Journal of Innovative Technology Convergence
Vol. 1, No. 1 June 2019, pp. 39-50

QuickyPOS Software: A Systematic Point-of-Sale Development Approach

Benjie J. Sullano [1], Ronnie D. Caytiles [2], Jacinto P. Valera [3], Jason P. Sermeno [4*]

Abstract: This paper deals with a systematic approach to the development of point-of-sale (POS) software referred to as QuickyPOS. It aims to alleviate day-to-day business transactions by providing tons of benefits. Transaction processes are automated to simplify operations, quicken payments, provide better management for customers, orders, purchasing, supplies, and inventory. The systematic approach for the QuickyPOS development is based on the five basic phases of systems development which are the requirements, analysis, design, implementation, maintenance phases. Business diagrams will be presented to show the systematic design of the features of the POS system.

Keywords: Point-of-sale (POS), QuickyPOS, Software development, Systematic approach, MVC framework

1. Introduction

A key component of running a successful business is the use of point-of-sale (POS) software [1][2]. In earlier days, point-of-sale was just a cash register [3]. Because of progressive technology, POS software has made significant advancements [4]. This has allowed QuickyPOS software to give business owners a convenient way of checking out customers and of recording sales. It keeps a record of the store inventory, updating it when an order is processed. It also prints out receipts and reports.

QuickyPOS software makes business accounting a lot easier by creating reports on inventory, sales, etc. Since it is already recording each sale, it can easily tell the sales and revenue of the day. It could help speed up the business transaction process while providing detailed and accurate receipts thereby improving customer satisfaction. QuickyPOS software increases customer sales, improves customer satisfaction, and makes a business run smoother and more efficiently.

[1] College of Computer Studies, University of Antique, Sibalom, Antique, Philippines
Email: sullanobenjie@yahoo.com
[2] Multimedia Engineering Department, Hannam University, Daejeon, South Korea
Email: rdcaytiles@gmail.com
[3] College of Computer Studies, University of Antique, Sibalom, Antique, Philippines
Email: jhing_vp@yahoo.com
[4*] College of Computer Studies, University of Antique, Sibalom, Antique, Philippines
Email: jason.sermeno@antiquespride.edu.ph (Corresponding Author)

Received [November 11, 2018]; Revised [January 20, 2019]; Accepted [February 5, 2019]

Published by InnoCon Publishing.
ISSN 2704-4440

QuickyPOS software is generally easy to install and easy to use. It includes features to update inventory and record a price change for an item. It usually provides an easy-to-use interface to do this. It can make the job of the cashier a lot easier by automating the routine tasks of the day.

QuickyPOS software may be the right solution for sales businesses and can provide tons of benefits. Keeping customers and employees happy is an essential part of running a successful business. The use of QuickyPOS software will help create happy customers, stress-free employees, and easier business management. It will result in a much more profitable business.

This paper deals with the systematic approach to the development of QuickyPOS software to enable convenient transactions for the payment of products or services. The development is based on the five basic phases of systems analysis and design to guarantee the quality of service (QoS) for the features and functionalities of the developed system. In addition, the Use Case modeling approach and model-view-controller (MVC) framework were utilized in modeling the systems architecture where the identified system requirements have been broken down into components known as modules.

The rest of this paper is organized as follows: Section 2 discusses the analysis of the different requirements for QuickyPOS software; Section 3 outlines the QuickyPOS software components using the MVC framework; the various QuickyPOS subsystems and the entity relationships were identified in Section 4; the graphical user interface (GUI) design is illustrated in Section 5; and the concluding remarks is presented in Section 6.

2. Requirements Analysis

This section provides a discussion on the requirements analysis phase of the Software Engineering (SE) based point-of-sale (POS) system [5][6]. The different user requirements and modules for the development of QuickyPOS software are outlined and described.

2.1 User Requirements

Based on the problem statement, the following requirements are identified of “what QuickyPOS software must do”:

- *Create and modify user accounts and account types*. The system should be able to manage user accounts and modify each account type.
- *Manage product items* which include item brands, item categories, and its suppliers. The system should be able to manage item brands, item categories, and suppliers of each item brand.
- *Manage order transactions*. The system should be able to create an order list, complete order transactions, and modify orders after completion of the transaction.
- *Manage sales transactions* including discounts on privileged customers and generate daily, monthly, and yearly sales reports. The system should be able to process purchases accordingly.
- *Generate reports*. The system should be able to generate reports; all brands reports, all categories reports, all employees reports, all items reports, brand reports, category reports, customer reports, daily reports, data range reports, employee reports, item reports, profit reports, tax reports, and charts (weekly, monthly, and yearly).
- *Modify delivery and update inventory*. The system should be able to modify delivered items and update inventory.

- *Manage system configuration*. The system should be able to modify the company profiles for System GUI, reports, and receipts.

These requirements will then be broken down into components known as modules. Each module will define the functionality of QuickyPOS software.

2.2 Modules

Based on the identified user requirements, QuickyPOS software's functionality is then broken down into components known as modules. Each module defines each functionality of the QuickyPOS software. The different modules identified are as follows:

- *User Accounts*. This module manages user account information.
- *Account Types*. Used to create new account types, modify existing account types, and modify account privileges.
- *Suppliers*. This module manages product suppliers.
- *Brands*. This module manages item brands.
- *Categories*. This module manages item categories.
- *Items*. This module manages items.
- *Reorder Transactions*. Used to create order lists, complete order transactions, and modify orders after completion.
- *Sales Transactions*. This module starts a new sale, apply extra discounts at the time of sales, completes sales transactions, handles cash transactions, abandons a sale, and modifies a sale after completion.
- *Reports*. This module generates reports; daily sales, monthly sales, yearly sales, profit, and custom range reports.
- *Discounts*. This module manages discounts on items.
- *Delivery*. This module is used to modify delivered items and update inventory.
- *Cash Count*. A module that generates a report on cash count after user shift.
- *System Configuration*. This module is used to modify the company profiles for System GUI, reports, and receipts.

2.3 Use Case Diagram and Scenarios

Use case diagrams are a type of behavioral diagram defined by and created from a Use Case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (*i.e.*, represented as use cases), and any dependencies between those use cases [7][8].

The main purpose of a use case diagram is to show what system functions are performed for which actors. Roles of the actors in the system can be depicted. Figure 1 shows the system functions of QuickyPOS software which are best discussed in the scenarios identified in Table 1 which depicts the functionality of the different systems in interaction with the users of QuickyPOS software.

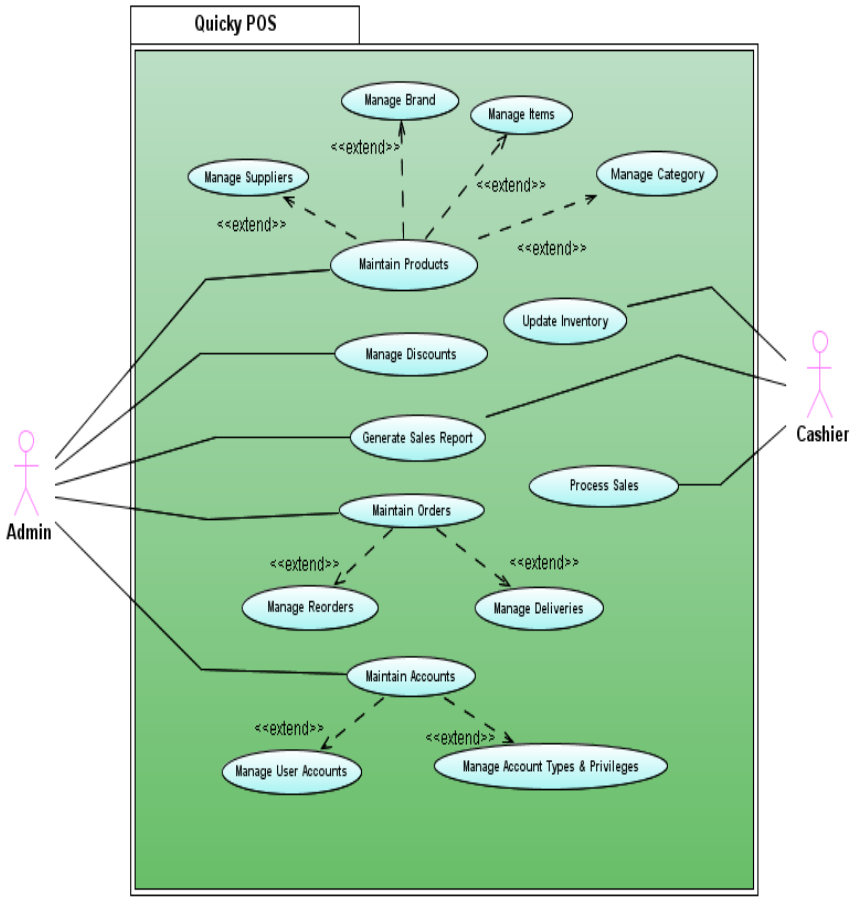

Figure 1. Use CASE Diagram Showing the System Functions Performed by the Users of QuickyPOS

Table 1. Use CASE Scenarios Showing the Realistic Transactions Made with QuickyPOS Software

Scenarios	**Description**
Maintain Products	In this aspect of the system, only the Administrator has the privilege to access this module.
Manage Items	• New products can be added to the existing database • You can select a brand, category, and supplier or a specific product • Details of the existing products can be modified and changed • Existing items can also be deleted and removed from the database • Update the product
Manage Brands	• Brand for certain products can be added, modified, and omitted
Manage Category	• Category for certain products can be added, modified, and omitted

Manage Suppliers	• Supplier for certain products can be added, modified, and omitted • Modify or changed details of suppliers
Maintain Orders	Only the Administrator has the privilege to access this module.
Manage Reorders	• You can produce a list of products that need to be reordered • You can customize the selection of products to be included in the reorder list • Generate a new order list

3. MVC Framework for QuickyPOS

Model–View–Controller (MVC) is an architectural pattern used in software engineering [9][10]. Successful use of the pattern isolates business logic from user interface considerations, resulting in an application where it is easier to modify either the visual appearance of the application or the underlying business rules without affecting the other [11].

In MVC, the Model represents the information (the data) of the application; the View corresponds to elements of the user interface such as text, checkbox items, and so forth; and the Controller manages the communication of data and the business rules used to manipulate the data to and from the model [12][13].

3.1 Model

Model is the domain-specific representation of the information on which the QuickyPOS operates. The project as a whole was broken down into components called modules. These modules based on the user requirements specified are shown. Model is depicted in Figure 2.

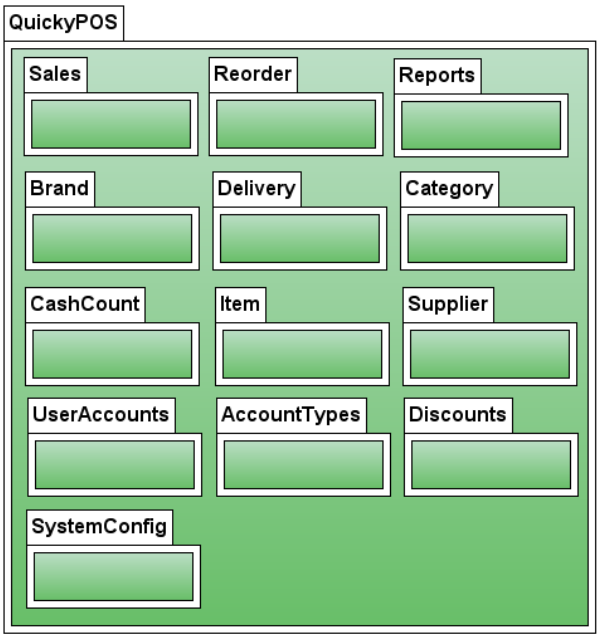

Figure 2. Model shows the different subsystem components of the Quicky POS software

3.2 View

View renders the model into a form suitable for interaction, typically a user interface element. Every model has its own view as indicated in Figure 3.

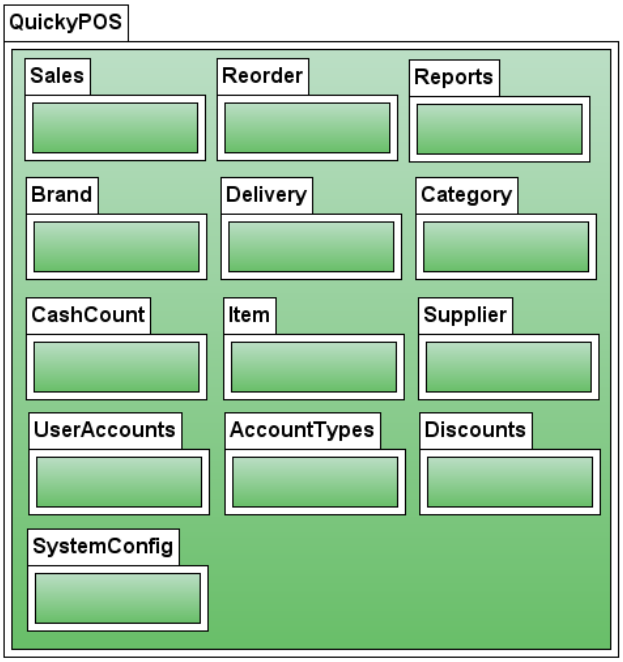

Figure 3. View Components of Quicky POS software

3.3 Controller

Controller represents processes and responds to events (typically user actions) and may indirectly invoke changes on the model.

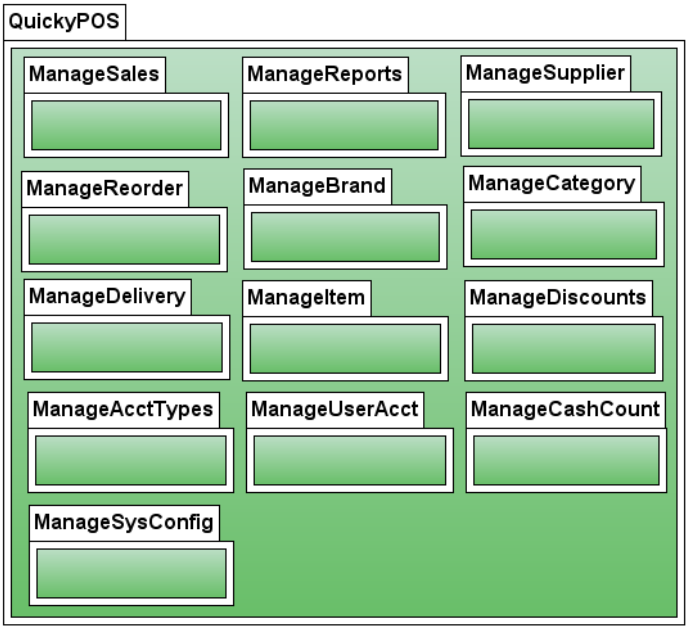

Figure 4. Controller shows the actions done for Model subsystems

4. System Components of QuickyPOS Software

This section provides a discussion on the system components of QuickyPOS software. Its subsystems have been identified as the basis for the creation of its subsystem interfaces that can illustrate the different interactions between the QuickyPOS subsystems. The entity relationship diagram is also presented to outline the relationships between the different entities.

4.1 QuickyPOS Subsystems

The subsystems identified on the MVC framework are as follows:

1. *User Accounts*. This subsystem is used to manage user account information.
2. *Account Types*. This subsystem is used to create new account types, modify existing account types, and modify account privileges.
3. *Suppliers*. This subsystem is used to manage product suppliers.
4. *Brands*. This subsystem is used to manage item brands.
5. *Categories*. This subsystem is used to manage item categories.
6. *Items*. This subsystem is used to manage items.
7. *Reorder Transactions*. This subsystem is used to create an order list, complete order transactions, and modify orders after completion.
8. *Sales Transactions*. This subsystem starts a new sale, apply extra discounts at the time of sales, completes sales transactions, handles cash transactions, abandons a sale, and modifies a sale after completion.
9. *Reports*. This subsystem generates reports on daily sales, monthly sales, yearly sales, profit, and custom range reports.
10. *Discounts*. This subsystem is used to manage discounts on items.
11. *Delivery*. This subsystem is used to manage delivered items and update inventory.
12. *Cash Count*. This subsystem is used to generate a report on cash count after the user shift.
13. *System Configuration*. This subsystem is used to manage company profiles for System GUI, reports, and receipts.

4.2 Subsystem Interfaces

Every subsystem is associated with each other. The interfaces that illustrate the interactions between QuickyPOS software subsystems are as follows:

1. *Item/Brand/Category/Supplier/Reorder/Reports*. The user is the one who manages purchases of items where the items are classified according to brand, category, and supplier. From these processes, orders and reports are derived.
2. *Sales Transactions/Item/Reports*. In every sales transaction, purchases are derived from items. Sales reports are generated daily, monthly, and yearly.
3. *Item/Reorder Transactions*. The quantity of items in inventory is deducted in accordance with every purchase. A reorder transaction is made when the quantity of items reached the order level.
4. *Reorder/Delivery/Item/Brand/Category/Supplier*. In every reorder transaction, the delivery of ordered items is classified according to brand, category, and supplier.

5. *Cash Count/Report*. Cash count report is generated every end of user shift.
6. *Account Type/System Configuration/User Accounts/Reports*. Privileged users can manage user accounts, company profiles, reports, and receipts.
7. *Discounts/Sales Transactions/Items*. The user is the one who manages discounted items in every purchase.
8. *Item/Brand/Category/Supplier*. Inventory is classified according to brand, category, and supplier.

4.3 Database Relational Diagram

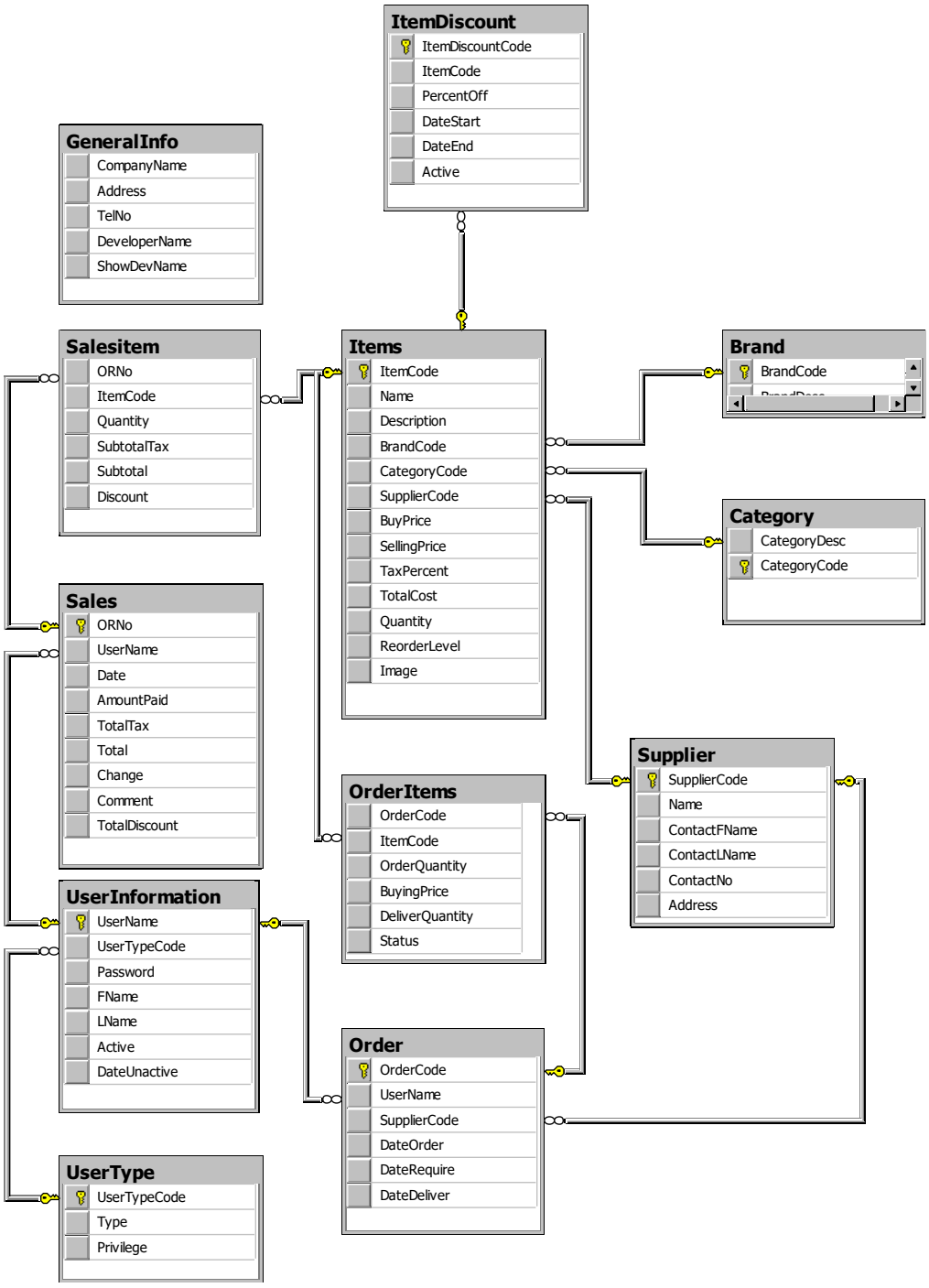

Figure 5. Entity Relationship Diagram (ERD) for QuickyPOS software

The entity relationship (ER) diagram of QuickyPOS software depicted in Figure 5 illustrates the interrelationships between entities in a database. QuickyPOS software ER diagrams use symbols to represent three different types of information. Boxes are commonly used to represent entities. Diamonds are normally used to represent relationships and ovals are used to represent attributes. Relationships between entities are indicated by their connections.

5. System Components of QuickyPOS Software

The QuickyPOS software is characterized to be a user-friendly and up-to-date point-of-sale system. Figure 6 depicts a screenshot of the QuickyPOS main screen wherein the descriptions are described in Table 2.

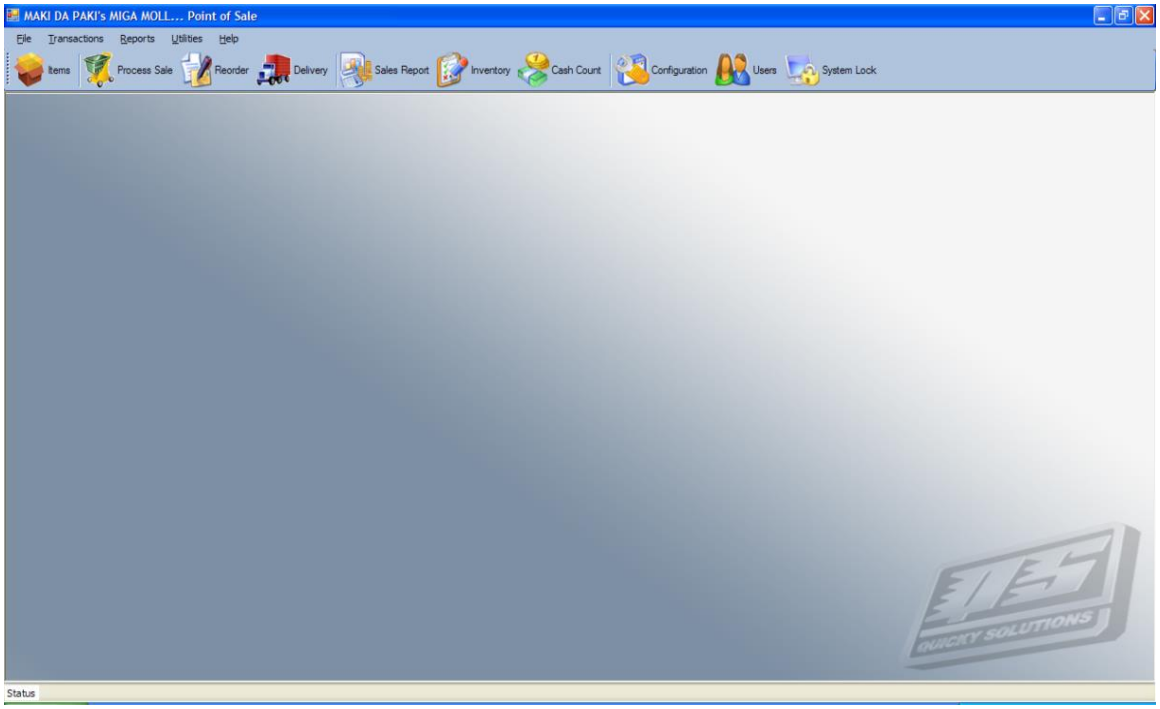

Figure 6. QuickyPOS software Main Screen

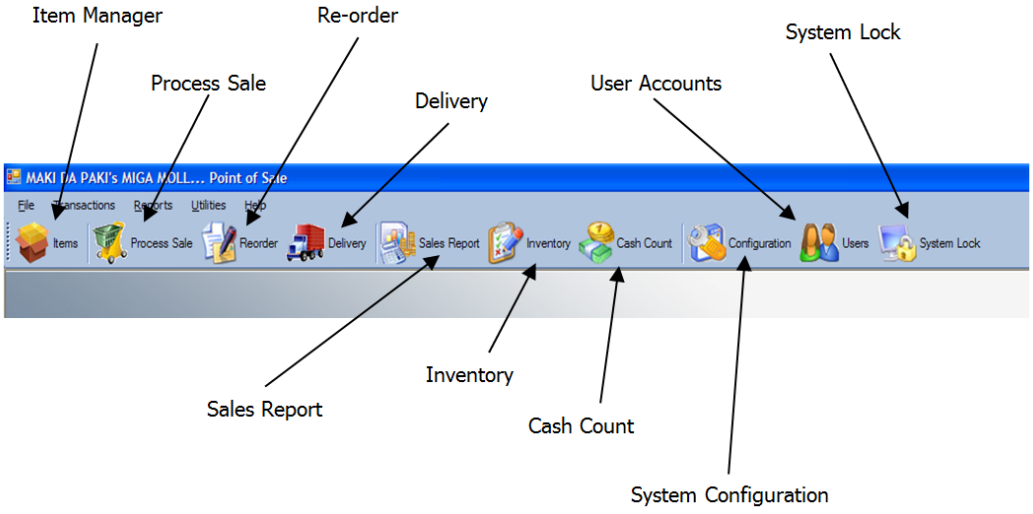

Figure 7. QuickyPOS software Toolbar

Table 2. The QuickyPOS software main screen menu descriptions

Menu	Name	Description
File	Manage Items	Contains submenu to manage items, brand, category, supplier, and discounts.
	Items	A feature for managing item descriptions.
	Brand	A feature for defining the product's Brand, Category, & Supplier Information.
	Category	
	Supplier	
	Discount Manager	Manages the discount for every defined product.
	Log-off <username>	Logs off a current user.
	Exit	Terminates the application.
Transactions	Process Sale	Processes a sale transaction.
	Reorder	Processes and generates a re-order checklist.
	Delivery	Processes a delivery from a re-order checklist.
Reports	Sales	Daily, weekly, and monthly Sales report and Profit report.
	Re-order checklist	Checklist of re-order transactions.
	Inventory	Inventory of products.
	Cash Count	At the end of a shift, a cashier/user is required to submit a printout of all sale transactions he/she have done for the day.
Utilities	System Configuration	Defines the system profiles such as the company name and address of the company.
	User Type	Defines the account type of a user. A utility for defining the name of an account type.
	User Account	Defines the user of the system.
	System Lock	A feature for locking the system GUI.
Help	Contents	A site for technical assistance.
	About	Contains the system information.

The menu items of the QuickyPOS software main screen are defined in Table 2 which include File, Transactions, Reports, Utilities, and Help. That is, menu names are indicated in the first column, the commands within the specific menu in the second column, and their descriptions in the third column.

The different tools (*i.e.*, shortcut icons) that can be used on the QuickyPOS software toolbar are depicted in Figure 7. These shortcut icons can make the operations faster and easier to use, thus, increasing the organization's productivity. These include Item Manager, Process Sale, Re-order, Delivery, User Accounts, System Lock, Sales Report, Inventory, Cash Count, and System Configuration.

6. Conclusion

This paper outlines a systematic approach to the development of QuickyPOS software for business accounting systems. It has comprehensively analyzed the user requirements which are the most essential phase for any software development. The QuickyPOS software includes features such as keeping track of store inventory, printing out receipts and reports, and managing day-to-day business transaction processes. It can speed up business transaction processes that significantly improve customer satisfaction.

In the future, a thorough evaluation of the QuickyPOS software through test cases or scenarios will be provided. The integration of new technologies such as Quick Response (QR) codes, Bluetooth, Radio Frequency Identification (RFID), *etc.* In addition, integration of QuickyPOS software into networked systems will be studied.

References

[1] D. S. Laar, J. K. Konjaang, B. A. Tankia, "*Design and Development of a Sales Management System for SMEs in Northern Ghana*", International Journal of Innovative Research in Advanced Engineering (IJIRAE), vol. 2, no. 5, May 2015, pp. 66-77.

[2] Y. G. Kim and J. Lim, "*A POS system based on the remote client-server model in the small business environment*", Management Research Review, vol. 34, no. 12, November 2011, pp. 1334-1350, doi: 10.1108/01409171111186432.

[3] M. Hudson, "*Cash Registers vs. Point of Sale (POS) Systems*", www.thebalancesmb.com/cash-registers-vs-point-of-sale-pos-systems-2890121 (Accessed October 8, 2018).

[4] A. Hayes, "*Point of Sale (POS)*", www.investopedia.com/terms/p/point-of-sale.asp (Accessed November 2, 2018).

[5] J. T. Catanio, "*Requirements Analysis: A Review*", in Advances in Systems, Computing Sciences and Software Engineering, T. Sobh, K. Elleithy, Eds., Netherlands: Springer, Dordrecht, 2006, pp. 411–418, ISBN: 978-1-4020-5263-7, doi: 10.1007/1-4020-5263-4_64.

[6] ReQtest, "*Requirements Analysis – Understanding the Process & Techniques*", https://reqtest.com/requirements-blog/requirements-analysis/ (Accessed November 2, 2018).

[7] R. Klimek and P. Szwed, "*Formal Analysis of Use Case Diagrams*", Computer Science, vol. 11, January 2010, pp. 115-131, doi: 10.7494/csci.2010.11.0.115.

[8] A. Gemino and D. Parker, "*Use Case Diagrams in Support of Use Case Modeling: Deriving Understanding from the Picture*", Journal of Database Management, vol. 20, no. 1, January 2009, doi: 10.4018/jdm.2009010101.

[9] A. Majeed and I. Rauf, "*MVC Architecture: A Detailed Insight to the Modern Web Applications Development*", Peer Review Journal of Solar & Photoenergy Systems, vol. 1, no. 1, September 2018, pp. 1-7.

[10] R. F. Grove and E. Ozkan, "*The MVC-Web Design Pattern*", In Proceedings of the 7th International Conference on Web Information Systems and Technologies, 2011, pp. 127-130, doi: 10.5220/0003296901270130.

[11] D. P. Pop and A. Altar, "*Designing an MVC Model for Rapid Web Application Development*", Procedia Engineering, vol. 69, 2014, pp. 1172-1179, doi: 10.1016/j.proeng.2014.03.106.

[12] M. Badurowicz, "*MVC architectural pattern in mobile web applications*", Actual Problems of Economics, vol. 6, no. 120, January 2011, pp. 305-309.

[13] Brainvire, "*Six Benefits of Using MVC Model For Effective Web Application Development*", www.brainvire.com/six-benefits-of-using-mvc-model-for-effective-web-application-development/ (Accessed November 2, 2018).