tan\(^5\) similarly used the UTAUT model to understand the factors affecting the Use of English e-Learning websites in Taiwan. Similarly, Al Mursalin\(^5\) explored 255 Bangladeshi SMEs to conceptualize current trend of information systems usage and validates the UTAUT model to explain information systems adoption behavior. The study shows that although a number of Bangladeshi SME’s are exposed to computer-based information systems, their usage, in most cases, are limited to operational level of activities. Tago et al.\(^5\) provided a conceptual framework for the understanding of mobile commerce applications based on the well-reasoned and revised TAM\(^{35,42,50}\) that comprises of the distinct features from mobile commerce, electronic business, and information systems domain. Abubakar and Ahmad\(^6\) similarly used a conceptual extension of UTAUT\(^6\) with concern for customers in a study of the determinants of behavioral intention to use e-payment system in the Nigerian retail industry. Also, Raeisi and Lingjie\(^6\) similarly used an integrated model of theories (TAM, TAM3, UTAUT, IDR, etc.) to study which factors influence consumers’ intention to use mobile commerce and influence mobile commerce adoption by Chinese people.

Min et al.\(^6\) described a theoretical framework that incorporates the unique characteristics of mobile commerce to enhance our understanding of mobile commerce acceptance and usage in China. The study, based on the revised UTAUT model, includes the characteristics of mobile commerce, inclusion and consideration of information system user satisfaction theory, and Chinese culture and context. Venkatesh et al.\(^5\) Attuquayefio and Addo,\(^6\) and Sharma and Mishr\(^6\) provide a comprehensive examination of the UTAUT model. Similarly, a comprehensive examination of technology adoption models and theories for the novelty of technology are available in References 50, 60, and 66-68.

Many studies\(^5,69-79\) have examined the factors that significantly influence customer satisfaction and customer loyalty in e-commerce business. Several factors such as perceived usefulness, perceived ease of use, customer risk perception, customer support, customer trust, promotion, information quality, transaction security, website design, and brand image of e-commerce influence satisfaction when shopping on an e-commerce website and has significant impact on customer loyalty. Khalid et al.\(^80\) concluded that e-commerce service quality (particularly related to security and payment method) is the principal element affecting customer satisfaction with e-commerce system. Also, information quality, system quality, service quality, product quality, delivery quality and perceived price have been identified as factors that affect the overall user quality of experience in using e-commerce applications.\(^81\) Tabaei et al.\(^82\) found product features and website quality as key factors for electronic customer satisfaction. They conclude that service quality dimensions in terms of ease of use, website design, responsiveness, customization and assurance, amongst other factors, have positive relationships with customer satisfaction and trust.

Cao et al.\(^83\) identified, examined, and integrated four sets of factors (system quality, information quality, service quality, and attractiveness) that capture e-commerce web site quality using the TAM model. They provide practical guidelines for e-commerce managers and web designers. Their result supports the position of Chakraborty et al.\(^84\) in the study of antecedents of website effectiveness. Also, Dwairi and Kamala\(^45\) stated that ease of navigation, navigation time, page download, and use of multimedia to improve Web site’s visual appeal are the attributes that influence the design factor in evaluating the quality of B2C e-commerce Web sites.

Gelard and Negahdari\(^74\) posit that a number of web specific factors including navigation, interface and customized content and update are critical elements in e-commerce systems development. Marszalekowsk\(i\) et al.\(^86\) analyzed load time factor in search engine ranking which is essential for marketing plans of many e-businesses and they show that the load time factor plays a significant role in the ranking of search results. Vollset\(^87\) identified page loading time as one of the major complaints by e-commerce website users. The speed of a site negatively impacts a user’s session depth, thus a website...
page speed affects a merchant’s revenue because loading speed is a major contributor to page abandonment. Slower page response times result in an increase in page abandonment.

We rely on the innovation diffusion theory (IDT) and refined IDT frameworks to model the design of electronic commerce applications interfaces. We also draw on the conclusions on the use of TAM and its extensions and UTAUT models as the theoretical basis for the design of electronic commerce applications interfaces. Using the revised TAM, which includes distinct features for electronic commerce and information systems domains, helps to increase our understanding of electronic commerce applications and acceptance. We focus on the theoretical contributions of these models rather than issues that bother on method because this approach would help practitioners to comprehend insights into how to promote electronic commerce to customers. We concur, however, with the conclusion by Tago et al. that method issues can influence findings and conclusions.

3 SOFTWARE DESIGN GOALS

Design, the creative process of defining how to implement a system based on the customer’s requirements, is a problem solving activity. Design problems have unlimited number of alternative solutions. Therefore, a thorough analysis becomes a key requirement in advance of any implementation. It is essential that a precise linkage is established between the use cases and requirements and the designs. Therefore, our design abstractions must be grounded in reality by properly evaluating entities that are both internal and external to the desired software application.

The goal of the design often dictates the design approach; whether to use the object-oriented design or use traditional function-oriented design methods, whether to design for reuse in mind or not, adopt the principle of designing for testability, etc. The primary software design goals include:

- Functionality
- Performance
- Security and safety
- Availability
- Maintainability
- Usability

The classical software quality attributes indentified in McCall et al. and Boehm et al. and consolidated in ISO/IEC 9126-1:2001 recommendations are applicable here and these resources provide a relationship between the quality factors and quality criteria in a tabular format. These will not be repeated in this article and readers should see References 90-94 for the detailed discussion of software quality characteristics and attributes. It suffices, however, to state that achieving and satisfying the quality factors (usability, integrity, efficiency, correctness, reliability, maintainability, testability, portability, reusability, interoperability, flexibility, availability, etc.) of software is the primary goal of a good software design.

The testing process is not the exclusive confine of the coding activity. Testing of the system design with walkthroughs and other design review techniques is very useful. This is challenging for e-commerce applications developers if not properly handled. Thus, a thorough analysis of the design in advance of implementation is crucial and this can be made easy by adopting consciously the principle of designing for testability.

The design of any software system must, therefore, meet the performance, security, safety, availability, maintainability, usability requirements of the system in order to meet the needs of the user/client, on the one hand, and the needs of the developer on the other.

4 DESIGN ISSUES IN E-COMMERCE APPLICATIONS

In this section, we classify the design issues/challenges into five categories: high-level design, development team, user interface, detailed design, and environment. Some of these issues are specific to e-commerce, while others are more general concerns found in Web applications. The more general issues can be applied to any software system. Pressman characterizes Web applications as:

… network intensive, content driven and continuously evolving.
This description summarizes the features of Web applications; and in a way points to the numerous inherent challenges designers face with e-commerce applications. Network issues are a consequence of the client/server architecture and distributed model of e-commerce applications and the relative inadequacy of the infrastructure of the Internet. The Web, initially designed to display only text, today is used for pictures, sounds and movies and almost any sort of application that one can imagine. This is a challenge for designers who have to deal with the growing demand for the integration of modern and emerging technologies such as video, text, and voice data communications. Although the Web naturally supports these media elements, their integration to deliver better quality of user experience is a nontrivial matter.

In the following subsections, we examine in details the design challenges enumerated above.

## 4.1 High-level design Issues

Recall, high-level design is concerned with high-level decisions and every one of such decision is critical to the future of the development process of the e-commerce application. This subsection explains in details the critical issues at this phase of the development process. In each area, we focus on only the problems encountered in the high-level design. Other issues related to an area, but not related to high-level design, are described in the corresponding subsections. This is necessary to put the subsequent discussion that follows in proper context.

### 4.1.1 Security

The security of e-commerce applications is critical to the success of the e-commerce business model. According to Qualys,

> **Vulnerabilities in web applications are now the largest vector of enterprise security attacks.**

It is necessary for e-commerce application designers to clearly understand all the threats and vulnerabilities landscape and the intricacies of the security of a web site and its underlying supporting infrastructure and to master the techniques used to counter the various threats. Non-Web applications run mostly on local networks and, therefore, the security threats are limited and can easily be isolated and contained. The network infrastructure are under the control of system administrators and it is easy to prevent any sniffing of the information going through the network using some well understood intrusion prevention and intrusion detection systems implementations. Equipment is usually locked in secured areas and access to them is restricted and monitored. The access to the data and software residing on servers is controlled by access control measures/ implementations, including the popular two-factor authentication login procedure. For example, to login, a user must provide a password, and the device from which a user tries to login must be registered as a valid access point. In conclusion, most non-Web applications run on controlled infrastructure and restricted access; therefore, security threats are reduced to a comfortable level.

E-commerce applications, on the other hand, run on external infrastructure. Communication between customers and the e-commerce site is done through predominantly unsecured devices and network links. It is safer to view these infrastructure as unsafe and, therefore, take every possible step to prevent security attacks, rather than assume any false comfort level of security. Furthermore, the customers and the e-commerce servers are not located in the same building or network controlled by the same entity. Consequently, subjects like authentication and nonrepudiation are major concerns. Ehikioya and Olukunle provide a detailed examination of these concepts and recommend their deliberate integration into the design of a framework for e-commerce security. The two most critical design issues regarding security in e-commerce applications are network security and the transaction security.

**Network security**

Data travel through different media that are not under the control of the e-commerce site’s management. This situation poses some challenges in the security and safety of an e-commerce application. Designers must ensure that despite this handicap, data can travel safely with a minimum risk of being intercepted. All protection techniques are computationally demanding and, therefore, it is important to classify data when it is necessary and to make it available when it is not necessary. For example, during shopping, it might not be needed to protect the transfer of shopping cart information. On the other hand, all transfers containing customer information or payment information must be secured as much as
possible. To effectively attain this functionality, the Adaptive Secure Protocol (ASP) proposed in Reference 100 becomes relevant. According to Tak et al.,

\[ \text{The Adaptive Secure Protocol dynamically adapts the security level based on the nature and the sensitivity of the interactions among participants. The security class incorporates the security level of cryptographic techniques with a degree of information sensitivity.} \]

The traditional protocol used to secure communications channels during data transfer is the Secured Socket Layer (SSL) protocol. This is considered the first level of security. A new protocol, Transport Layer Security, has been developed. Designers must either be competent in these technologies or hire experts to incorporate these technologies into their designs. Figure 2 shows the relationship of the application layer, SSL, and the Communication layer.

**Transaction security**

The main concern of the SSL protocol is to ensure that what is received is what was sent. Once that has been achieved, there is still no proof that the person or entity is what it claims to be. This is referred to as the authentication issue. Furthermore, one of the parties can deny being an actor in an e-commerce transaction. This is the nonrepudiation issue. e-Commerce servers collect a large amount of information about each customer. The company responsible for the e-commerce site must take appropriate action to prevent any unauthorized person to access this data. This is the confidentiality issue.

Designers must address all three of these issues (authentication, nonrepudiation, and confidentiality) in order to ensure the success of the business running the e-commerce application. The solutions to these issues involve the use of sophisticated cryptographic techniques. Designers must either be proficient in this area or seek help from specialists. Further, safety and security, and then reliability (including resilience) must be consciously built into the design of e-commerce applications to account for the atomicity, consistency, isolation, and durability (ACID) properties required of e-commerce transactions. The challenge, however, is that many e-commerce applications designers ignore or fail to treat appropriately these key nonfunctional requirements due to their lack of knowledge or experience in modeling these dependability properties of an e-commerce application.

### 4.1.2 Communications and networking

This area deals with the networking issues specific to the design of e-commerce applications. Recall, Web applications, including e-commerce, are network intensive. Thus, if a team made up of network administrators and technicians is in place, designers can safely assume that the physical security of the hardware is maintained and protection, such as firewall, has been installed.

A number of issues are of prime concern to designers. e-Commerce applications are very network intensive and, therefore, factors such as performance and bandwidth are critical as well as network security discussed previously. Once the network administrators have established the best possible environment, designers must find a way to develop their applications within these limitations. The network security, although vital, is not enough by itself for e-commerce transactions but transaction security mechanisms consciously designed into the application itself is crucial.

The bandwidth limitation is more serious in the “last mile” of the connection because bandwidth is smaller on the edges of the network than on the backbone. This limited capacity has a significant impact on the design of web pages. Music and video, as well as heavy graphics/pictures, must be limited, or even better, not placed at all on the web pages of an e-commerce web site. The heavy presence of these media items reduces the capability of the site to present some products adequately. It is important to note that page-loading speed is the most common complaint among customers. If the loading of web pages is too slow, customers will most likely switch to another site due to customer poor quality of user experience. In particular, with the prevalence and ubiquity of mobile devices, often with limited power and small display real estate footprint, the careful design of web pages is vital in order to capture the mobile commerce user segment.

**FIGURE 2** Application layer, Secured Socket Layer (SSL), and communication layer logical relationship
The network architecture of the Internet is an issue which designers do not have much control. The architecture of the Internet is heterogeneous. It includes, for example, wireless workstations which are difficult to locate and have usually very limited computational power. The location poses a problem when the system tries to authenticate a customer. The IP address representing the customer can vary depending on which base transceiver station the wireless computer is connected to. Tracking the customer by IP address alone, which is one of the four methods described by Thomas, is almost impossible. Other means of tracking customers such as cookie or tag propagation must be used. The limited power of [mobile] devices (such as mobile phones, personal digital assistants [PDA], etc.) forces designers to either offer two versions of their Web site, one with regular content and the other with limited graphic content, or restrict completely the use of graphics to small size pictures only when absolutely necessary. Good quality user experience mandates that web pages be properly designed to compensate for low bandwidth, low power of devices, and so forth.

The stateless characteristic of the Web protocol is one of the major issues in designing an e-commerce site. A stateless protocol does not maintain session information; rather every page opened is considered a new and isolated transaction. This type of protocol is very simple but neither suited for nor designed for e-commerce. e-Commerce applications designers must use “tricks” such as cookies to artificially maintain session parameters. This creates a very particular style of design where each page passes parameters to the next page. Parameters can also be saved in session variables on the server or cookies on the client. This method of designing applications makes retraining of designers a mandatory requirement to help them acquire the requisite skills to move from a local Windows environment to a Web design environment.

4.1.3 Languages and tools

One of the early high-level design decisions is that designers must decide what languages/tools will be used to develop their applications. e-Commerce applications usually require a wide range of languages or tools. Perl, Java, ASP.NET, Javascript, PHP, and HTML are some of the most common. The choice of one language over another is a difficult one because the designer must anticipate future changes. In addition, new languages are emerging at a rapid pace, and no one can comfortably guess their longevity, popularity, and acceptance. Besides, some modern languages, such as ASP.NET/VB.NET are linked to a particular operating system (in this case to the Windows operating system) and do not work with other platforms. This protectionism/propriety property or nature must be taken into account to avoid incompatible technologies. Most of these languages do not provide the full complement of functionality of a traditional programming language; for example, some do not have a debugger or an integrated development environment which is essential in program development.

In traditional applications, programmers are experienced in some specific programming languages and they stick to these languages for years. Web applications development is more demanding, fluid and dynamic, and the retraining of programmers (and other developers) is an absolute necessity. Designers must take these into account when designing Web applications.

4.1.4 Database

e-Commerce applications are by nature network-intensive, dynamic, distributed, and client-server applications. Relational databases are almost universally used to store data. Relational Database Management Systems such as DB2 from IBM, Sybase, Microsoft Structured Query Language (SQL), or MySQL are the most common DBMS found. Manipulation of data is done with the SQL. DBMSs have been used for years in traditional applications and offer the robustness necessary to ensure proper transaction processing. Unfortunately, Thomas observed that:

The Web was never designed to be a transaction-oriented environment.

Workarounds have been found to allow database web applications to interface with DBMS. e-Commerce transactions must conform to the inherent ACID properties of transactions. Out of these four ACID properties, atomicity is the most difficult to achieve in an e-commerce application execution environment, and consequently the consistency. As an example, during the checkout process, items bought are transferred from the shopping cart to the order table in the database. Doing so will actually split the checkout transaction into two insert transactions. This could potentially put the database into an inconsistent state. To prevent this, a number of workarounds have been designed. Microsoft,
for example, has added a Transaction Server to its suite of software. This server service works on top of the web server and allows web pages using the ASP.NET/VB.NET scripting language to submit transactions by including the command \texttt{TRANSACTION = REQUIRED} code at the beginning of the transactional ASP page. According to Mihalcea,

\begin{quote}
From a database perspective, the atomicity is a fixed property, but everything else may be traded-off for performance/scalability reasons.\end{quote}

Durability is better off untouched in e-commerce systems because of the need to meet the enterprise system business requirements that mandate durable transactions when a transaction succeeds.

Increasingly, nonrelational database model, such as noSQL, is gaining prominence as the underlying data storage mechanism in some popular e-commerce systems. For example, MongoDB, a cross-platform scalable and flexible document-oriented database with support for query and index functionalities, is being used by many well-established organizations (such as HSBC, eBay, Uber, The Gap, and Urban Outfitters, etc.). Interoperability support between applications and data sources is a tremendous challenge because each application/data source comes with a unique interface and value proposition. To function like the standard SQL-based relational database, some workarounds have emerged for noSQL functionalities. CData Software Inc. succinctly posits that:

\begin{quote}
While SQL does not marry perfectly with NoSQL structures, SQL-based drivers based on established data standards like ODBC, JDBC, and ADO.NET provide a powerful and ubiquitous bridge between NoSQL and modern and legacy applications alike.\end{quote}

NoSQL may offer better performance, scalability, availability, affordability, and flexibility to meet the above needs of the customer because NoSQL databases make managing product, order, and customer information relatively easier, potentially freeing merchants to collect more shoppers and sales data. Merchants could then analyze that data more readily and implement innovative ecommerce solutions that genuinely improve the shopping experiences and could improve management of big data, thereby gaining a competitive advantage.

e-Commerce applications generate a large number of transactions. This number is often too much for a single server. Database designers must often use distributed databases to solve this problem. The distributed model is much more difficult to work with but allows a greater throughput by relocating most commonly used database resources (tables) closer to customers. However, the customers of an e-commerce merchant may be from other continents and very far from the merchant. Another technique used to improve the data availability is the replication of the data. Duplicate copies of the same data are replicated on multiple servers. These two techniques are not special to e-commerce but were already used in non-Web applications.

e-Commerce applications are often designed on top of existing applications such as accounting or financial applications. These legacy systems could be built with older technologies which are not easy to integrate with.

\section*{4.2 Development team issues}

Software companies, which have been developing conventional software for a long time, have well-trained teams of developers. It might seem that these companies could tackle any software project with ease. It is certainly not true for e-commerce applications. The teams required for these e-commerce applications projects are heterogeneous. Graphic designers, responsible for the visual aspect of the application, and marketing specialists, responsible for the presentation of products to customers, must also be part of the team. These team members must likely do not have software engineering backgrounds. This heterogeneity could lead to different points of view, which might contribute to a better design but a less efficient development team. Part of the team focuses mainly on the technical side, while the other part is more concerned about aesthetics and other conceptual designs, thereby adopting a two-part design process approach. A user interface that satisfies the requirements of both parties might be more difficult to find. This type of conflict creates a much more difficult environment to manage.

Project managers face other challenges. They must take into account not only the heterogeneity of a larger team but also the impact of new technology on the project schedule. New technology is an inevitable part of an e-commerce application, whether it is in the form of new multimedia, new payment methods or a new programming language. The lack of historical data and experience in these technologies makes it difficult to predict the time required to complete
project tasks, which affects the prediction of delivery times. The time constraints are critical because they have a direct impact on the e-commerce store (e-store) sales record. This is referred to as the “time to market” a product. The time constraint is so important that some authors use the word “immediacy” for these applications. This feeling of immediacy should not be transformed to a feeling of emergency and put pressure on the team. This would lead to a quickly and poorly designed application. Management plays a crucial role in the organization of any software development effort, thus, the role of management directly affects the software development. Ahmad et al. examined management issues related to time, finance, manpower, and management itself in software development. They recommended that software managers should have both soft skills and technical knowledge in order to effectively handle different resources (such as time, finance, people, hardware, and software) during software development. These issues impact on the software development process and affect the efficiency of the management.

Project managers can take advantage of charting techniques such as the Gantt chart and Program Evaluation Review Technique (PERT). A Gantt chart shows sequence and time of tasks. A PERT chart provides a very precise method for scheduling tasks as it allows project managers to easily determine the critical path. Each task time is reported on an edge of a graph representing the PERT chart. The expected task time is calculated as follows:

\[
\text{Expected task time} = \frac{T_{\text{opt}} + 4 \times T_{\text{ml}} + T_{\text{pes}}}{6}
\]

where \(T_{\text{opt}}\) is the estimated time for the task in the most optimistic conditions, \(T_{\text{ml}}\) is the most likely time for the task, and \(T_{\text{pes}}\) is the most pessimistic time for the task. This type of chart is very convenient for e-commerce projects, where a wide range of technologies are used.

Training or retraining is a critical issue for project managers. Technology is evolving rapidly and employees do not stay on the same position for a very long time (i.e., they leave the company or get a new position). These two factors lead to a constant need for training. It is time consuming (a few weeks, a year for each employee) and expensive. This puts a lot of stress on companies and has a negative impact on the project schedule.

The challenge is finding people who have both the experience and the ability to apply that experience to new situations; especially in transitioning from traditional nonweb-based environments to the web environment of e-commerce applications that require very high degree of interactivity. According to Laudon and Traver,

\[
\text{The development of e-commerce requires a great deal more interactive functionality (e.g., ability to respond to user input, order taking, clear credit card transactions, consult price and product databases, etc.).}
\]

The interactivity (i.e., interaction with the user) feature of e-commerce applications allows for a two-way communication between the merchant’s application and consumers. The consumers engage in processes that dynamically adjust the experience to the individual, and make the consumer a coparticipant in the process of fulfilling a purchase transaction. This requires a new set of skills that must be appropriately used in the design process of e-commerce applications. Besides, the distributed nature of e-commerce transactions requires multitiered system architecture to handle the processing load. These systems are often complex and challenging to design appropriately. It requires appropriate knowledge and experience to achieve an appropriate and adequate successful design.

All these team and management issues must be examined very carefully to create the most efficient development team. This is ultimately the responsibility of the project manager.

### 4.3 Navigation and interface design issues

The user interface of non-Web applications and the navigation between windows/forms is not a major concern to user interface designers because well-known design patterns exist and can be used as references to efficiently build applications. Web applications, in particular e-commerce applications, do not have a lot of past experience to rely on. A number of surveys by Gehrke and Turban and Godwin and Marquis present a reflection of the trends before 2000. Since then, many technologies have disappeared and other new ones have emerged, rendering parts of their studies incomplete or irrelevant. However, several recent empirical studies agree with their findings. A number of basic concepts (page loading, navigation efficiency, business content, security, and marketing/consumer focus) unique to e-commerce applications are critical for the design of the navigation and the user interface. Dwarai and Kamala state that ease of navigation, navigation time, page download/page-loading speed, and use of multimedia to improve Web
The customer concept is fundamental in e-commerce applications and it comes with its challenges that are customer-centric. Web applications do not have end users, as they are called in non-Web applications, but customers. End users are consulted during the user interface design. Customers, on the other hand, are not consulted, but they buy products or services and, therefore, are critical to the survival of the business. Customers have different backgrounds, cultures, and possibly live in different parts of the world. This diversity makes it difficult to build a universal user interface and a navigation that will please everybody. According to Jahng et al., e-commerce applications developers should:

… make provisions to deal with the heterogeneous nature of both products and consumers.

Customers around the world must be able to access Web pages with a reasonable page-loading speed. Page-loading speed is a common complaint by customers.

e-Commerce merchants are striving to provide more relevant, more personalized, and more compelling shopping experiences to their customers. According to Roggio, customers want, for example, relevant, specific, and personalized search results delivered in milliseconds, regardless of their geographic location or device. They also want messaging, content marketing, product suggestions, and a buying experience tailored for them.

Using efficient styles and techniques for page design and content can reduce response times by a factor of about 2 to 5 seconds. Page generation speed can be enhanced by segregating computer servers to perform dedicated functions. Similarly, using specialized content delivery networks or increasing local bandwidth can significantly enhance the speed of page delivery.

The navigation of an e-commerce site must be viewed by designers as similar to a real brick and mortar store. The visual impression left by the navigation through the site should be comparable to the visit to a store. Online designers must place promotional products for easy discovery and access. As an example, people with minimum computer skills should easily master the management of the shopping cart just as customers of a physical store should easily operate the mechanical shopping cart without being experts in order to compete and remain in business. The same principle applies to online stores. In fact, it is even more critical in the case of online stores because the customer is isolated and cannot ask for help.

The marketing influence of Web applications and e-commerce systems cannot be overlooked. The interface design must take into consideration e-commerce applications peculiar marketing requirements. This is so important that Gehrke and Turban claim:

A Website should be evaluated as to how well it markets its products and services.

A typical way to market products on the website is by adding banners at the top of the screen. These banners pose new challenges to designers. Often they are animated and may come from external sources. Counters must be set up to track how many people click on these banners. The placement of adverts on the web page is critical, and the design of the rest of the page must be done around them. All these design challenges must be addressed in order to meet the marketing requirements. Web pages can be viewed as tools to help customers make purchase decisions and, also as billboards for companies selling their products. Therefore, careful design decisions must be made in order to ensure their usefulness and effectiveness.

The nature of the electronic store design can significantly affect the decision of a consumer to shop online. A well-designed electronic store front can attract consumers to the store and increase their willingness to purchase items on the store. Such a design must take into account customer requirements and functions necessary for transaction processing and store management that must cater to the perspectives of the consumer and the electronic store merchant. Meeting these needs adequately has created a challenging arena for e-commerce applications development due to the diversity of issues to handle, from varying customers with varying cultures, power and size of devices (including smart phones, PDAs, etc.), (nature, types, and placement of) advertising materials, security, etc.

4.4 Detailed design issues

The detailed design process results in a set of design artifacts (documents) that precisely and accurately specify how to realize the functionalities of a system. These outputs are mostly diagrams when a model-driven approach of the design is used. However, these may be represented in code of the program where agile development methods are used.
Here, the designers are concerned with: (a) the structural organization of the system and shows the system and its relationship with other systems or the structure of the data that are processed by the system; (b) the behavioral or dynamic behavior of the system and how it responds to events; and (c) the interactions between the system and its environment, or between the components of the system. These design activities—architectural, behavioral, and interactions designs—are challenging and many issues (e.g., architectural design decisions) that could lead to poor design solutions often arise. The detailed design must meet the performance, security, safety, availability, maintainability, and usability requirements of the system.

Establishing the right relationship between cohesion and coupling is crucial. Cohesion is a measure of the degree to which the elements of a module are functionally related; thus, cohesion is a measure of the strength of relationships within a module. Coupling is the measure of the degree of interdependence between modules. A software designer, therefore, should design the modules of a software application with the goal of high cohesion and low coupling because improperly modularized designs would result in nontrivial enhancements in future. A problem with coupling and cohesion is the difficulty of quantifying them systematically.\textsuperscript{122} Although they provide measures that can be used to help the designer with assessing the relative merits of possible options, the challenge of finding ways of measuring them in an acceptably objective way remains.

The detailed design descriptions are used to generate system implementation equivalence, and therefore, the designs must be both correct and complete\textsuperscript{18,93} in addition to design clarity and understandability, the right level of abstraction, and maintainability properties. However, analytical methods for software design are generally unavailable. Software designers often aim at producing optimal designs, but quantitative criteria for optimal software design are largely unknown, therefore, mostly relying mainly upon judgment and experience with development standards still largely missing as the attendant consequence. According to Marsic,\textsuperscript{14}

\textit{Unfortunately, at present software engineering discipline is unable to precisely specify the quantitative criteria for evaluating designs. Some criteria are commonly accepted, but there is no systematic framework.}\textsuperscript{14}

The transformation of an informal design to a detailed design is challenging as incompleteness, incorrectness, and ambiguity could easily creep in due to lack of experience (and/or knowledge) in the use of formal design methods by most e-commerce applications developers.\textsuperscript{1,2,9} By using a formal design method, one can reason about the correctness and completeness of the designs, which enhances design clarity, which in turn eases implementation, debugging, testing, documenting, and maintenance of the resultant software product, and ultimately saves overall development cost. As aptly stated by Foster,\textsuperscript{123}

\textit{Every software engineer knows how to write programs; few can excellently design software.}\textsuperscript{123}

This deficiency often leads to many unintended consequences that could negatively impact on the completeness and correctness and the overall quality of the resulting software design.

The stateless environment that e-commerce applications execute in presents some challenges to the developer in his/her ability to track or collect some important data/variables required for wholesome activities of an e-commerce system. Event-driven applications, such as Windows-based applications, give control to the user. This is possible in part because applications keep track of the status of each process or windows. This is, however, not the case with Web applications such as e-commerce applications. These applications use the Web browser environment, which is a stateless environment, for their execution. Therefore, these variables must be kept by other means in order to provide the same functionality. Designers must rely on cookies, session variables, and application variables to maintain the states of the applications.

The implementation languages to use are a source of challenges to developers. The decision about the implementation language to develop an application is often usually made at the high-level design stage. Detailed systems and software designers must, therefore, be knowledgeable in the chosen language. Most languages used today to develop nonweb applications are object oriented. This is not the case with Web languages. The designers must juggle with both types of languages (object oriented and nonobject oriented).

A key design challenge often overlooked in the development of e-commerce applications is when and where to use formal design tools in the design process, particularly the use of formal design notations. Generally, the use of formal design notations/tools in software development is largely negligible, and for e-commerce applications in particular is nearly
nonexistent. Experience in use of formal design notations is generally lacking. Ehikioya\textsuperscript{9} provides a lucid examination of this situation and points out numerous benefits of using formal specification methods in reasoning about e-commerce transactions. Interested readers should see the following additional resources\textsuperscript{1,2,4,11,21,43,124,125} for detailed treatment of the use of, and need for, formal specification methods in e-commerce transactions.

4.5 | Environment

e-Commerce is a disruptive technology and has changed the way we do business and live. To adapt to this new environment, businesses need to define a new code of ethics, as according to Laudon and Traver\textsuperscript{3}:

\[\ldots e\text{-}commerce disrupts existing social and business relationships and understandings.\]

These environment factors are business, government, and international market. We briefly examine these factors while isolating the design challenges in each.

4.5.1 | Business environment

The Internet is a less regulated environment for business, and businesses trying to establish e-commerce presence face difficult challenges. Designers of e-commerce sites must be fully aware of the issues to ensure the viability of these businesses. The volatility of the e-commerce market forces businesses to react quickly, and therefore, the time available to market e-commerce applications is very short.\textsuperscript{126} This time constraint, referred to as immediacy,\textsuperscript{3} forces designers to look for the best development methods to achieve their goals. The constraints are made even more difficult by the lack of well-established design standards and the instability of development technology. In areas where standards are already defined, new technologies and needs make them quickly obsolete.

4.5.2 | Government environment

Governments face a difficult dilemma as to whether e-commerce should be regulated or not. A number of guidelines and ethical rules try to protect the consumers and help designers to provide better services to customers. Most governments have passed laws\textsuperscript{127-129} to protect the information provided by customers in online sales. Designers must be aware of these laws and regulations and design their applications accordingly. Privacy is a fundamental principle in our society. Laudon and Traver\textsuperscript{3} define privacy as:

\[Privacy \text{ is the moral right of individual to be left alone, free from surveillance or interference from other individuals or organizations, including the state.}\]

From this definition follows information privacy which concerns the privacy of information provided by customers during online sales.

4.5.3 | International market

e-Commerce makes it now possible for (most) businesses to sell their products on the international market. Without proper international legislation in place, selling internationally can be a risky business. It is common to read about businesses that got into difficulty or even closed after problems with international sales.

The developers of e-commerce applications must consciously plan these environment dependent issues into their designs so that they can be adequately considered and provided for through appropriate mechanisms such as privacy preserving processes, support for nonrepudiation, maintenance of standards of the user interfaces and process description. These are a great challenge to the uninformed or unenterprising designers.
5 | DESIGN CHALLENGES IN A NUTSHELL

We summarize below, in a tabular format, the essential elements and their relationships with design goals, e-commerce functionality, and design features. Table 1 identifies the key challenges in the design of e-commerce applications and juxtaposes these with e-commerce applications functionality to ensure good customer quality of user experience and the overall quality of the resultant software application.

Ehikioya and Olukunle emphasize the need to develop secure (providing a framework for guaranteeing security of customers' and merchants' data), reliable (producing exact results in a consistent manner), and error-free e-commerce applications that guarantee the correctness (producing consistent and correct operational and transactional results) of online transactions. Guaranteeing the correctness (entails various stages in e-commerce transactions producing consistent results) of an e-commerce application is difficult, due to many situations in e-commerce transactions where errors could emerge, resulting in unexpected system failures. The application of mathematical and logic-based formal specification techniques in developing e-commerce applications is widely believed to result in reliable and dependable applications. Although several specification methods have been identified as suitable for capturing e-commerce systems requirements, these extant specification methods only have functionalities that can be used in modeling some components of e-commerce systems. However, neither of these existing formal specification methods has all the requisite functionalities to model all the components of an e-commerce system. It suffices to reiterate that provision of a secure environment for e-commerce transactions is paramount because of the critical tasks of business processes in e-commerce transactions.

A detailed discussion of the application of the traditional transaction atomicity concepts in determining correctness of e-commerce transactions is available in and will not be discussed further in this article.

6 | DISCUSSION

Design challenges manifest in several ways. The design issues in e-commerce applications development are pervasive; ranging from the heterogeneity, openness, concurrency, reliability, performance, transparency, and scalability to security, failure handling, nonrepudiation, and quality of service. We provide a longitudinal analysis of the design challenges in developing e-commerce transaction systems and characterize these challenges/issues vis-à-vis e-commerce applications requirements. The longitudinal approach provides insights into suitable and sustainable design strategies for e-commerce merchants to enhance consumers’ perceived relative advantage and acceptance, and reduce perceived complexity in online shopping.

The research findings delineate a significant transition of e-commerce design issues in the e-fulfillment strategies among online merchants. Evidence is found for the desirability for the development of service metrics as well as strategic shifts in the design of e-commerce applications to enhance the overall quality of user experience of customers during their fulfillment of online shopping activities. Therefore, using techniques that enhance page response times/page delivery times is crucial, which supports the views enunciated in References 74, 84, 85, and 87 thus underlining the crucial role of navigation speeds and page loading times.

We provide a retrospective look at database support for transactions while x-raying the relevance of the unstructured data models (such as noSQL) to support e-commerce transactions systems. We agree with CData Software Inc. and Mihalcea that handling the database issues is necessary to enable e-commerce transactions deal with the large volume and variety of data types, as well as the often varying data that they generate and support computational results correctness and transaction execution correctness. The resolution of the database design challenges should provide and guarantee safe environment for supporting e-commerce activities.

Project management challenges are many; ranging from team membership dynamics and heterogeneity, impact of new technology on project schedule due to lack of historical data and experience in the use of the technologies. Evidence from our research indicates that project management requires both soft skills and technical knowledge. Therefore, in order to effectively handle the different resources (such as time, finance, people, hardware, and software) during software development, the project management-related issues which impact on the software development process and affect the efficiency of the management must be adequately handled. This is consistent with the views of Ahmad et al. and amplified by the high degree of interactivity and distribution of e-commerce application transactions. This situation suggests that substantial errors/problems might emanate from the improper handling of project management issues.
| Design goal | e-Commerce functionality | Recommendation to achieve design features |
|-------------|--------------------------|-------------------------------------------|
| Performance | Performance of the system • Throughput, and • Dependability issues | • Appropriate level of coupling and cohesion • Optimal algorithms • Appropriate application and network architectures • Accommodation for the power and size of real estate of mobile devices |
| Security    | Security of transactions | • Use appropriate algorithms for handling the information security mechanisms • Implement appropriate access control policies/mechanisms • Design appropriate logging mechanisms for handling nonrepudiation • Appropriate use of SSL and other Web protocols • Enforce network and transaction security • Provide and guarantee secure environment for e-commerce activities • The transactions should be secured o maintain the integrity of the application system’s data o guarantee minimum security by implementing security modules for authorization, authentication (verification and validation) of users and their access rights, and the encryption of users’ sensitive data such as personal data and credit card information o maintain the confidentiality of confidential information • Maintain and guarantee the integrity of communications between users |
| Safety      | Safety of transactions and execution environment | • Design of appropriate execution safety mechanisms • Provide and guarantee safe environment for e-commerce activities • Use of appropriate transaction commands |
| Availability| Availability of systems • Available for use by users whenever desired (over a given period) | • Appropriate maintainability features • Understandable designs • Appropriate cohesion and coupling • Appropriate safety and security mechanisms |
| Maintainability | Maintainability of systems | • Appropriate modularization • Appropriate coupling • Separation of functionalities • Structuring of operating systems and other system dependent components • Extensible — A system is extensible if it is designed to easily allow the addition of new features at a later date, for example, through the use of extensible interface designs |
| Usability   | Usability of systems • Easy to use by customers | • Good interface design • Appropriate and intuitive navigational buttons • Good human-computer interaction design |
| Correctness and completeness | Correctness of transactions execution results | • Use of standardized design methods • Formal design methods • Requirements traceability • The system’s (design) specification must guarantee the following correctness types: o computational results correctness o transaction execution correctness—begin and abort transaction atomicity construct o operational (sequence) correctness—order of transaction steps/serialization o payment atomicity correctness o delivery atomicity correctness o product atomicity correctness |
| Clarity and understandability | Low cognitive demand • Easy to use by customers • Simple and easy to understand so minimal implementation defects | • Use formal design methods • Intuitive design models • Responsive Web design • Use universal standards based on the Internet and the Web • Adopt design that lowers the cognitive energy (ie, the mental effort) required to complete a task |

Abbreviation: SSL, Secured Socket Layer.
We found the adoption of software engineering or development methods, such as MBSD, formal methods, and standardized software development methodologies, and so forth in the development of e-commerce transactions systems instructive. Although using these methods is generally challenging, they provide measures that can be used to help the designer with assessing the relative merits of possible options. An e-commerce application software designer, therefore, should design the modules of an e-commerce application with the goal of high cohesion and low coupling because improperly modularized designs would result in future nontrivial enhancements. Besides, using formal methods is consistent with the findings in References 1, 2, and 9 because the detailed design descriptions undergo transformations to generate system implementation equivalences that must be both correct and complete, have design clarity and understandability, with the right level of abstraction, and demonstrate maintainability properties. The orthogonal use of informal designs and transforming them into detailed designs, most often, could result in incompleteness, incorrectness, and ambiguity due to lack of experience (and/or knowledge) in the use of formal design methods by most e-commerce applications developers. The benefit of using a formal design method is that one can reason about the correctness and completeness of the designs, which enhances design clarity, which in turn eases implementation, debugging, testing, documenting, and maintenance of the resultant software product, and ultimately saves overall development cost.

The largely negligible or nearly nonexistent use of formal design notations/tools in e-commerce applications development in particular, creates its negative side effects. We have answered the question of when and where to use formal design tools in the design process, an issue often overlooked in the development of e-commerce applications. Designers should adopt formal design methods early in the design process in order to ensure that the system's (design) specification guarantees both the computational results correctness and transaction execution correctness, and operational (sequence) correctness—order of transaction stepsserialization. This position agrees with the views canvassed in the resources 1, 2, 4, 9, 11, 21, 43 for the use of, and need for, formal specification methods in e-commerce transactions.

Obviously, an e-commerce transactions system that is intuitive in its design, user friendly and easy to use, and trusting is more desirable and attractive for online shoppers because of the low cognitive demand. Our findings indicate that factors such as perceived usefulness, perceived ease of use, customer risk perception, customer support, customer trust, promotion, information quality, transaction security, website design, and brand image significantly influence customer satisfaction and customer loyalty when shopping on an e-commerce website. Interestingly, e-commerce service quality (particularly related to security and payment method) is the principal element affecting customer satisfaction with e-commerce system. In addition, information quality, system quality, service quality, product quality, delivery quality and perceived price affect the overall user quality of experience in using e-commerce applications. These factors positively influence customer satisfaction and trust, which corroborates the established literature that capture e-commerce web site quality using the TAM model. These results correlate with the positions of Gelard and Negahdari, 74 Chakraborty et al. 84 Vollset, 87 and Dwairi and Kamala 85 that ease of navigation, navigation time, page download, and use of multimedia to improve Web site’s visual appeal are the attributes that influence the design factors in evaluating the quality of B2C e-commerce Web sites. A website’s page speed affects a merchant’s revenue because page-loading speed is a major contributor to page abandonment.

The primary goal of a good software design is to satisfy the software quality factors in consonance with the ISO/IEC-9126-1:2001 specification. e-Commerce applications designs should inherit these properties in order to provide good quality of user experience for online shoppers. Our recommendations for achieving the desirable design features (see the third column in Table 1) are essential ingredients for meeting the design goals of e-commerce application functionalities. It is our hope that when software designers adopt these recommendations, the resolution of the key challenges in the design of e-commerce applications becomes easy and manageable; thereby resulting in e-commerce applications functionalities that ensure good customer quality of user experience and the overall quality of the resultant software application.

7  |  CONCLUSION

In the 1990s, designers faced the challenge of switching from the DOS world to the Windows environment which gives more control to end users. Applications developed with this environment are known as event-driven applications. While some designers never made it to this new environment, others adapted without any problem.

In recent years, the Internet brought its share of difficulties. A new adaptation was necessary. The Web application model can be viewed as an extension of the event driven model. In the latter, users are given more control on what action they want to perform and when they want to do it. Despite this relative freedom, they do not have the choice of
the location from which they execute their commands. Web applications have brought that new dimension. Customers can now choose when, what and where they use the application, that is, do their shopping. The addition of this new dimension puts a lot of stress on the network infrastructure supporting Web applications. Security is also an issue. Network administrators do not have the visual control anymore. Other techniques, such as cryptography, must be used to confirm the identity of customers. e-Commerce web sites also have to be able to provide proof of their identity to online customers.

Customers using the e-commerce model to buy products or services are evolving in a much larger environment than they were in traditional applications. They may be subject to government regulations and controls. e-Commerce applications bring more critical design issues than any other application. Designers must carefully choose their development methods and tools, and they must try to use formal methods whenever it is possible or required. In this article, we identified the key challenges in the design of e-commerce applications and juxtaposed these with e-commerce applications functionality to ensure good customer quality of user experience.

Further, we examined the challenges in the design of e-commerce applications/systems to accommodate the increasing sophistication of e-commerce applications and the increasing demands of users for higher quality of user experience.

This study of the design issues in e-commerce systems development deals with issues that ultimately affect the overall system quality and hence the factors that affect customer satisfaction of e-commerce website. Thus, this article has both academic and practical implications for e-commerce website management, design, and performance enhancement.

In particular, we focused on three broad main challenge areas: (a) how technologies, frameworks, and architectures influence and cause e-commerce applications developers to take certain premature and undesirable design decisions, (b) the interactivity and complexity of e-commerce applications which requires developers to provide several models rather than a single model for the same problem, and finally (c) impact of existing quality assessment criteria for e-commerce applications design.

Our article theoretically contributes to software development/software engineering literature. While we examine the relevance of acceptance of technology model and its variants to assess users' loyalty and satisfaction at a web site for designing e-commerce applications interfaces, the core of our contributions deals with the design issues that are germane to software development. Our work supports theoretical findings such as MBSD, formal requirements and design specification of e-commerce systems, security concerns, tools support (including databases and programming languages), interaction design of e-commerce applications interfaces, and software development management issues. Also, we elevate the need to always evaluate the environment effects and implications of e-commerce because e-commerce is a disruptive technology, inherently distributed, and cross jurisdictional, thus supporting the need for the protection of information.

We omit a quantitative framework for evaluating software designs because of the complexities involved since doing so would distract us from the main focus of this article. However, this omission does not detract or minimize our assessment of the design issues in e-commerce transaction systems. Also, a characterization of designs suitable for e-commerce transactions and systems is not considered in this article. Ehikiyoa reports some aspects of this problem and, therefore, it is unnecessary to repeat same in this article.

Although the longitudinal analysis of the design challenges in developing e-commerce transaction systems provides insights into suitable and sustainable design strategies for e-commerce merchants to enhance consumers' experience in online shopping, we did not implement our design recommendations. It would be educative to implement our recommendations in a prototype B2C e-commerce platform to gather empirical data to assess the extent of the elimination of these challenges/issues vis-à-vis e-commerce applications requirements and the attainment of the design quality goals.

We plan to extend this work in several directions. First, we hope to provide a quantitative framework for evaluating software designs. Although this is by no means a trivial extension because of the complexities involved, such a framework would help in the easy and quick evaluation of the several candidate designs that could be produced instead of relying on common sense and intuitive criteria. Second, by establishing a clearly identifiable pathway that links software design and the principle of network security and application safety and security, e-commerce applications designers/developers would have access to a crystallized body of knowledge to assist them in making the right design choices in respect of network security and application safety and security. Finally, we plan to provide a characterization of designs suitable for e-commerce transactions and systems.

ACKNOWLEDGMENT
The authors thank the anonymous review panel for their many helpful suggestions that enriched this final revised version of the article.
CONFLICT OF INTEREST
The authors have no conflict of interest with respect to this article.

ORCID
Sylvanus A. Ehikioya  https://orcid.org/0000-0002-9743-2410

REFERENCES
1. Ehikioya SA, Barker K. A Formal Specification Strategy for Electronic Commerce. Proceedings of International Database Engineering and Applications Symposium (IDEAS ’97), Montreal, Canada, IEEE Computer Society Press; April 1997:201-210.
2. Ehikioya SA. A formal characterization of electronic commerce transactions. Int J Comp Inform Sci. 2001;2(3):97-117.
3. Laudon KC, Traver CG. e-Commerce 2018: Business, Technology, Society. 14th ed. Boston, MA: Pearson; 2019.
4. Zhang C, Li X, Hu J, Feng Z, Song J. Formal modeling and analysis of fairness characterization of E-commerce protocols. J Appl Mathem. 2014;2014:138370. https://doi.org/10.1155/2014/138370.
5. Turban E, Outland J, King D, Lee JK, Liang T-P, Turban DC. Electronic Commerce 2018: A Managerial and Social Networks Perspective. 9th ed. Switzerland: Springer International Publishing; 2018.
6. Schneider GP. Electronic Commerce. 11th ed. Stamford, CT: Course Technology, Cengage Learning; 2015.
7. Jelassi T, Enders A. Strategies for e-Business: Creating Value through Electronic and Mobile Commerce—Concepts and Cases. New York, NY: Prentice Hall, Pearson Education Limited; 2004.
8. Kraemer KL, Dedrick J, Melville NP, Zhu K. Global e-Commerce: Impacts of National Environment and Policy. Cambridge, UK: Cambridge University Press; 2006.
9. Ehikioya SA. Specification of Transaction Systems Protocols [PhD thesis]. Winnipeg: University of Manitoba; 1997.
10. Mihalcea V. A Beginner’s guide to ACID and database transactions. Java Code Geeks. 2014; https://www.javacodegeeks.com/2014/01/a-beginners-guide-to-acid-and-database-transactions.html. Accessed September 26, 2018.
11. Tygar TD. Atomicity versus anonymity: Distributed Transactions for Electronic Commerce, Proceedings of the 24th VLDB Conference, VLDB’98, Morgan Kaufmann Publishers Inc., New York, NY; 1998:1–12.
12. Sherif MH. Protocols for Secure Electronic Commerce. 3rd ed. Boca Raton, FL: CRC Press; 2016.
13. Chan H, Lee R, Dillon T, Chang E. E-Commerce: Fundamentals and Applications. West Sussex, England: John Wiley & Sons Ltd.; 2001.
14. Marsic I. Software Engineering. NJ: Prentice Hall; 2012.
15. Whitten J, Bentley L. Systems Analysis and Design Methods. 7th ed. New York, NY: McGraw-Hill; 2007.
16. Davis A. 201 Principles of Software Development. New York, NY: McGraw-Hill; 1995.
17. Ehikioya SA, Olukunle AA. A formal model of distributed security for electronic commerce transactions systems. International Journal of Networked and Distributed Computing (IJNDC). 2019;7(2):68-84.
18. Sommerville I. Software Engineering. 10th ed. New York, NY: Addison-Wesley; 2015.
19. Menasse DA. A Reference Model for Designing an E-Commerce Curriculum, IEEE Concurrency, January-March, 2000.
20. McNay HE, UML for e-Business: New use for Use-Cases, Proceedings of IEEE International Professional Communications Conference (IPCC-2001), Santa Fe, New Mexico, October 24-27, 2001:245-250.
21. Ehikioya SA, Hiebert K, A Formal Model of Electronic Commerce. First International Conference on Software Engineering, Networking, and Parallel and Distributed Computing, May 19–21 2000, Champagne-Ardenne, France, ACIS; 2000.
22. Larman C. Applying UML and Patterns: an Introduction to Object Oriented Analysis and Design and Iterative Development. 3rd ed. Upper Saddle River, NJ: Prentice Hall PTR; 2004.
23. Brünet J, Murphy GC, Terra R, Figueiredo J, Serey D. Do Developers Discuss Design? Proceedings of the 11th Working Conference on Mining Software Repositories (MSR 2014), New York, NY:340-343.
24. Bourque P, Fairley RE(D), eds. SWEBOK Guide V3.0: Guide to the Software Engineering Body of Knowledge. Washington, DC: IEEE Computer Society; 2014.
25. Hummel O, Eichelberger H, Giloj A, Werle D, Schmid K. A Collection of Software Engineering Challenges for Big Data System Development. Euromicro SEAA 2018 in Prague, IEEE; 2018.
26. Sousa L, Oliveira R, García A, et al. How Do Software Developers Identify Design Problems? Proceedings of 31st Brazilian Symposium on Software Engineering (XXXI SBES), Fortaleza, Ceará, Brazil, September 10, 2017.
27. MacCormack A, Rusnak J, Baldwin C. Exploring the structure of complex software designs: an empirical study of open source and proprietary code. Manag Sci. 2006;52(7):1015-1030.
28. van Gurp J, Bosch J. Design erosion: problems and causes. J Syst Software. 2002;61(2):105-119.
29. Silva MCO, Valente MT, Terra R. Does Technical Debt Lead to the Rejection of Pull Requests? Proceedings of the 12th Brazilian Symposium on Information Systems (SBS’16); 2016:248-254.
30. Xiao L, Cai Y, Kazman R, Mo R, Feng Q. Identifying and Quantifying Architectural Debt. Proceedings of the 38th International Conference on Software Engineering (ICSE ’16), ACM, New York, NY; 2016:488-498.
31. Sousa L, Oliveira A, Oizumi W, et al. 2018. Identifying Design Problems in the Source Code: A Grounded Theory. In ICSE ’18: 40th International Conference on Software Engineering. May 27-June 3, 2018, Gothenburg, Sweden. ACM, New York, NY; 14.
32. García J, Popescu D, Edwards G, Medvidovic N. Identifying Architectural Bad Smells. Proceedings of the 2009 European Conference on Software Maintenance and Reengineering (CSMR ’09); March 24-27, 2009, Kaiserslautern, Germany: IEEE Computer Society, Washington, DC; 2009:255-258.
62. Qingfei M, Shaobo J, Gang Q. Mobile commerce user acceptance study in China: a revised UTAUT model. *Tsinghua Sci Technol.* 2008;13(3):257-264.
63. Raesi S, Meng L. Factors influencing to M-commerce adoption in China. *Int J Business Manage.* 2016;4(3):372-384.
64. Atteuquayefo S, Addo H. Review of studies with UTAUT as conceptual framework. *Eur Sci J.* 2014;10(8):249-258.
65. Sharma R, Mishr R. A review of evolution of theories and models of technology adoption. *IMJ.* 2014;6(2):17-29.
66. Lai PC. The literature review of technology adoption models and theories for the novelty technology. *Inform Syst Technol Manage.* 2017;14(1):21-38.
67. Lai PC, Zainal AA. Consumers’ intention to use a single platform E-payment system: a study among Malaysian internet and mobile banking users. *J Internet Bank Commer.* 2015;20(1):1-13.
68. Lai PC. Design and security impact on consumers’ intention to use single platform E-payment. *Interdis Inform Sci.* 2016;22(1):111-112.
69. Napitupulu TA, Aditomo HC. Factors affecting customer loyalty in business e-commerce: a case of Indonesia. *J Theor Appl Inform Technol.* 2015;76(3):386-392.
70. Xiang J, Chen X. Customer Satisfaction of E-Commerce Websites. International Workshop on Intelligent Systems and Applications, May 23-24, 2009, IEEE Xplore: 2009.
71. Lin GTR, Sun C-C. Factors influencing satisfaction and loyalty in online shopping: an integrated model. *Online Inform Rev.* 2009;33(3):458-475. https://doi.org/10.1108/14684520910969907.
72. Sanyal S. Factors affecting customer satisfaction with ecommerce websites—an Omani perspective. *Int J Business Manage Inven.* 2019;8(9):25-30.
73. Eid MI. Determinants of E-commerce customer satisfaction, trust, and loyalty in Saudi Arabia. *J Electron Commer Res.* 2011;12(1):78-93.
74. Geland P, Negahdari A. A new framework for customer satisfaction in electronic commerce. *Aust J Basic Appl Sci.* 2011;5(11):1952-1961.
75. Irantaj G, Huseynov F. Factors influencing customer satisfaction level in an E-commerce platform: a case study analysis of Digikala in Iran. *Int J Manage Appl Sci.* 2018;4(6):19-27.
76. Al-Jahwari NS, Khan MFR, Al Kalbani GK, Al Khansouri SS. Factors influencing customer satisfaction of online shopping in Oman—Youth perspective. *Human Social Sci Rev.* 2018;6(2):64-73.
77. Sobihah M, Mohamad M, Ali NAM, Ismail WZW. E-commerce service quality on customer satisfaction, belief and loyalty: a proposal. *Meditier J Soc Sci.* 2015;6(2):260-266.
78. Hila Ludin IHB, Cheng BL. Factors influencing customer satisfaction and E-loyalty: online shopping environment among the young adults. *Manage Dyn Knowl Econ.* 2014;2(3):462-471.
79. Mahmood SMF. E-commerce, online shopping and customer satisfaction: an empirical study on E-commerce system in Dhaka. *Int J Entrepen Develop Stud.* 2016;4(3):323-338.
80. Khalid A, Lee O, Choi M, Ahn J. The effects of customer satisfaction with E-commerce system. *J Theor Appl Inf Technol.* 2019;96(2):481-491.
81. Lin C-C, Wu H-Y, Chang Y-F. The critical factors impact on online customer satisfaction. *Proc Comp Sci.* 2011;3:276-281.
82. Tabaei Z, Fathian M, Gholamian MR. Effective Factors on Electronic Customers Satisfaction. 2011 3rd International Conference on Information and Financial Engineering. IPEDR Vol. 12, IACSIT Press, Singapore; 2011.
83. Cao M, Zhang Q, Seydel J. B2C E-commerce web site quality: an empirical examination. *Ind Manag Data Syst.* 2005;105(5):645-661.
84. Chakraborty G, Lala V, Warren D. An empirical investigation of antecedents of B2B websites’ effectiveness. *J Interact Market.* 2002;16(4):51-72.
85. Al-Dwairi RM, Kamala MA. Business-to-Consumer E-Commerce Web Sites: Vulnerabilities, Threats and Quality Evaluation Model. 20th International Conference on Electronics and Computers (CONIELECOMP), February 22-24, 2010, Cholula, Mexico, IEEE; 2010.
86. Marszałkowski J, Marszałkowski JM, Drozdowski M. Empirical study of load time factor in search engine ranking. *J Web Eng.* 2014;13(1&2):114-128.
87. Vollsle T. eCommerce Apps: the 10 Most Common Consumer Complaints. Octopus Mercantile Research Report; 2018.
88. Rogers EM. *Diffusion of Innovations.* 3rd ed. New York, NY: Free Press; 2003.
89. Moore GC, Benbasat I. Development of an instrument to measure the perceptions of adopting an information technology innovation. *Inform Syst Res.* 1991;2:192-222.
90. McCall JA, Richards PK, Walters GF. *Factors in Software Quality.* Vol 1. Springfield, VA: National Technical Information Service; 1977.
91. Boehm BW, Brown JR, Kaspar JR, Lipow ML, MacCleod G. *Characteristics of Software Quality.* New York, NY: Elsevier; 1978.
92. International Standards Organization, ISO/IEC 9126-1:2001. *Software Engineering—Product Quality—Part 1: Quality Model.* Geneva, Switzerland: International Standards Organization; 2001.
93. Aggarwal KK, Singh Y. *Software Engineering.* 3rd ed. New Delhi, India: New Age International Publishers; 2007.
94. Menasce DA. Scaling for e-Business. Proceedings 8th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems, IEEE; 2000:511-513.
100. Tak SQ, Lee Y, Park EK, Stach J. Design and Evaluation of Adaptive Secure Protocol for E-Commerce. Proceedings of 10th International Conference on Computer Communications and Networks, 15-17 October 2001, Scottsdale, AZ:IEEE;2001.

101. Thomas B. Recipe for e-commerce. IEEE Internet Comput. 1997;1(6):72-74.

102. Garfinkel S, Spafford EH. Web Security, Privacy and Commerce. 2nd ed. Sebastopol, CA: O’Reilly Media Inc.; 2002.

103. Oppliger R. Contemporary Cryptography. 2nd ed. Boston, MA: Artech House; 2011.

104. Hyde A, Peirano M, Oliver J, Vasiliev D, Wolf A, Gerber J, et al. CryptoParty Handbook. Auckland, New Zealand: Book Sprints Ltd; 2013. https://github.com/cryptoparty/handbook).

105. Ehikioya SA, Zhang C. A real-time multi-agents architecture for E-commerce servers. Int J Network Distrib Comput. 2018;6(2):88-98.

106. Gehrke D, Turban E, Determinants of Successful Website Design: Relative Importance and Recommendations for Effectiveness, Proceedings of the 32nd Hawaii International Conference on System Sciences, IEEE, 1999.

107. Lang J, Howell E. Researching UX: User Research. Victoria, Australia: SitePoint Pty Ltd; 2017.

108. Reynolds M. Beginning E-Commerce with Visual Basic, ASP, SQL Server 7.0 and MTS. Birmingham, England: Wrox Press Ltd; 2000.

109. CData Software Inc., SQL Abstraction for NoSQL Database. Solution Brief, CData Software (www.cdata.com) https://www.cdata.com/resources/sql%20for%20nosql.pdf. Accessed May 19, 2019.

110. Ahmad M, Imran M, Wahab A. Management Issues in Software Development. Recent Advances in Software Engineering, Parallel and Distributed Systems: Proceedings of the 9th WSEAS International Conference on Software Engineering, Parallel And Distributed Systems (SEPADS’10), University of Cambridge, UK, February 20-22, 2010, WSEAS Press, 232-237.

111. Project Management Institute. A Guide to the Project Management Body of Knowledge: (PMBOK® Guide). 6th ed. Newton Square, PA: Project Management Institute; 2017.

112. Cloyd MH. Designing user-centered web applications in web time. IEEE Software. 2001;18(1):62-69.

113. The BasicsofUserExperienceDesign, TheInteractionDesignFoundationhttp://www.interaction-design.org. AccessedJune29, 2017.

114. Budgen D. Software Design. 2nd ed. Harlow, England: Pearson Education Limited; 2003.

115. Foster EC. Software Engineering: A Methodical Approach. New York, NY: Apress Media; 2014.

116. Ola B. A Comparative Evaluation of Specification Techniques for Electronic Commerce Systems [master’s thesis]. University of Manitoba, Department of Computer Science, Winnipeg, Manitoba, Canada; 2002.

117. Ehikioya SA, Ola B. A Comparative Study of Specification Methods for Electronic Commerce Systems. The 3rd ACS/IEEE International Conference on Computer Systems and Applications; 2005:108

118. European Parliament and Council of the European Union. Regulation on the Protection of Natural Persons with regard to the Processing of Personal Data and on the Free Movement of such Data, and repealing Directive 95/46/EC (Data Protection Directive)”—General Data Protection Regulation (GDPR), Official Journal of the European Union, L119; May 4, 2016: 1–88.

How to cite this article: Ehikioya SA, Guillemot E. A critical assessment of the design issues in e-commerce systems development. Engineering Reports. 2020;2:e12155. https://doi.org/10.1002/eng2.12155