Distributed Retail Systems

A. Asani*, T. Dimovski* and I. Hristoski**
* "St. Kliment Ohridski" University – Bitola / Faculty of ICTs – Bitola, N. Macedonia
** "St. Kliment Ohridski" University – Bitola / Faculty of Economics – Prilep, N. Macedonia
arijan.asani.bt@gmail.com, {tome.dimovski, ilija.hristoski}@uklo.edu.mk

Abstract - In the era of modern computing, along with the huge advancements achieved in the field of computer science, most of today's processes are digitalized. There is an increasing demand for reliability, availability, stability, and scalability. When it comes to retail business, retail shop software plays an important role, which points out the need for designing more sophisticated systems. One of the most popular approaches in addressing ever-increasing system requirements is the development of distributed systems. By definition, a distributed system represents a group of computers that process data/tasks together and appears as a single entity to the end-user of the system. In this paper, we are presenting the design and the development of a distributed system that is intended to be used in the retail sector and elaborate on the functionality of each of the individual entities comprising the model. The described distributed retail system represents a solid foundation for further improvements and upgrading.

I. INTRODUCTION

We live in a world where technology is evolving at an accelerated pace and is becoming an inevitable part of our daily lives. Information and communication technologies are increasingly present in almost all spheres of our society. The information systems, and especially distributed ones, are widespread today, playing a key role in the economies and societies across the globe. In the retail sector, the use of modern software solutions is necessary, as there is an increasing need to meet progressively complex customer requirements. Faced with the new challenges, there is an emphasized need for the design, development, and deployment of distributed retail systems.

Distributed systems are systems that are composed of multiple entities, usually spread out across different physical locations, which are mutually synchronized with each other and present to the end-user a picture of being a single entity. The implementation of such types of systems addresses the majority of the customer's needs, such as the need for increased system availability and dependability, better scalability and stability, greater performance improvements, as well as easier data integration and data analysis.

Having minded the previous, the main goal is to present the concepts behind a newly designed and implemented distributed retail system, which has great potentials to grow into a fully functional software solution intended for use by micro- and small-sized retailers.

The rest of the paper is structured as follows. Section II focuses on some of the most relevant research made on this topic recently. In Section III, a brief overview of several well-known retail systems is made. The major capabilities of contemporary retail systems are given in Section IV. Section V discusses the proposed model of a distributed retail system and elaborates on the functionality of each of the individual entities of the model. The last section highlights the conclusions and future work.

II. RELATED RESEARCH

Most of the recent research on this topic has been focused on the exploration of newer technologies that can be leveraged to improve retail business efficiency and performance, as well as to maximize the return on investment (ROI). In this context, some authors elaborate theoretically on the proposed frameworks and consequently describe the implementation of such systems. For instance, Adewumi *et al*. describe the design and the development of *SkyRetail*, a cloud-based retail management system [1]. In his white paper, Dion explores the impact of the deployment of technology in a retail business [2]. More specifically, he looks at the effects on sales, profitability, and productivity of the use of Point-of-Sale, Inventory Control, and Customer Profiling Software in small- to mid-sized retail stores. According to his findings, "the deployment of technology in a retail store leads to higher sales, reduced expenses, and increased gross margins, which have ultimately produced an increase in the overall profitability of those stores who not only have deployed the technology but have also learned how to use it." In his thesis, Cote elaborates on a project that highlights a systematic method for searching, identifying, evaluating, selecting, and recommending a Point-of-Sale (POS) system and Inventory Management (IM) system for a small business, based upon its specific industry needs [3]. According to his findings, information obtained from POS systems improves marketing by helping salespersons make better judgments and ultimately practice smarter selling. Data reports highlight specific needs and eliminate guessing and bias that employees develop throughout their work. Suriyantphupha & Bourlakis conduct further research on the usage and the role of IT in a traditional retail supply chain [4]. The result of the structured literature review they conducted on this topic reveals that the most widespread technologies are radio frequency identification (RFID), point-of-sale (POS), and other inter-organization systems providing visibility in the value chain, reduce labor, lower operation time, and minimize operating costs. However, such innovative technologies are still limited to major retailers due to implementation costs and the compatibility within existing systems and trading partners' systems. Lal *et al*. focus on the study of the effectiveness of POS data in managing the supply chain [5]. By carrying out a comparative analysis of questionaries' data regarding retail stores in a region in India, they conclude that the most notable benefits of adopting POS systems in the retail sector are improved inventory management, increased flexibility of response to customer demands, and reduction in costs and times. Hu *et al*. leverage two new technologies, blockchain, and edge computing, to propose, design, and develop a novel robust retail POS system, which can be used in 'weak' Internet environments [6]. In his technical report, Khaneja describes an efficient Point-of-Sales system that can generate and maintain transaction receipts, inventory reports, and sales records in

the big retail businesses, using UML diagrams and the COMET methodology, consisted of three steps for software development, requirement modeling, analysis modeling, and design modeling [7]. Plomp *et al*. investigate the extent to which Point-of-Sale (POS) systems support retail chain digitization, i.e. inter-organizational processes as being an exchange of orders-related and sales-related information, and consequently, they develop a two-dimensional maturity model for categorizing existing POS systems by their level of backward and forward chain digitization support [8]. Taylor elaborates on mobile payment technologies in the retail sector and reviews potential benefits and risks [9]. By linking research from diverse fields, her paper aims to elucidate the potential impacts of mobile technologies on retail theft and internal technological and process issues, before offering possible solutions.

III. A Brief Overview of Some Existing Retail Systems

In this section, an analysis of the general characteristics of some of the most advanced retail systems available nowadays is conducted. All of these solutions, which are used by millions of users and businesses worldwide every day, are built upon and run distributed systems.

Shopify [10], is one of the most widely used retail systems, developed by Tobias Lütke in Ottawa, Ontario, Canada, in 2004. Over time, the software was continually updated and upgraded, so it nowadays includes more advanced features tailored to the needs of software users. Shopify introduced a handful of new functionalities, such as payments without the need for third-party authentication providers. It is available for several platforms including the most popular ones, such as Android and iOS. Different types of technologies have been used in the development of Shopify, primarily the Ruby on Rails framework. This framework is based on the MVC (Model View Controller) approach to application development, where data transfer uses standards such as JSON and XML. Frontend technologies such as HTML, CSS, and JavaScript have also been used. Shopify uses MySQL as a relational database management system. After several years, the development team started using the PHP language as the primary backend language that has provided a great integration with MySQL databases. The mobile platform applications are developed in Swift/Java.

LightSpeed [11] is an all-in-one cloud-based POS solution, entirely based on the principles of distributed systems, founded by Dax DaSilva in 2005, with its headquarters in Montreal, Quebec, Canada. The technical aspects of the development are similar to those of other retail POS systems. The development tools that are used for the development of this software are the following ones: the TypeScript programming language is used for the frontend. It is a syntactic superset of JavaScript, so the code written in this programming language is also valid as a JavaScript file, i.e. the same files written in TypeScript and JavaScript are mutually compatible; the PHP programming language is used as a backend programming language; the database server runs MySQL RDBMS; Python programming language is used to analyze business data. As with most modern distributed solutions, Amazon Web Services (AWS) is being used as a cloud providing platform.

ShopKeep [12] is one of the most widely used POS systems, especially by small-sized businesses (e.g. retail shops, coffee shops, restaurants, and bars) throughout the United States and Canada. It was developed by Bill Walton in 2008. What is specific about this software is that it is intended for mobile clients, with a special emphasis on tablets (iPads and Android tablets). Distributed software consists of several entities that are synchronized with each other, so they give the end-user a picture as if it were a single whole. The central software is placed on the cloud platform, while the sales part (i.e. the terminal) is the tablet device itself. ShopKeep provides a real-time overview of the entire operation and if the user owns multiple outlets, the data will be grouped so that he can access them at any time. The system allows merchants to ring up sales, print or email receipts, pop a cash drawer, accept credit cards, and print remotely to the kitchen right from an iPad or Android tablet. The web-based BackOffice allows inventory, employee, and customer management, as well as advanced analytics and reporting. The smartphone dashboard app allows merchants to view real-time store sales remotely. SQLite databases are used and installed on tablets, while data on the main server is stored in the PostgreSQL database. The central software and database are set up on the Amazon Web Services (AWS). The MVC framework is Ruby on Rails.

Microsoft Dynamics Retail Management System (RMS) and POS [13] is an application developed by Microsoft that offers small- and mid-size retailers a complete Point-of-Sale (POS) solution that can be adapted to meet unique requirements. It provides centralized control for multi-store retailers and integrates with Microsoft Office applications. It also offers benefits in ease of use, automation, efficiency, flexible reporting, and scalability. It can be deployed for any form of a retail enterprise, from pharmaceuticals to grocery stores. The system, however, is not cloud-based and also targets only businesses that run the Windows operating system. Therefore, it is not a cross-platform solution.

Epicor Cloud Retail Software [14] is a SaaS-based retail solution, developed by Epicor Solutions. It serves small- to mid-sized retailers who want to leverage efficiently their insufficient IT resources. Epicor delivers a model that significantly reduces capital investments, implementation challenges, and ongoing requirements of managing IT. With Epicor, retailers can integrate their sales channels, order management, POS systems, inventory, and other operations to access the right information at the appropriate time. Epicor cloud retail software supports merchandising, store operations, CRM, audit and operations management, and planning. The system, however, is difficult to customize to suit well a particular organization.

IV. Capabilities of Modern Retail Systems

The major functions each retail POS systems is expected to support are the following ones:

- *Transaction execution*: Today, almost every retail POS system has an integrated scanner that automates the ringing up of sales by reading the bar code on each item a customer is purchasing. The Stock Keeping Unit (SKU) number and price are then entered directly into the POS software, which keeps a running total of the order and calculates the final amount due. Some retail POS systems include

a touchscreen component that is used, in conjunction with or in stores with very limited inventory, as an alternative to scanning. Instead of scanning items (or manually entering prices, as with an electronic cash register), store associates tap a corresponding icon on the touchscreen to 'build' the transaction. Unlike electronic cash registers, many retail POS systems also feature integrated payment processing capabilities. With these systems, retailers need not use a separate credit card reader to read the magnetic stripe on credit cards. Rather, this part of the transaction can be handled right on the POS terminal;

- *Inventory control*: In many instances, retail POS systems incorporate software modules that allow retailers to better manage their inventory: each time a customer purchases a particular item, information about it (e.g., its SKU number and the quantity bought) is automatically transmitted from the POS software to the inventory management module. The quantity purchased is then subtracted from the inventory 'tally', without any human intervention. Some retail POS systems generate an alert when inventory levels reach user-defined minimums; some also generate re-orders based on this information;
- *Labor management*: Over the past few years, vendors have introduced configurations in which the POS system features labor management functionality through time and attendance and/or scheduling module. With the former, employees clock in and out right on the POS terminal, making it easy to keep tabs on attendance and limiting employees' ability to punch in and out for each other or work beyond their scheduled hours. An interface with the POS system permits retailers to compare store traffic with sales patterns, and then adjust schedules based on historical information and real needs rather than guesswork;
- *Reporting*: A good POS system gives retailers the option to generate a myriad of basic and advanced reports and, in some cases, transfer them directly to other systems. Standard reports created by the retail POS system break out customer purchase histories and merchandise sold (by SKU), as well as indicate the cost of goods sold, gross sales, low inventory counts, existing inventory counts, customer purchase history, and item-specific sales. Some POS systems allow users to customize their reports and create new ones;
- *Marketing*: Retail POS systems track customers' purchase histories, along with their contact information. With this information in hand, retailers can identify the audience for a variety of marketing campaigns, ranging from the somewhat general (e.g. all female customers) to the very specific (for example, all customers who reside in a specific geographic area). A comprehensive POS package will allow for e-mail marketing, as well as direct mail reports so that marketing e-mails may be composed on and sent directly from the POS system. It is also helpful in face-to-face communication with customers. Most POS systems accommodate the addition of miscellaneous notes about each customer (such as his or her birthday and other specific details), that can be leveraged in marketing campaigns.

V. Distributed Retail System

The main objective of this paper is to present a model of a distributed retail system and the functionality of each of the individual entities of the model [15].

A. Model of a Distributed Retail System

In this section, we are presenting a model of a distributed retail system that we have used during the development and implementation phase. The model, presented in Fig. 1, consists of two types of nodes:

- Client nodes
- Central node

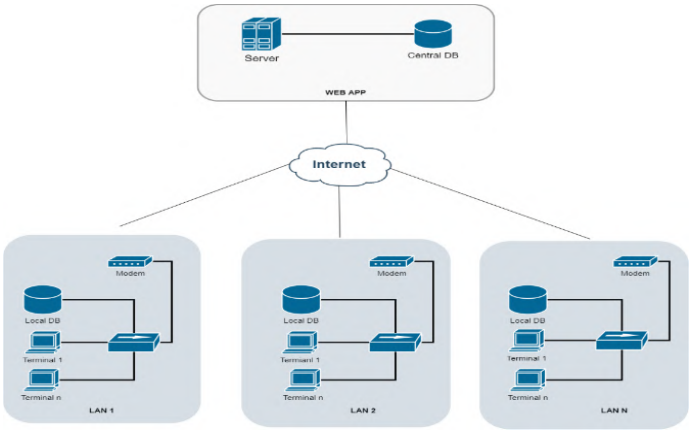

Figure 1. Model of a distributed system

The *client node* is composed of two main components.

The first component is the *terminal*. The terminal can be any kind of a desktop or a laptop computer, on which the *client software* is installed, which is part of the distributed sales system. Each terminal has a local database installed on it, which contains all the data needed for work/sales. The data in the local database is synchronized with the data in the central database, which is located on the central node of the distributed system. Additionally, each terminal may have a cash register and a bar-code reader. The client node can consist of *N* terminals, depending on the number of payment points required.

The second component of the client node is the *computer network*, which enables network communication and data exchange between the terminal(s) and the central node of the distributed retail system.

The *central node* of the distributed retail system consists of an *application server* and a *database management server*. The application server is responsible for the synchronization of the local databases from the client nodes with the central database in the central node, as well as for the functioning

and coordination of the distributed retail system. A central database is set up on the database management server, which synchronizes and stores the database data from the client nodes [16].

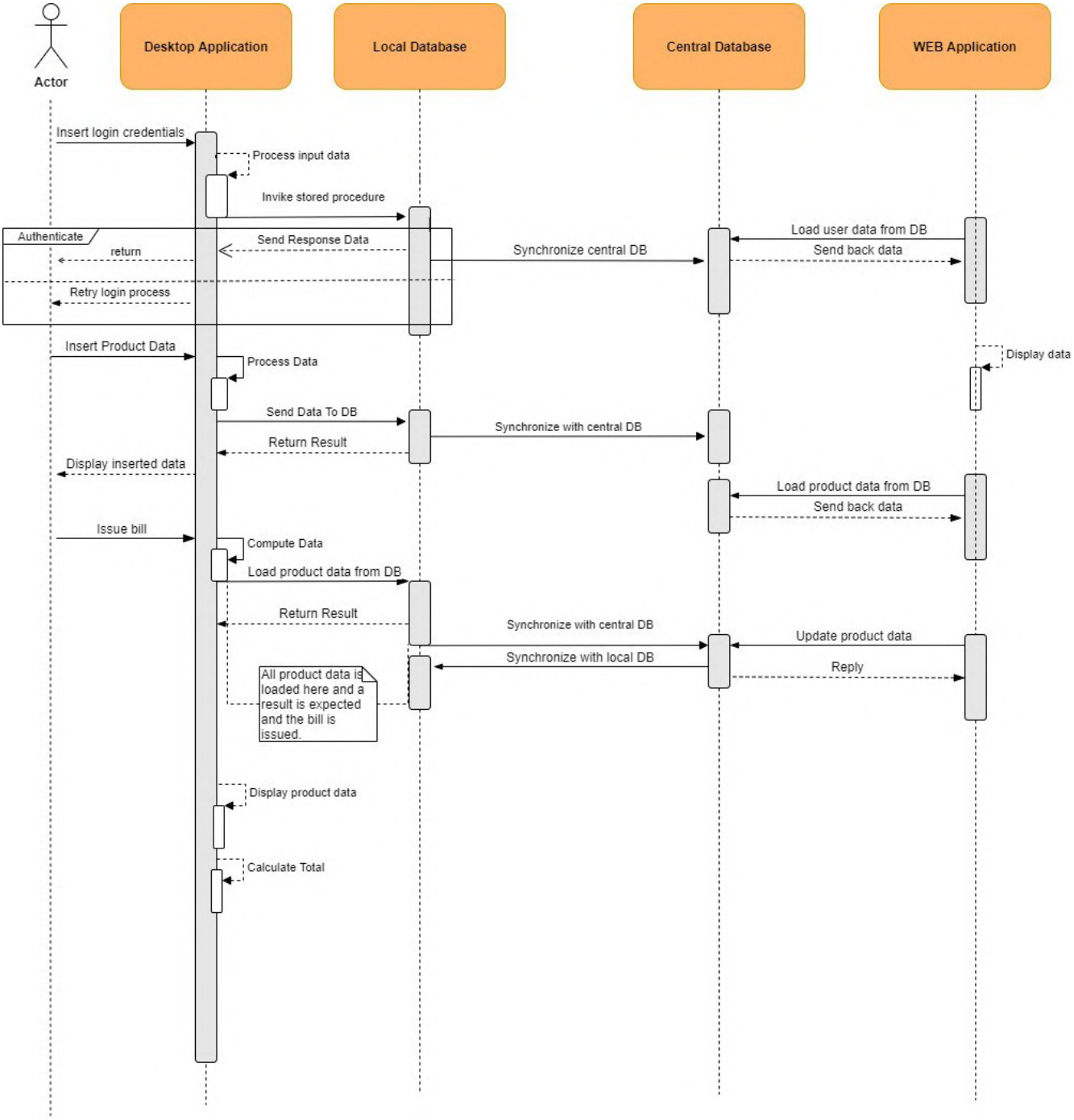

Figure 2. Sequence UML diagram

B. Functionality of the System Entities

This section presents the entities that have been developed and represent an integral part of the distributed retail system. Using the Sequence UML diagram, shown in Fig. 2, the interaction and communication between the individual model entities of the distributed retail system are shown. The distributed retail system entities are interconnected and synchronized. The entities, i.e. the separate parts of the designed solution, can be physically located in different locations, but at the same time, they are functioning as if they represent a single monolithic system.

The distributed retail system consists of the following entities:

- Desktop application
- Web application
- Local database
- Central database

The *desktop application*, schematically depicted in Fig. 3, is the client software that runs on the terminal.

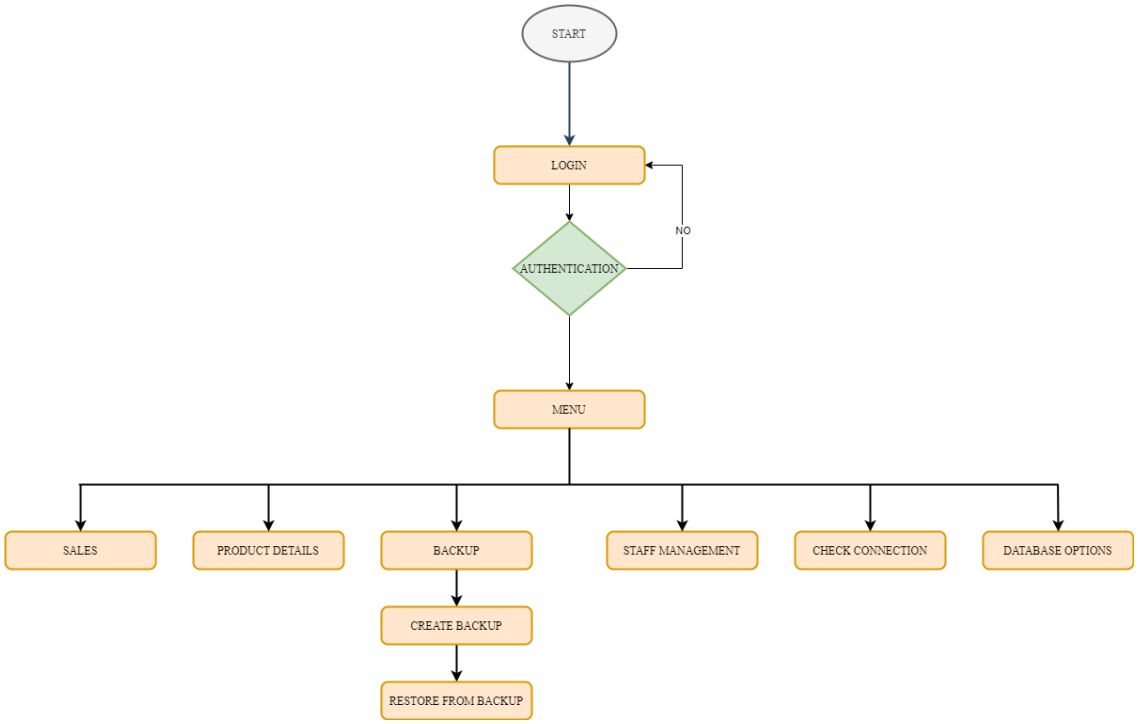

Figure 3. UI flowchart

User Login is a process in which a previously registered user logs into the desktop application. The desktop application performs authentication. If data entered by the user is correct, the working session begins. All processes covered by the login process are synchronized with the central database. In the proposed distributed retail system, two roles are defined, a user and an administrator. The UML Use Case diagram, shown in Fig. 4, provides a clear picture of each of the roles and privileges of each role in the system. The user role has reduced privileges as compared to the administrator role.

Terminal Sales is a process that enables the user to issue all sorts of receipts for items used for sales. Also, this process keeps track of the number of times each item has been sold.

Connection Checker is a module that checks the connection between client nodes and the central node, which is critical to their successful synchronization.

Through the web application, which is a part of the distributed retail system installed on the central node, items used for sale are registered, and the information is distributed to the desktop applications installed on the client nodes. The web application monitors database synchronization, generates product sales reports, registers users in the distributed system, and configures retail system parameters.

Central and local databases are one of the most important components because they contain the data and reflect the business model. Each terminal has a local database installed on it. The central database is located on the central node of the distributed retail system. By entering or changing any data in any local database, synchronization with the central database is done automatically by database replication. In our distributed retail system, we are using asynchronous replication, i.e. the local database copies the data to the central database after the data is already written to the local database. The replication process occurs following a previously defined schedule.

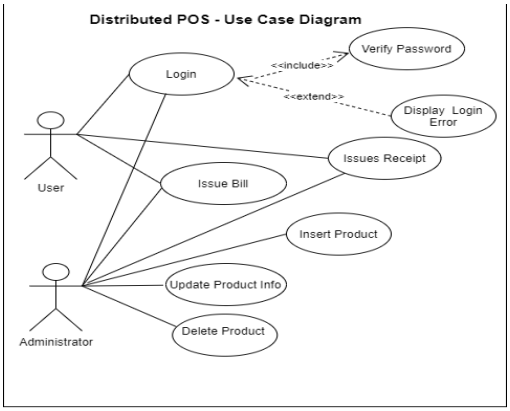

Figure 4. Use Case Diagram

C. Development Technologies

This section briefly lists the selected technologies used in the development of the developed distributed retail system.

Frontend technologies such as HTML, CSS, and JavaScript were used to create the web application. We used the Bootstrap development framework which allows creating interfaces that are compatible with all devices. Both .NET and C# programming language were used to create the backend part of the web application, as well as to create the desktop application. MySQL server was used as a relational database management system for running central and local databases [17–21].

VI. Conclusion

Retail shop systems dramatically changed over the past decade. Today's modern solutions provide users with the ability to control their business from literally any point, using even their mobile devices. The contemporary cloud-

based and distributed solutions deliver efficiency and, at the same time, save money and time. Distributed retail systems can operate and be efficient even if certain parts of the entire system fail. The future work vis-à-vis the hereby described software framework includes addressing its scalability and portability issues in terms of their improvement, as well as the deployment of the whole system to a cloud-based service. One of the projected directions for further development of the proposed solution is also its adaptation for mobile platforms. All aspects of the proposed retail management software framework need to be tested and validated in the real operating surrounding, so the usability study of the system will also be conducted as a part of the future work. As a way of improving the core features of the proposed software framework, integrating third-party tools, such as a payment gateway to process card payments, and implementing SMS and/or e-mail services to deliver instant messages or reports to the users, will be considered, as well.

REFERENCES

[1] A. Adewumi, S. Ogbuchi, and S. Misra, "A Cloud-Based Retail Management System," in R. Intan, C. H. Chi, H. Palit, L. Santoso, Eds. Intelligence in the Era of Big Data. ICSIIT 2015. Communications in Computer and Information Science (CCIS), vol 516, pp. 446–456, Springer, Berlin, Heidelberg, March 2015.

[2] J. E. Dion, "The Effects of POS Implementation and Retail Technology on Sales and Profitability for Small to Mid Sized Retailers," Fargo, ND, USA, Dionco Inc., White Paper, April 2003. [Online]. Available: http://www.dionco.com/public/articles/POS_Study_White_Paper.pdf

[3] M. Cote, "The Power of Point of Sale: Improving Growth, Profit, and Customer Service in a Retail Business," B.S. thesis, Bryant University, Smithfield, RI, USA, April 2015. [Online]. Available: https://digitalcommons.bryant.edu/cgi/viewcontent.cgi?article=1016&context=honors_marketing

[4] P. Suriyantphupha and M. Bourlakis, "Information Technology in a Traditional Retail Supply Chain: A Structured Literature Review," Projectics / Proyéctica / Projectique, vol. 22, no. 1, pp. 89–102, 2019.

[5] M. K. Lal, A. C. Shukla, and B. E. Tarangini, "Study of Effectiveness of POS Data in Managing Supply Chain," Industrial Engineering Journal, vol. 11, issue 10, pp. 1–9, October 2018.

[6] B. Hu, H. Xie, Y. Ma, J. Wang, and L.-J. Zhang, "A Robust Retail POS System Based on Blockchain and Edge Computing," EDGE 2018, LNCS 10973, pp. 99–110, 2018.

[7] D. Khaneja, "Point of Sales Terminal (POST) Using UML," Project Report, Carleton University, Ottawa, ON, Canada, December 2017. [Online]. Available: https://www.researchgate.net/publication/323073643_POINT_OF_SALES_TERMINAL_POST_using_UML

[8] M. G. A. Plomp, G. van Rijn, and R. S. Batenburg, "Chain digitisation support by point-of-sale systems: an analysis of the Dutch product software market," International Journal of Information Technology and Management, vol. 11, no. 4, pp. 257–272, 2012.

[9] E. Taylor, "Mobile Payment Technologies in Retail; A Review of Potential Benefits and Risks," International Journal of Retail and Distribution Management, vol. 44, no. 2, pp. 159–177, 2016.

[10] K. Shatrov, "E-Commerce at Scale: Inside Shopify's Tech Stack - Stackshare.io", Shopify Engineering, August 2018. [Online]. Available: https://engineering.shopify.com/blogs/engineering/e-commerce-at-scale-inside-shopifys-tech-stack

[11] LightSpeed, "Reinvent your business with the omnichannel retail POS system," 2020. [Online]. Available: https://www.lightspeedhq.com/pos/retail/

[12] ShopKeep, "Do More With ShopKeep's Retail POS System," 2020. [Online]. Available: https://www.shopkeep.com/business-types/retail-pos

[13] Microsoft, "Microsoft Dynamics RMS and POS," 2020. [Online]. Available: https://mbs.microsoft.com/customersource/Global/RMSPOS

[14] Epicor, "Epicor Retail Cloud: Retail Growth Simplified," 2020. [Online]. Available: https://www.epicor.com/en-us/retail-software/epicor-retail-cloud/

[15] A. S. Tanenbaum and M. van Steen, *Distributed Systems: Principles and Paradigms*, 2nd ed. London, England: Pearson, 2006.

[16] H. Garcia-Molina, J. D. Ullman, J. Widom, *Database Systems: The Complete Book*, 2nd ed. London, England: Pearson, 2008.

[17] J. P. McManus and C. Kinsman, *C# Developer's Guide to ASP.NET, XML, and ADO.NET*. Boston, MA, USA: Addison-Wesley Professional, 2002.

[18] D. Avison and G. Fitzgerald, G., *Information systems development: Methodologies, Techniques and Tools*, 4th ed. New York, NY, USA: McGraw-Hill Education, 2006.

[19] T. Barnaby, *Distributed .NET Programming in C#*, New York, NY, USA: APress Media, LLC, 2002.

[20] J. Bender and J. McWherter, *Professional Test Driven Development with C#: Developing Real World Applications with TDD*, 1st ed. Indianapolis, IN, USA: Wrox/Wiley Publishing, Inc., 2011.

[21] B. Schwartz, P. Zaitsev, and V. Tkachenko, *High Performance MySQL: Optimization, Backups, and Replication*, 3rd ed. Sebastopol, CA, USA: O'Reilly Media, 2012