Uduak Asuquo

Mobile Self-Checkout Application

Metropolia University of Applied Sciences

Bachelor of Engineering

Information and Communications Technology

Thesis

15 November 2017

Author Title Number of Pages Date	Uduak Asuquo Mobile Self-Checkout Application 34 pages 15 November 2017
Degree	Bachelor of Engineering
Degree Programme	Information Technology
Professional Major	Software Engineering
Instructors	Kimmo Sauren, Thesis Supervisor Sonja Holappa, Language Instructor

This thesis proposes the development of a mobile self-checkout application for a client's store as a supplementary payment system to the existing traditional customer payment procedure via the cashier checkout point. The application to be created would be mobile compliant and available across all platforms i.e. IOS, Android etc. With this application, customers would be able to take up the task of a cashier by scanning products they intend buy and complete payment themselves. This would boost the payment process and management system of the client's franchise.

The purpose of the thesis was to develop a functional model that fine points the key actors of the self-checkout application i.e. virtual shopping cart, store, employee and the customer by taking a comprehensive approach in illustrating the various process flows via UML diagrams emanating from each actor's interplay with the application towards accomplishing a set of objectives. Moreover, the purpose was to elaborate on the functional features of the application by delving into the technical facet of the development of the application, i.e. frontend and backend technologies used.

In this study prototypes of the UI were also designed to showcase the possible features of the mobile self-checkout application. A recommendation for the development and implementation of the application was also made.

Keywords	Mobile self-checkout application, Virtual shopping cart, Store, Employee, Customer, UML diagram, Product, Prototype

Contents

‘

List of Abbreviations

IOS	iPhone Operating system.
SACAT	Semi Attended Customer Activated Terminal.
UML	Unified Modeling Language.
UI	User Interface.
CVC	Card Verification Code.
CVV	Card Verification value.
CID	Card Identification.
MS SQL	Microsoft Structured Query Language.
MySQL	“My” Structured Query Language.
HTML	Hypertext Markup Language.
PHP	Personal Home Page (former name). Now stands for the recursive acronym Hypertext processor.
QR Code	Quick response code.
PSD	Photoshop Document.
PNG	Portable Network Graphics.
XML	Extensible Markup Language.

Table of figures

1 Introduction

The retail sales industry has undergone some noticeable transition in the array of payment methods over the years. It is imperative that an innovative payment solution is introduced to replace an old-fashioned payment system especially when there is growing demand for it. Technology has helped revamp the payment method of retail stores of today. The introduction of an innovative and sophisticated payment device like SACAT (Semi Attended Customer Activated Terminal) has added a more efficient means of making payments in a store compared to the cashier checkout system. Assuredly, the addition of a mobile self-checkout application in industry will be more prominent in the next few years.

The aim of this thesis was to develop a functional model for a mobile self-checkout application for a retail store of a client. The possible outcome of events when the key elements of the application i.e. the virtual shopping cart, store, employee and the customer interact with the application individually or collectively are clearly illustrated with UML diagrams highlighting various possible process flows.

The thesis also made propositions for the UI using prototypes to show the features on the application model as well as recommendations for technologies to be used at both the frontend and backend during development of the application.

1.1 Company in brief

The client’s franchise Bankynice Nigeria limited, is a retail store with three outlets strategically located in key areas of Lagos state, Nigeria. The company started its operation in July 2015. All three of its outlets have a satisfactory size of retail space, offering a wide range of products such as groceries, prepared foods, electronics, stationeries, beverages, small consumer goods etc.

The company has a medium size staff strength comprising majorly of the business owner, store managers, sales representatives, cashiers, securities and cleaners. It has impressively marked a niche for itself and is steadily becoming a key player in retail business particularly in the areas the outlets are located.

The franchise has been recording progressive growth since its inception with decent and competitive prices of products, efficient and innovative service, as well as excellent customer service. The company is presently in the process of exploring the option of further expanding the franchise by unveiling more outlets within and outside the city of Lagos, thus the need for the development of a self-checkout application that will inevitably boost the overall business operation of the franchise.

This thesis contains 4 sections. After the introduction, section 2 explores the working procedure of the application as well as the benefits of the mobile self-checkout application. Section 3 details the interactions between the key elements of the application namely; virtual shopping cart, store, employee and the customer with the mobile self-checkout application. Section 4 examines the technologies that can be used to develop the application and proposes a UI using prototype designs. Section 5 analyzed the whole scope of the thesis and made recommendations regarding suitable technologies needed to develop the mobile self-checkout application.

2 Product description

2.1 Mobile Self-checkout application

A mobile self-checkout application or the mobile form of a Semi Attended Customer Activated Terminal (SACAT) is a mobile payment system that allows a customer conveniently buy and complete the payment for products without the need to go through the cashier checkout point. This is achieved by using the mobile self-checkout application to scan the barcodes of products and adding them to the virtual shopping cart on the application and subsequently checkout after successfully making the payment [1].

2.2 Benefits of a mobile self-checkout application

The benefits of the mobile self-checkout application are significant. However, the prominent ones are:

- The tremendous slash in labour cost. With the customers acting like their own cashier courtesy of the mobile self-checkout application, less number of cashiers will be needed by stores where this application is implemented.
- The self-checkout application is mobile enabled thus makes it cost effective as it doesn’t require any procurement and installation of heavy and expensive machines or devices.
- It helps in a great measure cut off the long line at the cashier checkout points which can be frustrating at times for both customers and the cashier. Especially stores that frequently experience a large number of customers trying to checkout at the same time. The application will be a great value for the store during peak periods such as Christmas, Easter etc.
- The liberty of being able to complete the whole shopping process boost the customer's satisfaction, especially customers that are technologically savvy gives them some sort of self-fulfillment [2].
- It provides the customers privacy especially during the payment process.

2.3 Overview of the mobile self-checkout application

The mobile self-checkout application is operated via a virtual shopping cart. This virtual shopping cart is linked to the customer’s account which consists of the customer’s personal and payment card information created on the mobile self-checkout application. The payment card can be either credit or debit cards as well as other payment merchants like Paypal for a seamless payment system. The virtual shopping cart is associated with the store outlet visited by the customer. The customer must be within the peripheral of the store to view the products available for purchase, shop and subsequently complete the shopping process by successfully making payment afterwards.

In other words, the application allows the customer to scan through the products available in that store outlet, add or remove products of their choice and proceed to checkout by paying for the products bought all on the mobile self-checkout application. As for the efficiency criterion, the application reads and understands each product barcode to add or remove from the list of products in the customer’s virtual shopping cart.

The payment system a store incorporates in its franchise is an essential part of shopping. The system introduced to process payment must be reliable, effective, secured and easy to implement. The effectiveness of the payment process goes in a long way to portray how reputable a store is from the customer’s standpoint as far as paying for products is concerned. The mobile self-checkout application not only measures up to this criterion but also makes sure the customer is not double billed.

The application also creates room for the customer to update payment information with a new one before making a purchase while shopping. It identifies and prompts the customer to either cross check and input the correct payment information or input a new valid payment information when the current payment information is either wrong or cannot be used for purchases e.g. due to expired credit card or debit cards, security limit violations etc.

3 Product specification

3.1 Virtual shopping cart

The virtual shopping cart is one of the key actors of the mobile self-checkout application. The interaction between the application and the virtual shopping cart in terms of the process flows indicating a feasible chain of event can be illustrated with UML diagrams. Before customers can have access to the virtual shopping cart and shop there are some preconditions that need to be met before products can be scanned and added to the virtual shopping cart [3]. These preconditions require the customer to download and install the application on any mobile device of their choice. The customer should also make sure they are in the same proximity distance of the store outlet they intend to shop to view the product list and subsequently shop.

When the application is launched after download and installation, it lists the store location of the nearest outlets the application is initiated from. When the customer selects the outlet they intend to shop, an account page appears where customers can create an account if they are using the application for the first time.

Customers can log in if they already have an account. The application prompts the customer to re-login using correct account information if an invalid username and password is inputted as shown in figure 1.

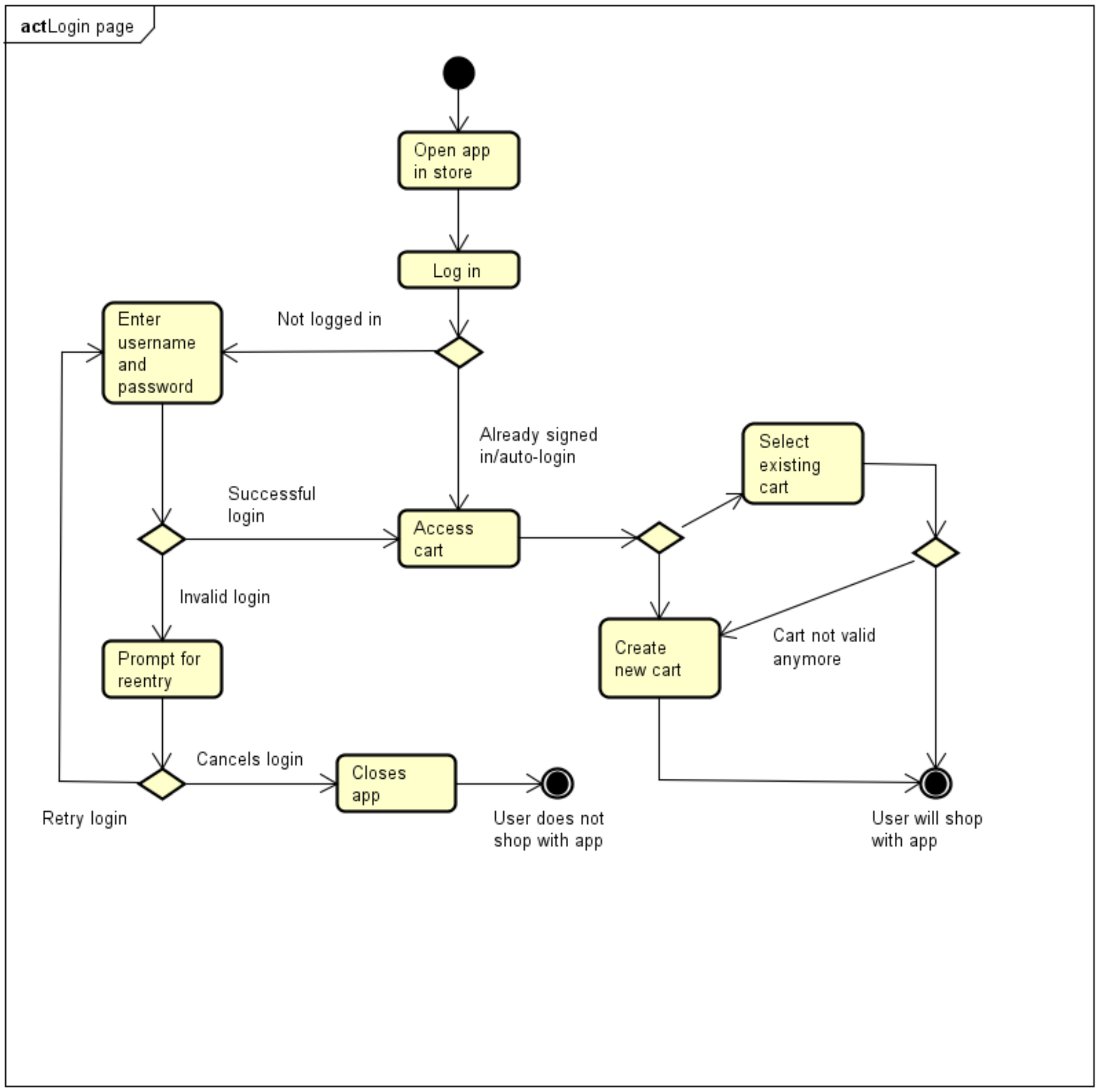


Figure 1. Activity diagram of login page.

When the customer creates a new account or logs in successfully, the virtual shopping cart offers the customer the liberty to scan any product of their choice and add or remove them.

The store clerk returns products that have been removed from the virtual shopping cart to their original position on the shelf if they are not returned by the customer as illustrated in figure 2.

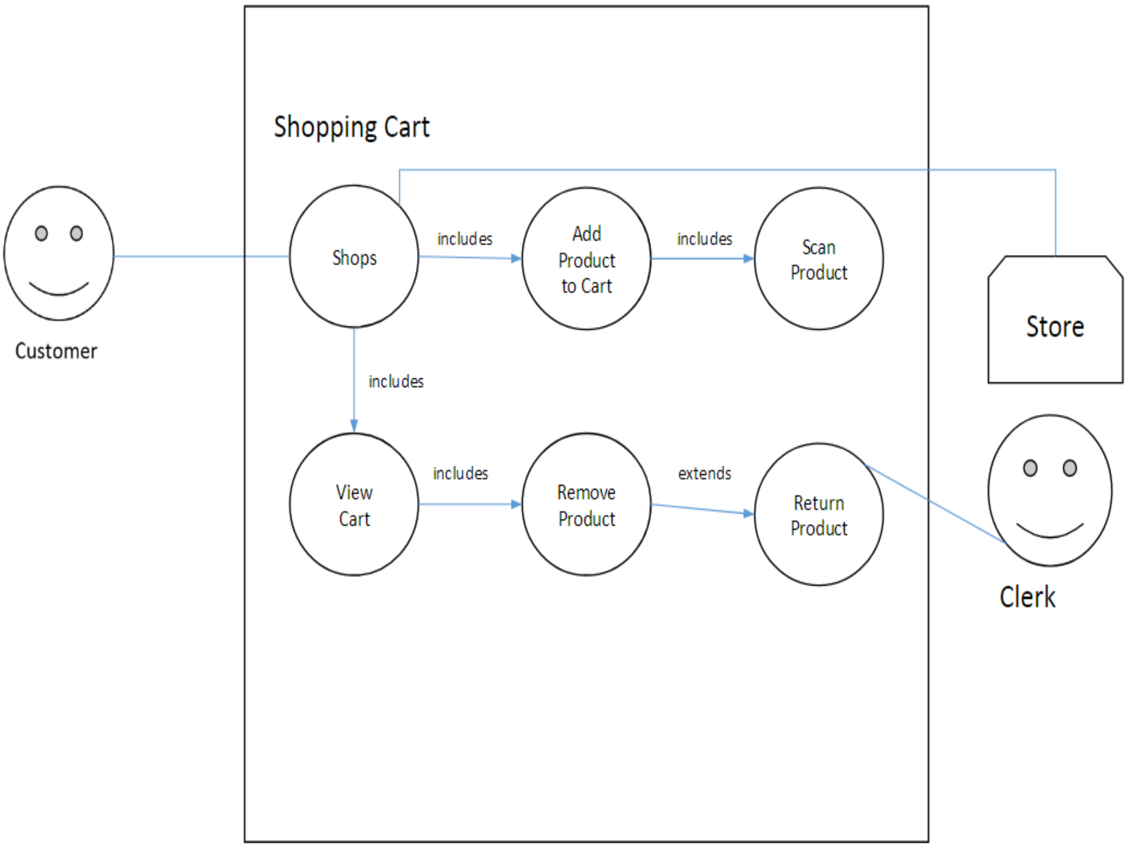


Figure 2. Use case diagram of virtual shopping cart I.

The virtual shopping cart automatically identifies the store outlet and loads up the product list of the store outlet from where the customer is initiating the mobile self-checkout application. The store outlet allows several customers to shop and use the mobile self-checkout application at the same time. However, the application only allows one virtual shopping cart per customer. Any specific number of clerks can return any item(s) not eventually returned by the customer after removing any product from the cart. Moreover, the application also notifies the customer via a prompt when there is an issue reading the product bar code as depicted in figure 3.

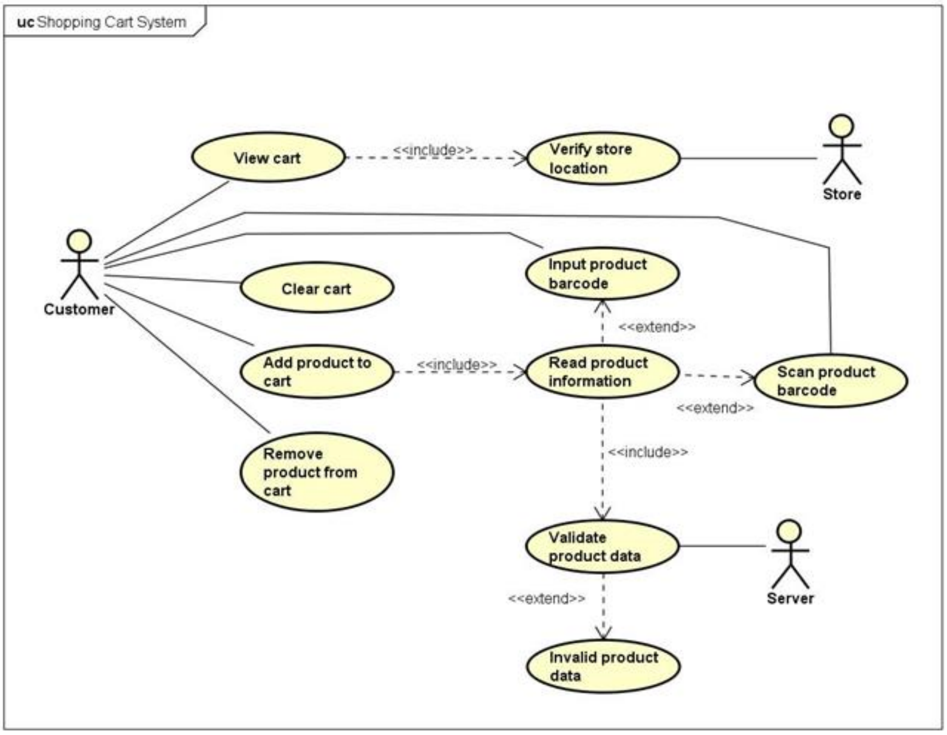
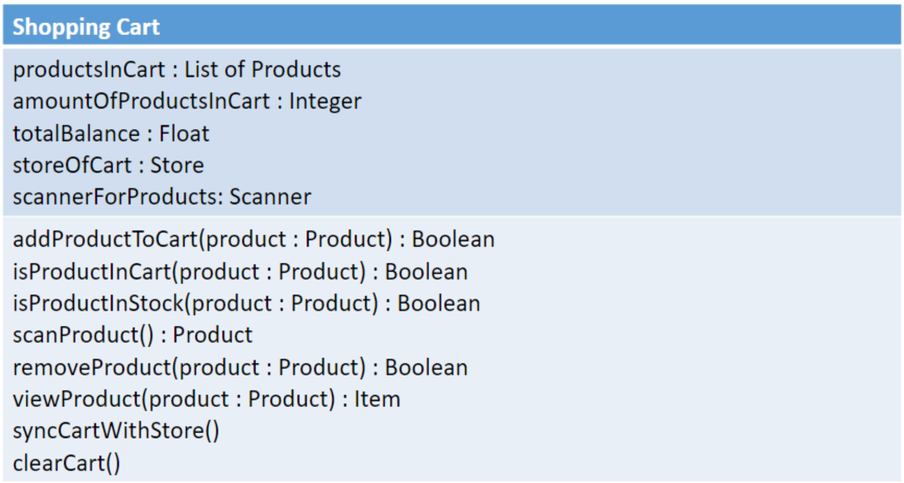


Figure 3. Use case diagram of virtual shopping cart II.

The synchronization of the virtual shopping cart with the store outlet is important for inventory updates and payment process.

The class diagram can be derived from the use case diagrams in figures 2 and 3 as shown in figure 4.



Shopping Cart

productsInCart : List of Products
amountOfProductsInCart : Integer
totalBalance : Float
storeOfCart : Store
scannerForProducts: Scanner

addProductToCart(product : Product) : Boolean
isProductInCart(product : Product) : Boolean
isProductInStock(product : Product) : Boolean
scanProduct() : Product
removeProduct(product : Product) : Boolean
viewProduct(product : Product) : Item
syncCartWithStore()
clearCart()

Figure 4. Class diagram of virtual shopping cart.

The component diagram of virtual shopping cart detailing the overall shopping process can be derived from the use cases as depicted in figure 5.

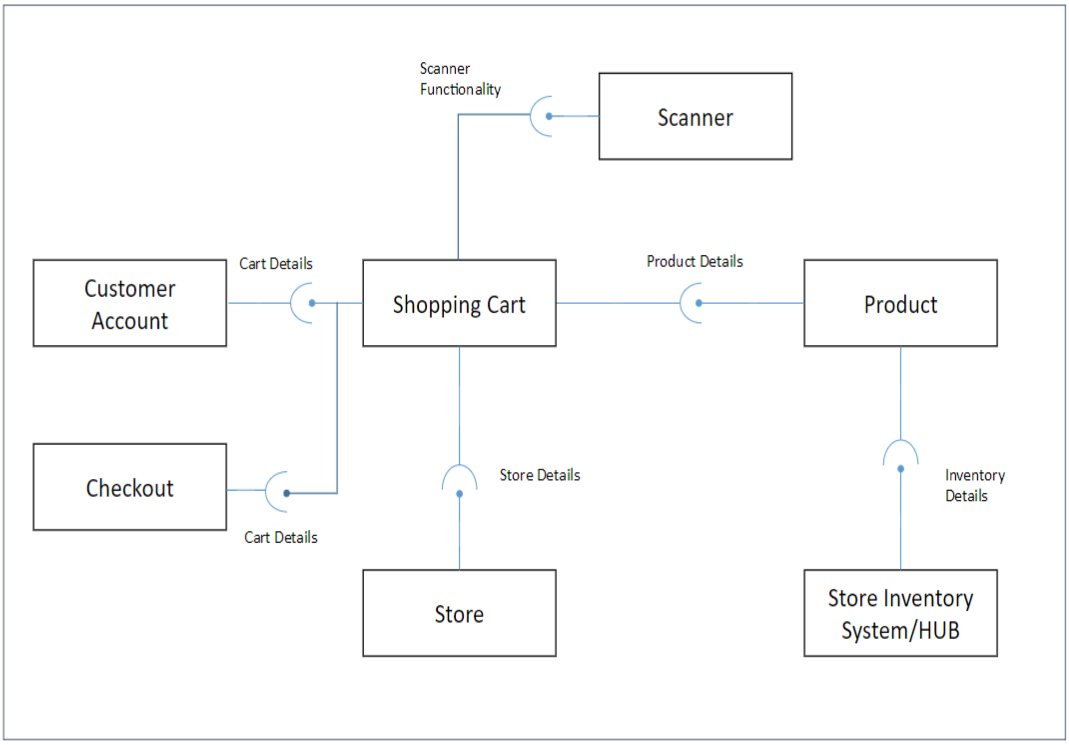


Figure 5. Component diagram of virtual shopping cart.

3.1.1 Virtual shopping cart: Adding products

On successful account creation, the customer will be able to view the virtual shopping cart and product list of the store outlet. Figure 6 depicts the activity diagram showing the addition of product to the cart.

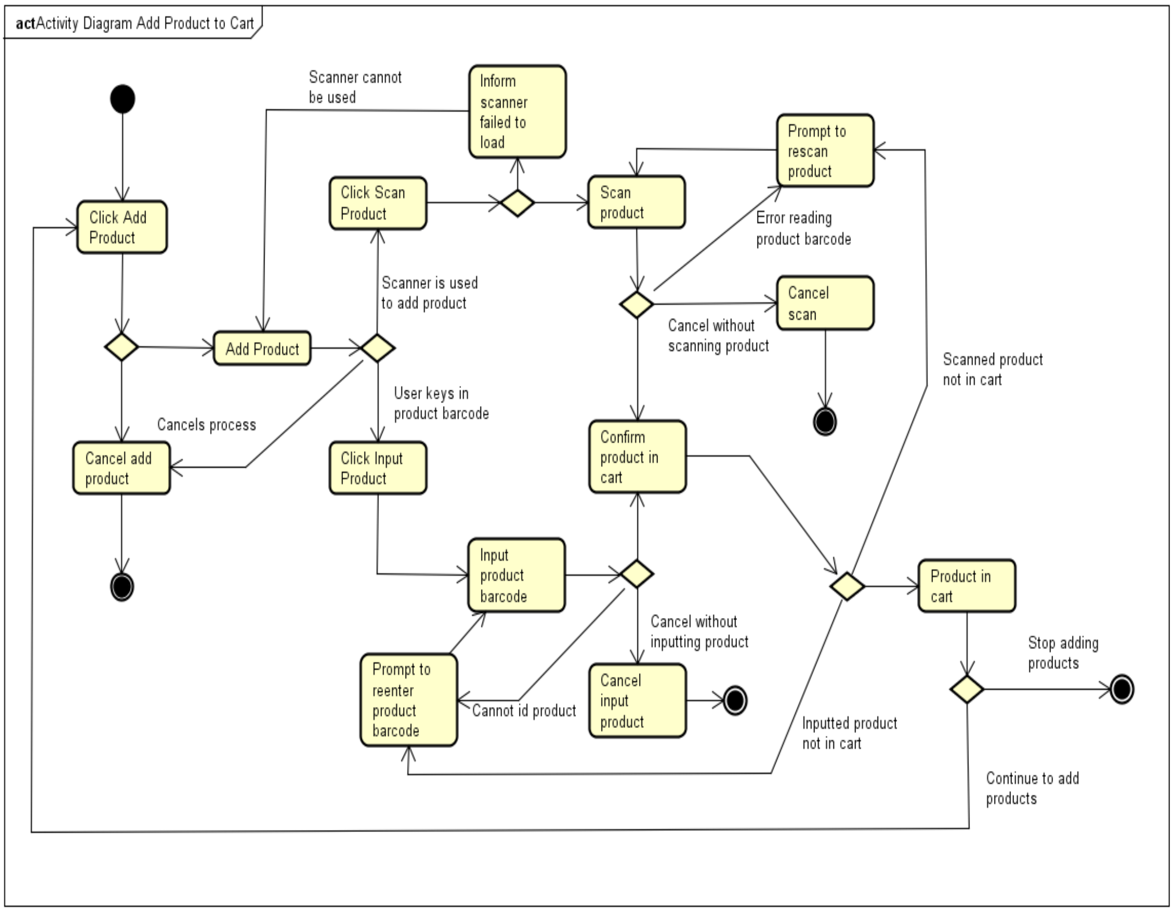


Figure 6. Activity diagram showing the addition of product to the cart.

As shown in figure 6, when the customer sees any product of interest and wants to buy it, the customer can conveniently do so by initializing the “scan barcode” feature on the application to scan the product and subsequently click the “add product” option to add the product to the virtual shopping cart. The application also allows the customer to key in the product barcode manually. The opening and closing of the scanner is handled internally. Depending on the scenario the barcode scanner can be kept on if the customer wants to keep adding more products. When the barcode of a product is not correctly scanned or inputted correctly, the application prompts the customer to rescan the product barcode correctly.

3.1.2 Virtual shopping cart: Removing products

In this instance the customers will still be able to view the virtual shopping cart and remove a product of their choice by selecting the product and clicking on “remove product” option without the need to rescan since the product has been scanned earlier and already exists in the virtual shopping cart. When the customer accepts the prompt to remove the selected product, the application confirms the removal and updates the contents of the virtual shopping cart.

The product can thereafter be returned to its original position by the customer or any of the store clerk afterwards. The use case diagram showing the process flow when removing products from the virtual shopping cart is illustrated in figure 7.

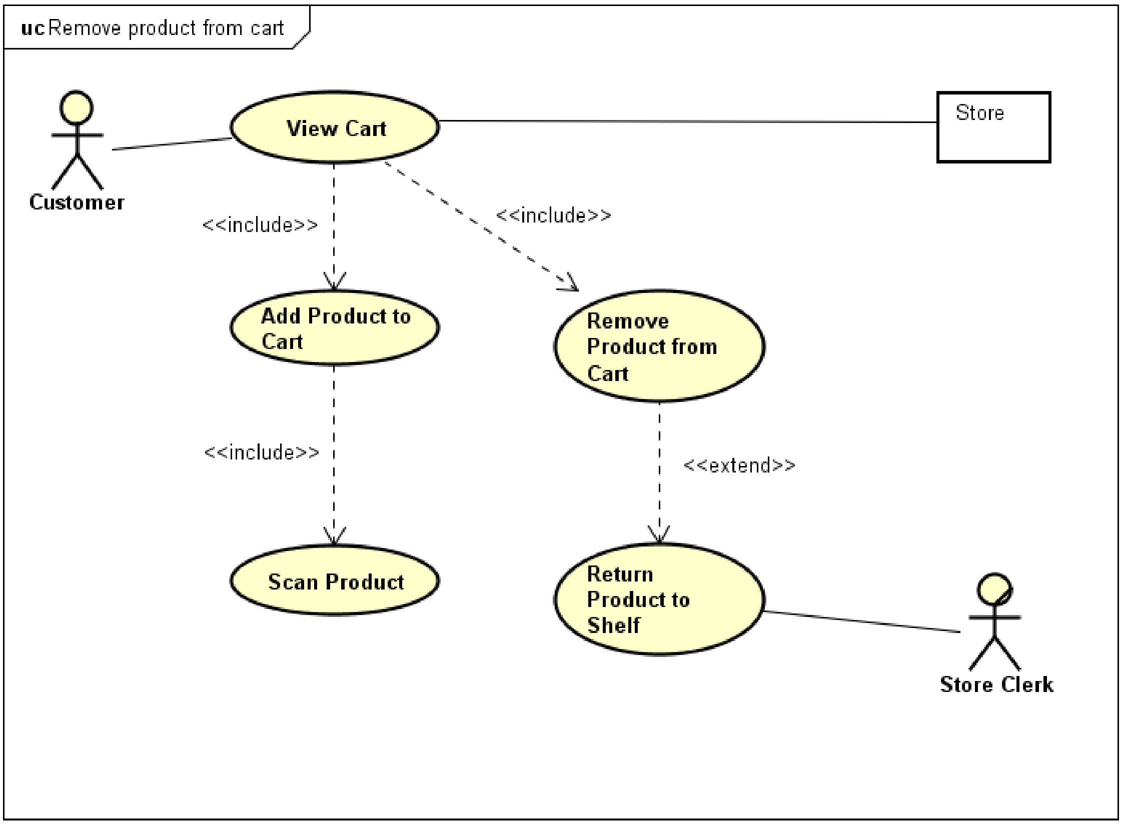


Figure 7. Use case diagram showing the removal of products from the cart.

3.2 Store

The store is another key actor of the mobile self-checkout application. The products available for purchase are categorized. Thus, customers can browse and know if their product of interest is available faster. The process flow for the sequence of activities relating to the interaction between the application and the store evolves around use cases such as inventory management, customer account registration, payment verification process and product life management. The role of the store in this instance also extends to important documentation such as generation of sales report.

3.2.1 Inventory management

The application affords the store the leverage to update and manage its inventory efficiently without oversight. The store directly engages the suppliers who are responsible for delivering the various products ordered by the store. This event is triggered by the employee in charge of managing the inventory. The employee checks the inventory for products that needs to be restocked. The supplier in charge of delivering the product is selected and a purchase order is generated and sent to the supplier which is subsequently documented. [4]

The activity diagram showing the process flow of these events is depicted in figure 8.

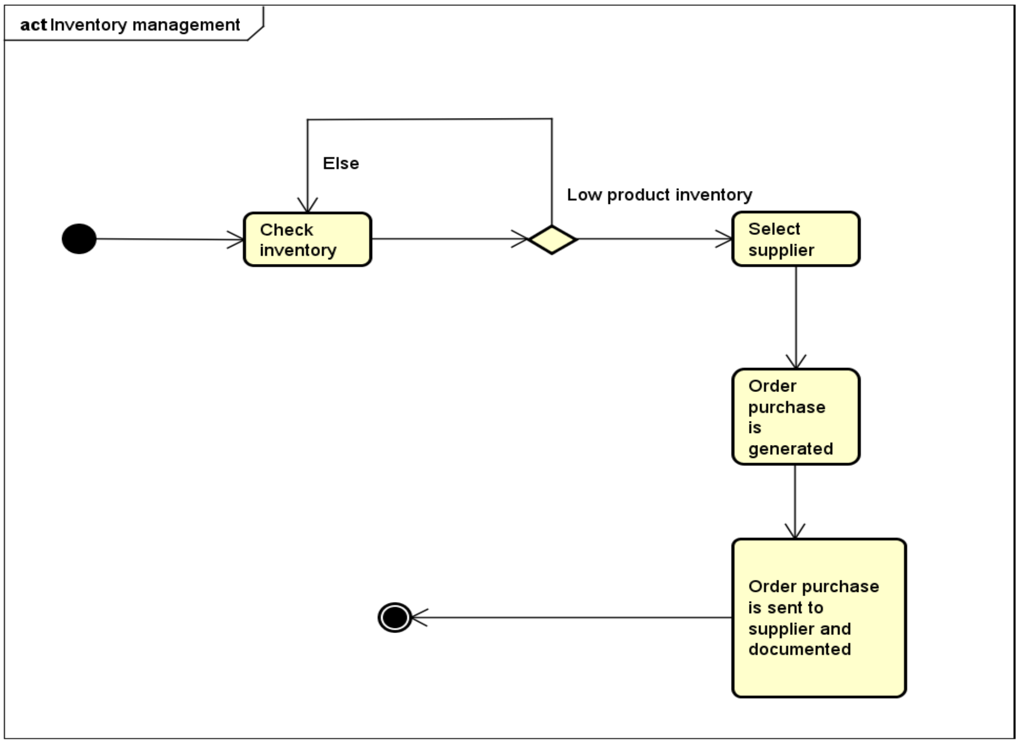


Figure 8. Activity diagram for inventory management.

3.2.2 Component diagram of the store

The component diagram of the store as delineated in figure 9 depicts stored customer and transaction records. It also shows the interaction between the shopping cart, the supplier and the store clerk or employee with respect to the sequential connection between the inventory details, cart details, product details.

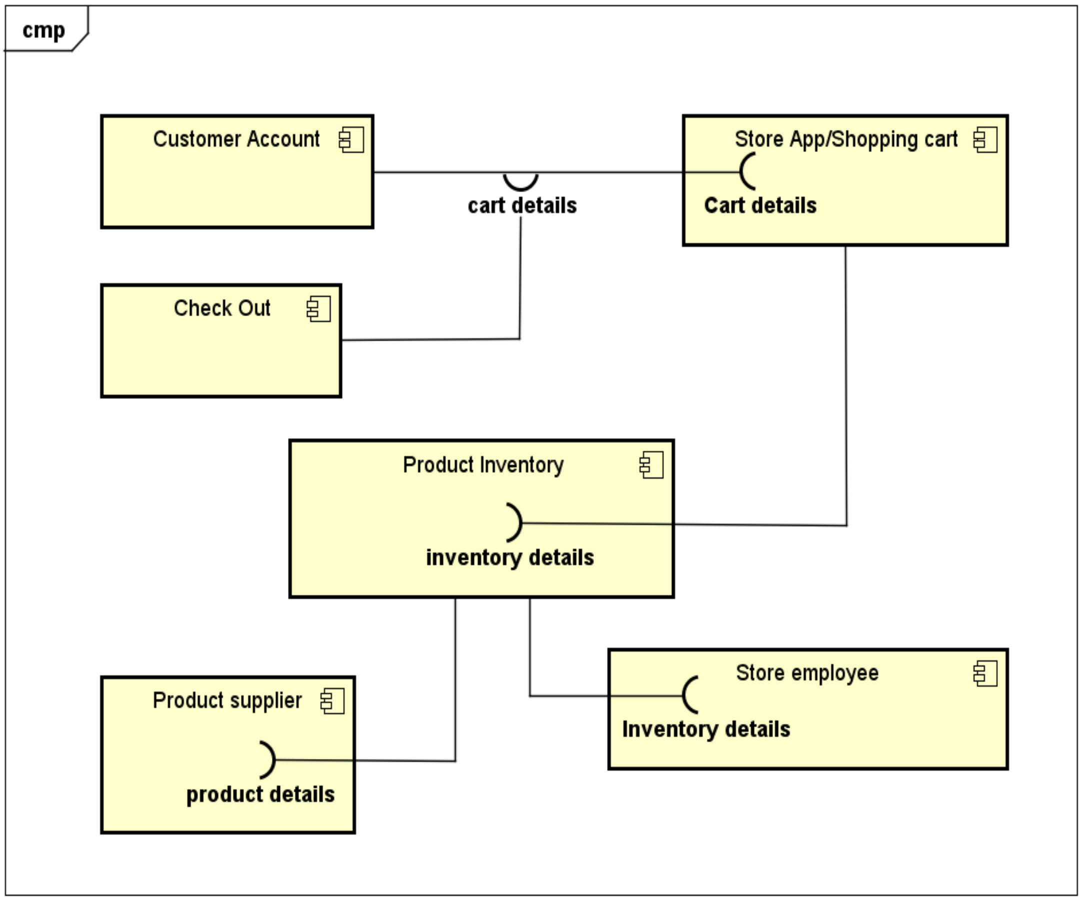


Figure 9. Component diagram of the store.

3.2.3 Verification

The importance of verification cannot be over emphasized. The store needs to verify that all transaction and payment by customers are thoroughly scrutinized and approved before a customer exits the store. Therefore, security put in place by the store to curb credit card identity theft must be top notch, high technological driven and be able to synchronize effortlessly with the mobile self-check application.

Payment card infiltration of any sort e.g. attempts of using expired, invalid or stolen credit cards, debit cards or illegal attempts of undermining any features are adequately checked by the security features of the application.

The mobile self-checkout application can conveniently carry out this payment card security task. The sequence of events the customer undergoes from account creation on the application to eventual payment verification process by the application is elucidated below as follows:

- User information registration.

  The customers' user details upon creating a profile account on the application are validated and sent to the server where it is stored.

- Card registration.

  The card information is validated by the card issuer. Upon a successful check, the card information is sent to the store and stored in the server. Alternatively, if the check is unsuccessful the card issuer informs the user and prompts the customer to input a valid card information.

- Adding of products.

  Customers have access to the virtual shopping cart when they successfully create an account or login in with an existing account. They can add products to cart by using the scanner feature of the application. When the customer is done with adding products to the cart. They can move to the payment page by clicking on the "checkout" option.

- Payment operation.

  As figure 10 illustrates, when customers initiate payment, the application directs the card information to the card issuer for validation. Upon successful payment verification, the customer can proceed to checkout otherwise the customer is informed by the card issuer after a failed verification to input a valid card information [5].

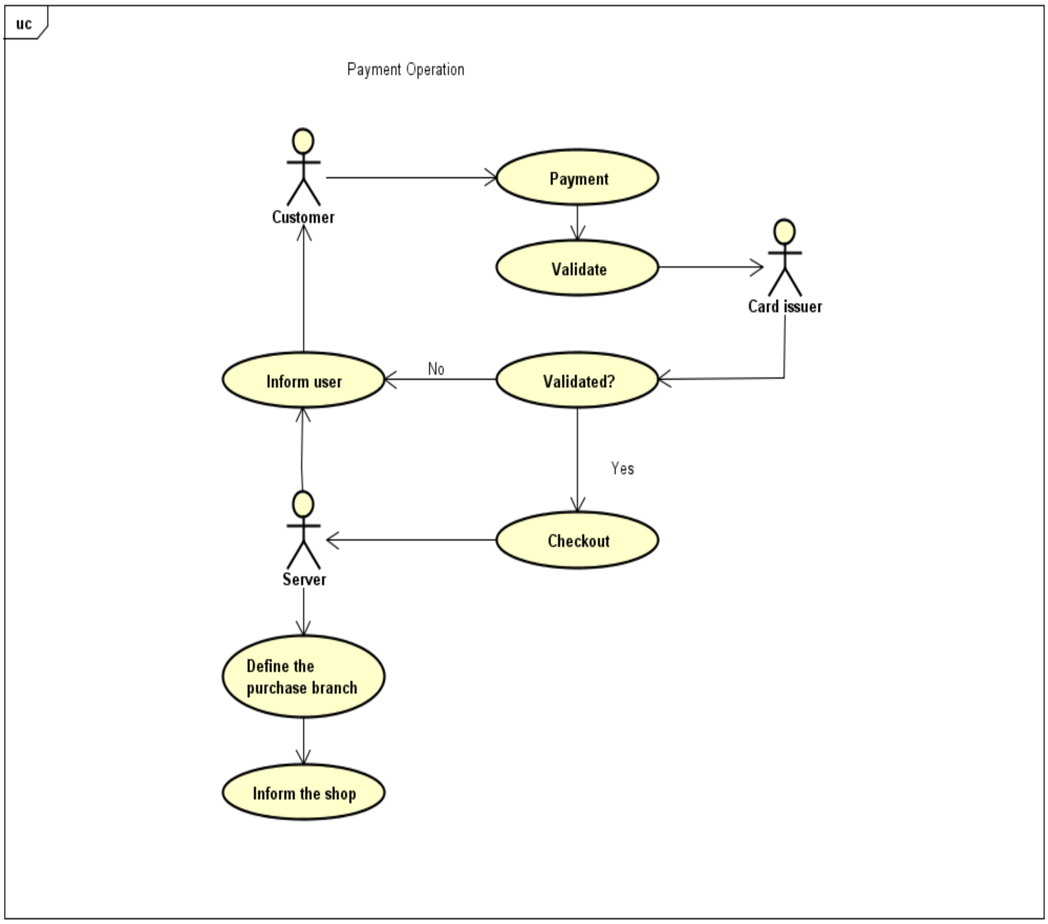


Figure 10. Use case diagram for payment operation.

- Checkout.

  When the payment is verified, the application generates a payment confirmation receipt which displays the QR code and transaction number. The customer can exit the store afterwards.

3.3 Employee

The employee as another key actor of the application emphasizes more all the activities carried out in the store especially during routine checks of products and restocking of products. The chain of events that occur during product routine checks as well as restocking of products can be elaborated with UML diagrams e.g. class and use case diagrams.

3.3.1 Product life management

At regular intervals, the store needs to fashion out a system to undergo a product routine check. During this routine check, products due for replacement either because of they have been expired or need to be discontinued can be effectively resolved and subsequently removed from the shelves.

Identifying and removing expired products is mandatory especially with consumables and cosmetics products. It not only prevents instances of customers being exposed to the dangers of consuming them, but the reputation of the store is maintained by making sure all safety standards are complied with.

A store can discontinue a product for various motives. One is when the demand of such product decreases. When a product remains on the shelf for a long period without being sold, it does not augur well for the store. The decision by the store to discontinue such a product is indeed in order. The store can also decide to discontinue a product that is out of current trend e.g. clothing items [6].

Moreover, a store is obligated to discontinue a product if the manufacturer ceases producing the product due to reasons like product defects posing harmful threats, manufacturing company folding up etc. [6].

A class diagram can be derived from these events as depicted in figure 11.

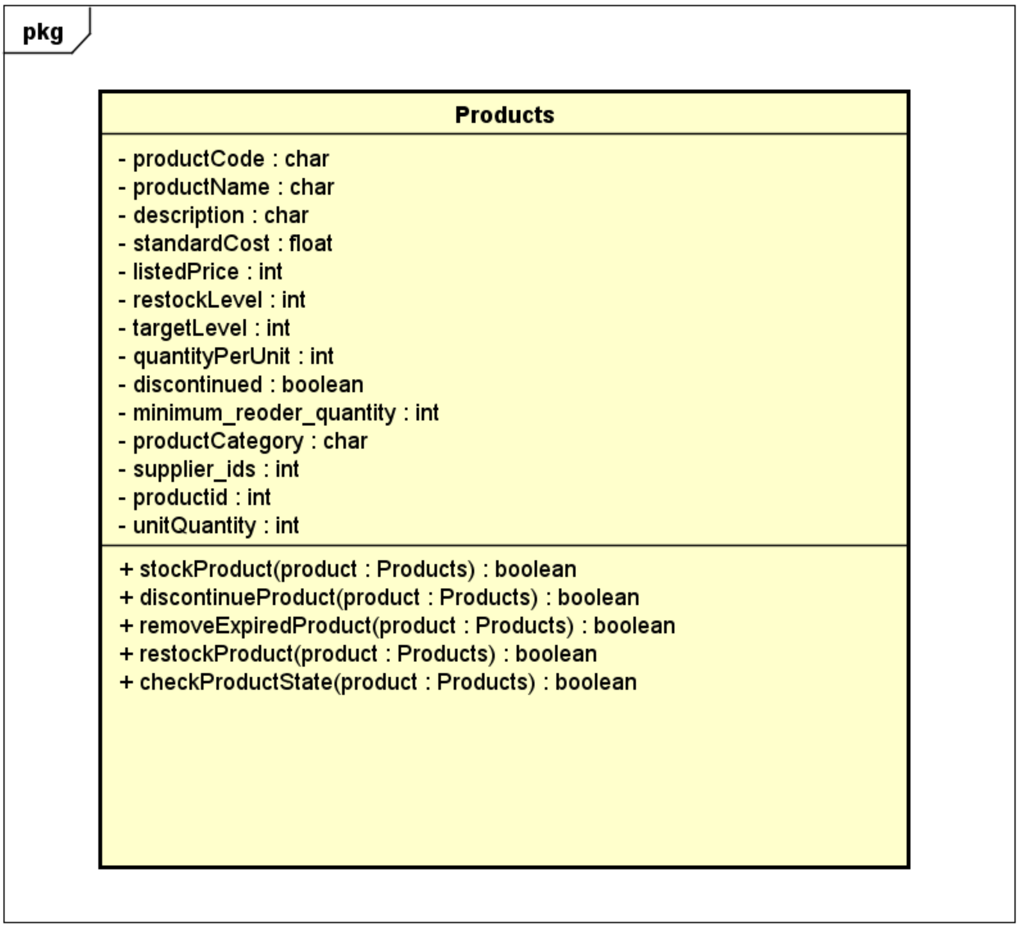


Figure 11. Class diagram for product life management.

3.3.2 Restocking of products

This section of the thesis describes the process flows the store takes in the event of restocking products in the store which is eventually used by the application. Moreover, the use case diagram of this event defines the sequence relationship between the store employee responsible for product restocking and the supplier.

As shown in figure 12, the store employee saddled with the responsibility of restocking products carries out the following task; checking inventory via the application to determine depleted products that requires restock, order for products and make payment for products. On delivery of products by the supplier, the products are checked for defects. Damaged products are reported to the supplier for replacement while the ones in good condition are put up on the store shelves for purchase. [7]

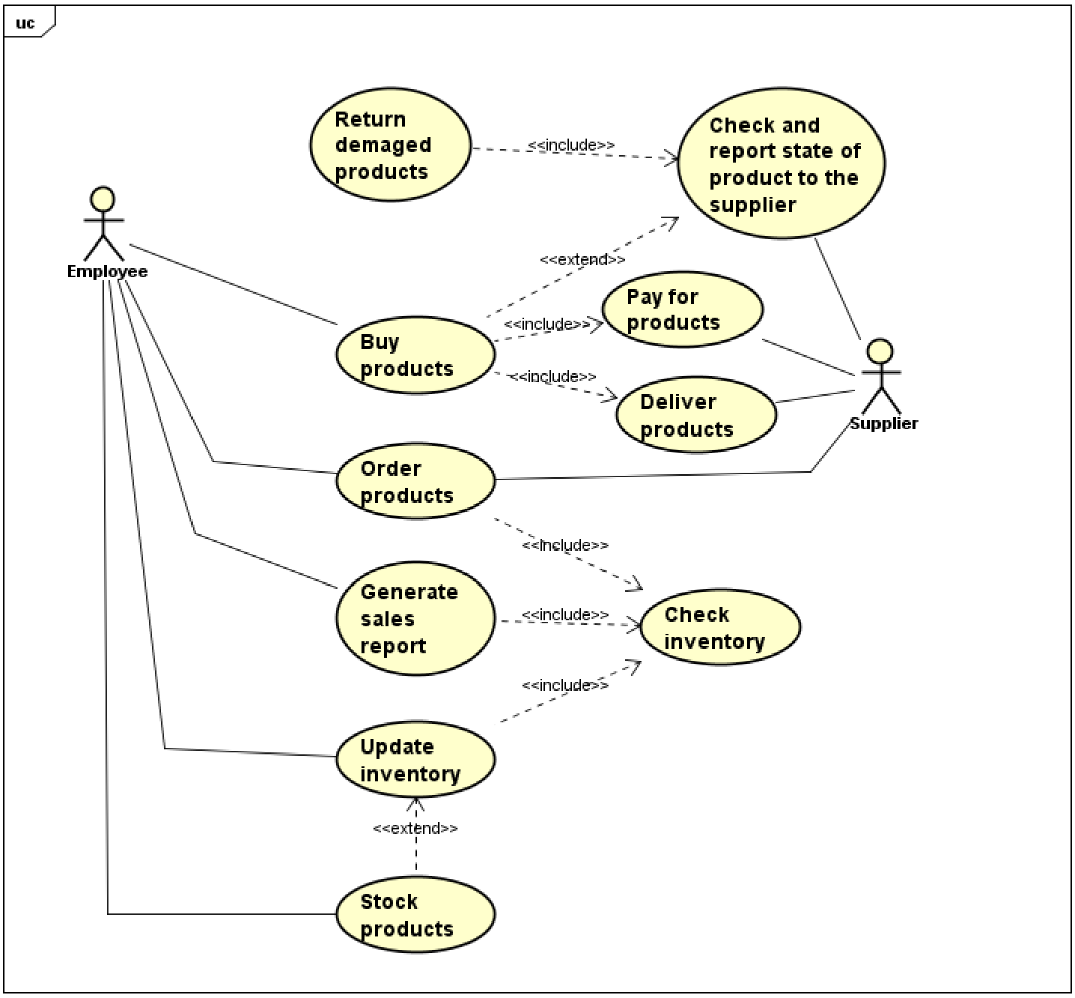


Figure 12. Use case diagram for restocking of products.

It is important to note that it is through this process the store via the application updates its inventory system based on sold products.

3.4 Customer

The customer is the initiator of the whole shopping process. The customer is one of the primary motives for the creation of the mobile self-checkout application. In this section, this thesis highlights the role the actor “customer” plays as it relates to the mobile self-checkout application.

The customer on their part must endeavor to satisfy the mandatory precondition before proceeding to use the application. These preconditions as earlier discussed require the customer to be within a certain radius of the store. The customer should download the application on any mobile device of their choice and create a user account profile with their payment information on the application after a successful download and installation of the mobile self-checkout application on their mobile device [3].

3.4.1 Customer: Adding and removing products

The customer can add any product of their choice at the store through the following simple steps:

- Customer *views* existing virtual shopping cart.
- Customer *clicks* “Add Product” to *add* product to cart.
- Customer *scans* barcode of product.
- Customer *confirms* product to be added to cart.

A Customer can successfully remove any product from their virtual shopping cart as via the following steps.

- Customer *views* virtual shopping cart.
- Customer *clicks* “Remove Product” to *remove* product from cart.
- Customer *accepts* prompt to *remove* product.
- Customer *returns* product to its place in store.

The use case diagram for adding and removal of products from the customer is shown in figure 13.

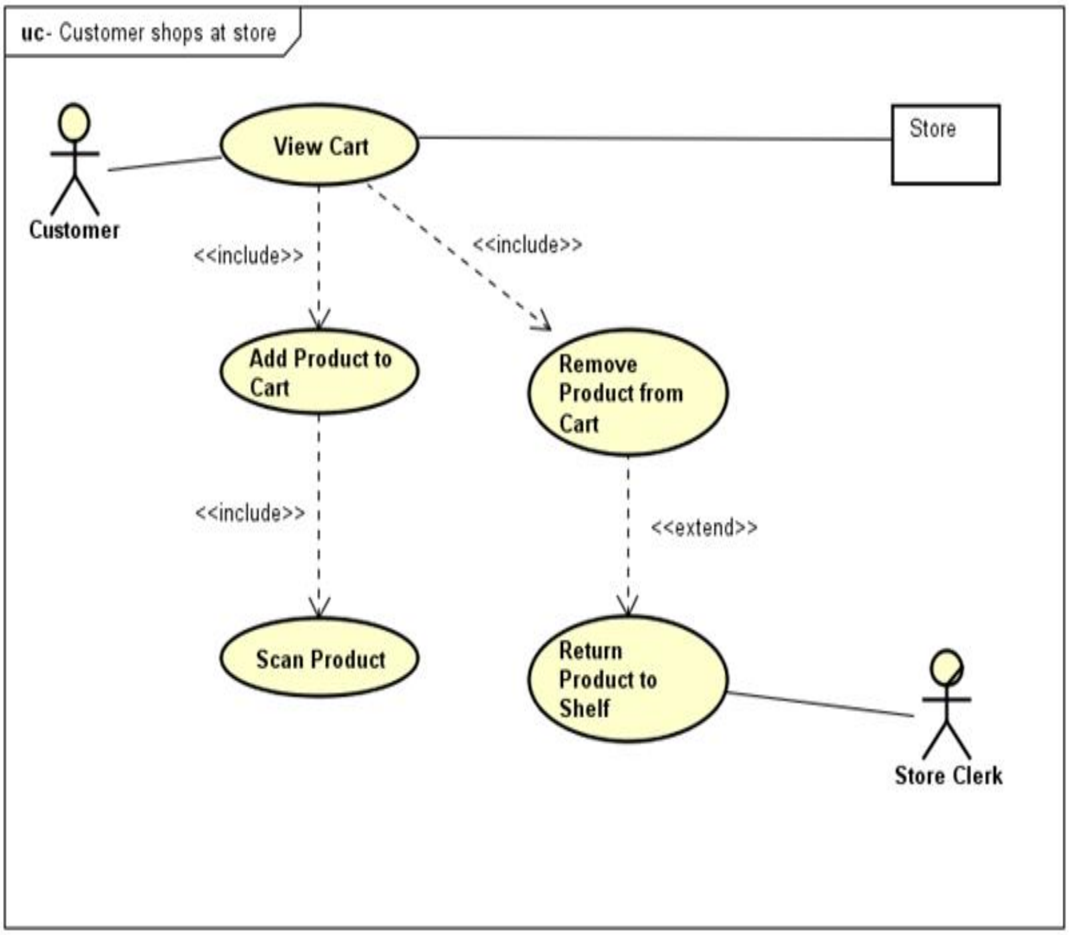


Figure 13. Use case diagram for adding / removing product from the cart.

3.4.2 Exception flow

There are instances where the mobile self-check application experiences difficulties in either adding or removing products in the virtual shopping cart [3]. These exceptions may be due to various factors.

There are occasions when the scanner is unable to read the product barcode. This may be because of an improper product scan by the customer. When the customer does not align the product correctly with the scanner. The application in this instance will prompt the customer to re-scan the product as depicted in figure 14.

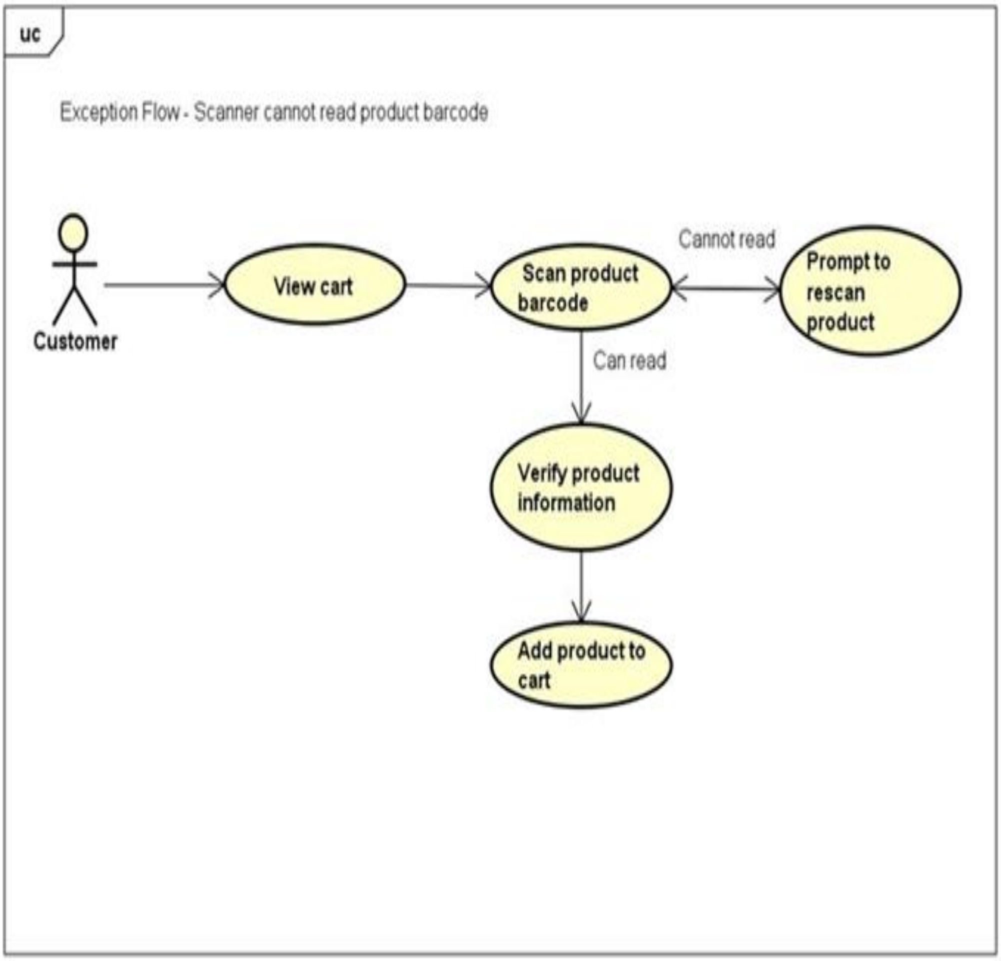


Figure 14. Use case diagram when scanner cannot read barcode.

The customer can also have trouble in viewing the shopping cart. This may arise because of software error e.g. out-of-date software version or GPS store location error. In this instance, the application promptly informs both the customer and the store. After a successful software update, the customer can delete the previous virtual shopping cart and create a new one afterward if the application crashes while the customer is shopping. Other reasons are when the customer tries to create a second instance of the application while shopping at the store. The application checks for an existing session and prompts the customer to use existing the existing session. The exceptional flow in this instance is shown in figure 15.

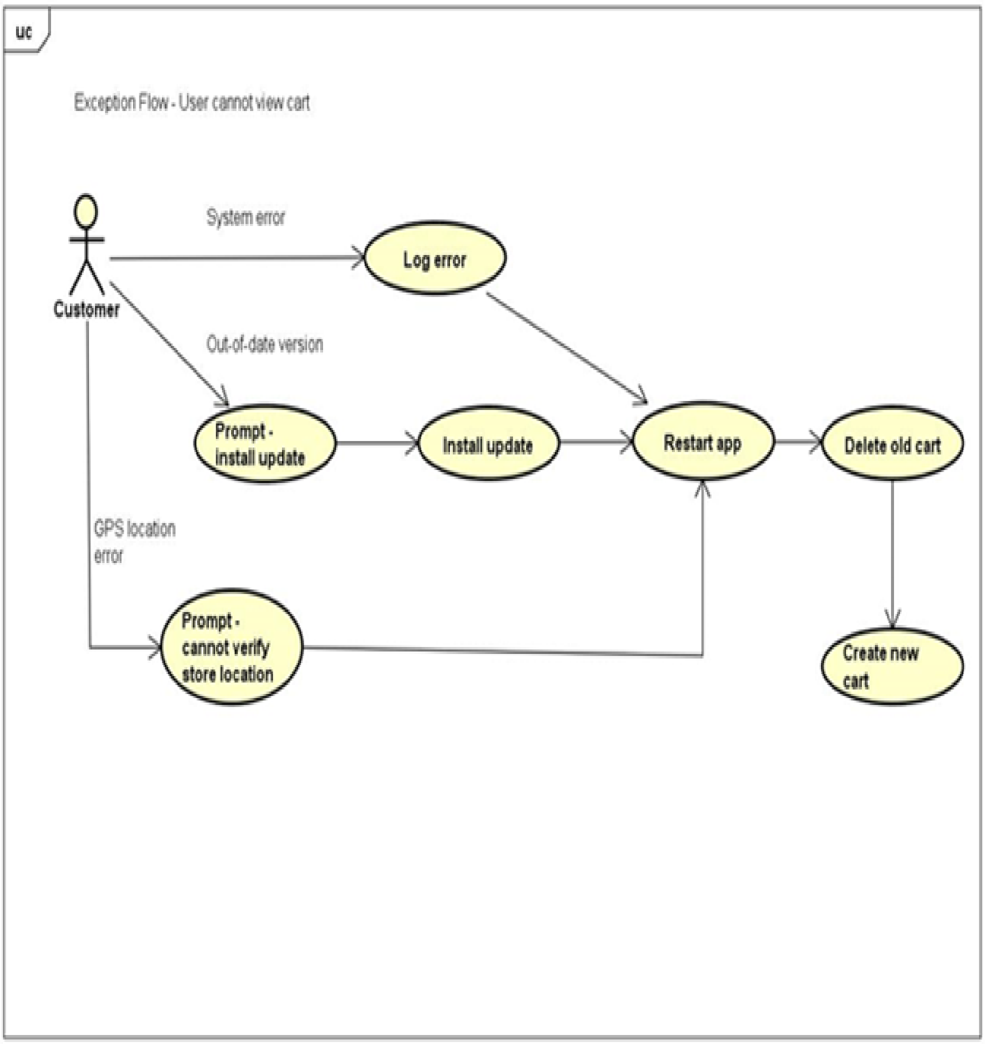


Figure 15. Use case diagram when customer cannot view cart.

4 Implementation planning

This thesis has so far elaborated on the functional features of the mobile self-checkout application. Use cases, state diagrams, activity diagrams and component diagrams have been employed to illustrate the pattern of flow of actions associated with each "actor" interacting with the mobile self-checkout application. The previous section also analyzed their functional cohesiveness as a unit with relevant figures for illustration.

However, this section of the thesis probes more into the technical facet e.g. technology used as far as the future development of the mobile self-check application is concerned. The author has disintegrated the implementation process of the mobile self-checkout application into two integral units. These units are the frontend application and backend application been proposed for the development of the application. A recommendation on how to piece together the technologies used to output a working mobile self-checkout application ready for testing and subsequently for use by customers in the client's franchise will also be given.

4.1 Backend application

Besides assembling an adept development team for the development of the application, it is important the team harmoniously agree and settle for technologies that will best achieve the goal of the project with optimal result. For this project therefore, the application should be compliant across all platforms i.e. Android, IOS etc. as well as most devices i.e. mobile phone, tablets, iPad etc. The backend applications are web based, they can be developed using technologies such as MS SQL, MySQL etc. The database for the application should be stored in the cloud. A cloud storage system is preferable because it outweighs the conventional in-house storage system by being more efficient, cost effective and reliable.

The database will house customers' user information (name, address etc.), payment information, store inventory etc. Even though the payment information of customers is also stored in the database as well, the CVV, card certification value is not. The credit card number will be encrypted revealing only the last four numbers. These are security measures to protect customers' bank details to curb the issue of identity theft.

4.2 Frontend application

The frontend application often comes with three options for software development of most applications. The frontend of IOS applications are natively developed with object c or swift programming languages. XCode which are source codes for object c or swift programming language, they have storyboards inherent in them that are used to design the UI. Android application on the other hand, the frontend is natively developed with Java technology mostly. Android studio is the source codes for Java technology. It has XML files inherent in them which is used to design the UI. The third option is to develop a hybrid application for the frontend of the application. Hybrid applications are those applications that when developed can be accessed by IOS and Android platforms unlike object c or swift which can only be accessible by IOS platforms, while Java technology is only accessible to Android platforms. Some programming languages used to develop hybrid applications include; IONIC, CORDOVA, TITANIUM etc.

4.2.1 User Interface planning

It is undoubtedly necessary that any self-checkout application a retailer introduces for his or her franchise should have noticeably user-friendly user interface features such as: [8].

- Include only important fields e.g. account creation should have only relevant fields like name and contact details. Information such as date of birth, hobbies are not relevant.
- Shop and checkout as a guest. The application should provide an alternative route for customers that wish not to create an account before shopping and checking out.
- Assurance. There should be messages assuring the customers that their sensitive information such as their bank details are protected and secure from theft.
- It should be easily accessible and managed across all platforms e.g. IOS, Android etc.
- Directives should be clearly understandable. Customers should be able to know their progress i.e. they should know where they are and the next step.

4.2.2 User interface prototype

In this thesis the author has employed the use of sample prototype to further illustrate how the features inherent in the mobile self-checkout application should look like. As figure 16 illustrates, on first initiation of the application a simple welcome screen pops up and requests the user to download and install the application to any of their mobile devices and platform. After installation the customers can view all the store outlets available and their distance from their current location as depicted in figure 17.

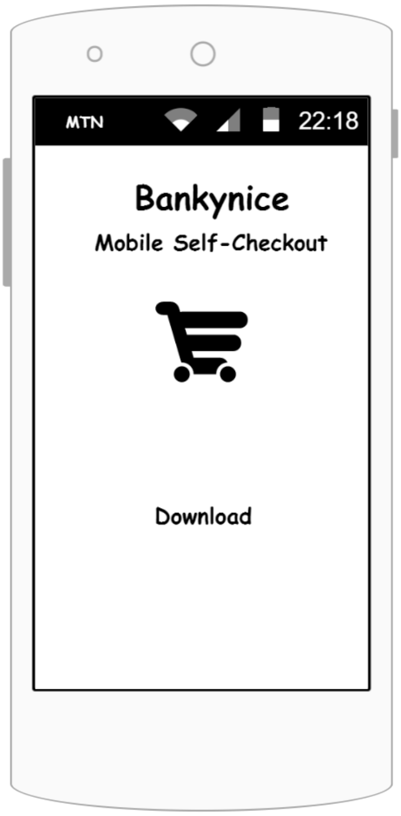
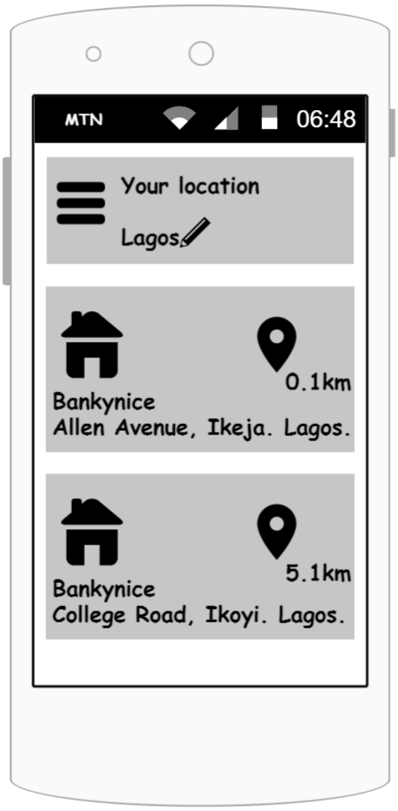


Figure 16. Welcome screen of the Application.

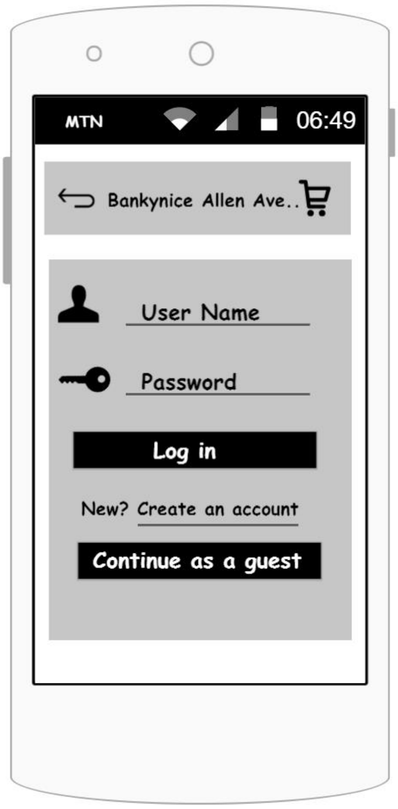


Bankynice
Allen Avenue, Ikeja. Lagos.
0.1km

Bankynice
College Road, Ikoyi. Lagos.
5.1km

Figure 17. Store outlet finder.

The user will have options to create an account, sign in if they already have an account or continue to shop and subsequently checkout as guest if they wish to shop without creating an account as illustrated in figures 18 and 19. If the customer chooses to shop as a guest it skips the account creation page.



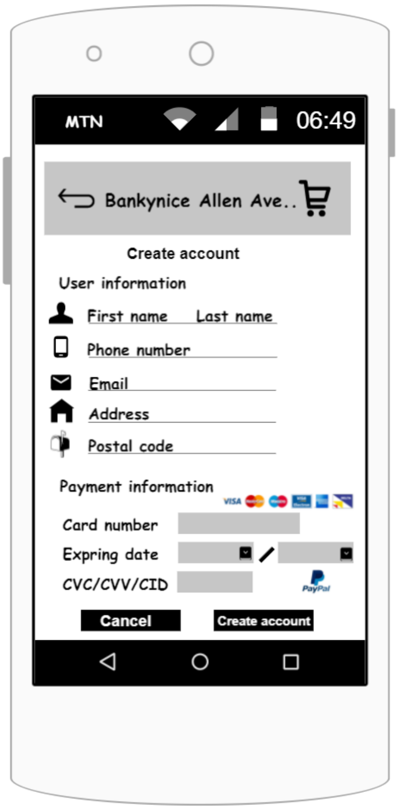
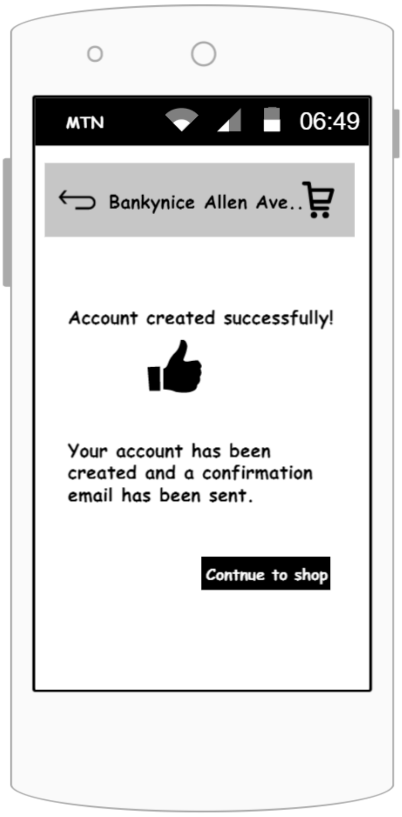


Figure 18. Login page.

Figure 19. User and payment information page.

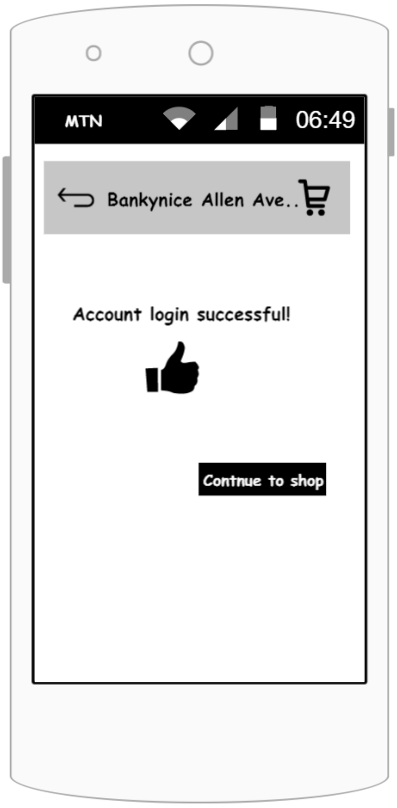
As figures 20 and 21 depict, after successfully creating an account or login with an already created account a confirmation indicating a successful account creation or login appears and there is also a direction for customers to move to the next step.



Account created successfully!

Your account has been created and a confirmation email has been sent.

Figure 20. Account confirmation page.



Account login successful!

Figure 21. Login confirmation page.

The next step reveals the product list of the outlet the customer is using the mobile self-checkout application from as illustrated in figure 22. Customers will be able to view all the products in the store which are categorized in a nested list. They are also able to start shopping by clicking the scar barcode button or checkout if they so wish.

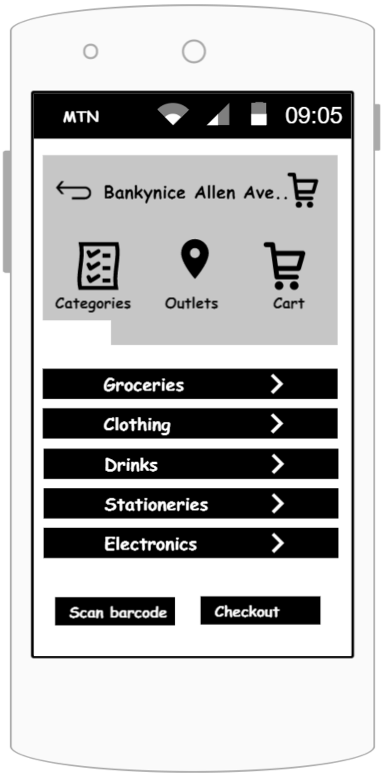
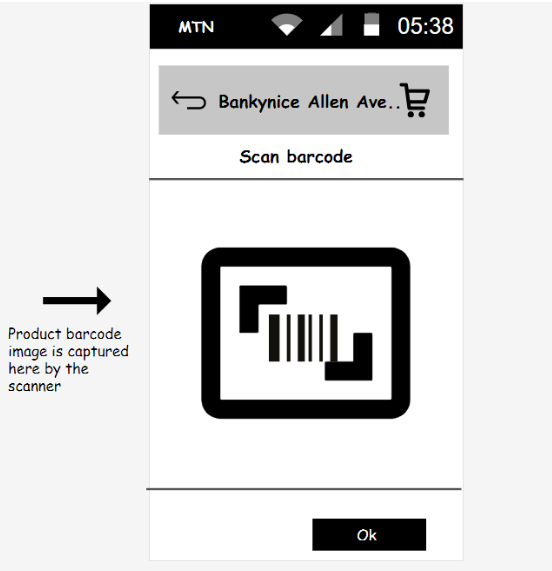


Figure 22. Product list page.

As figure 23 depicts, the customers will be able to scan their product by placing the product correctly on the designated area for the scanner to pick up the barcode. The next phase is to show the details of the scan product (product name and price), a prompt for customer to either add or remove the product from the virtual shopping cart as shown in figure 24.



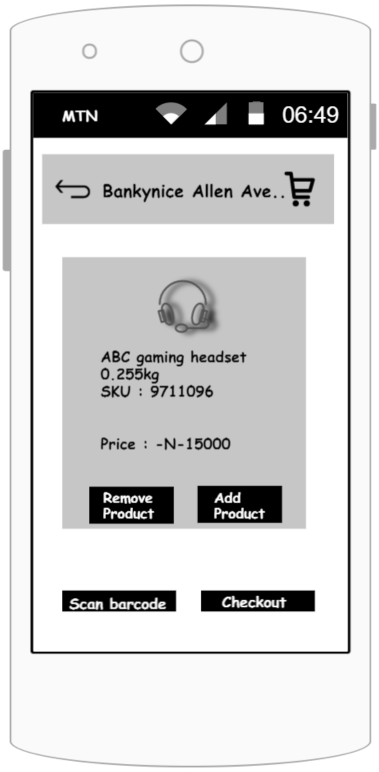
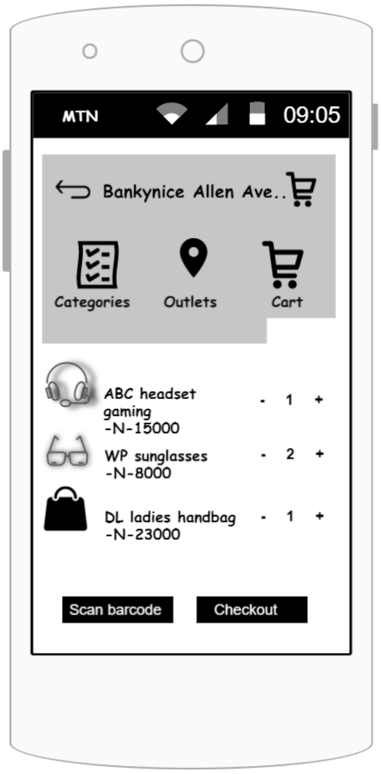


Figure 23. Barcode scanning page.

Figure 24. Add/remove product page.

The next phase shows all the products that have been added to the virtual shopping cart. The customer will be able to add or reduce the amount of the product, as well as the scan barcode option to continue shopping or checkout as illustrated in shown in 25. Clicking the checkout button at this point reveals a page that lists all the products in the cart and a “proceed to pay” button as illustrated in figure 26.


MTN
09:05
Bankynice Allen Ave..
Categories
Outlets
Cart
ABC headset gaming -N-15000 - 1 +
WP sunglasses -N-8000 - 2 +
DL ladies handbag -N-23000 - 1 +
Scan barcode
Checkout

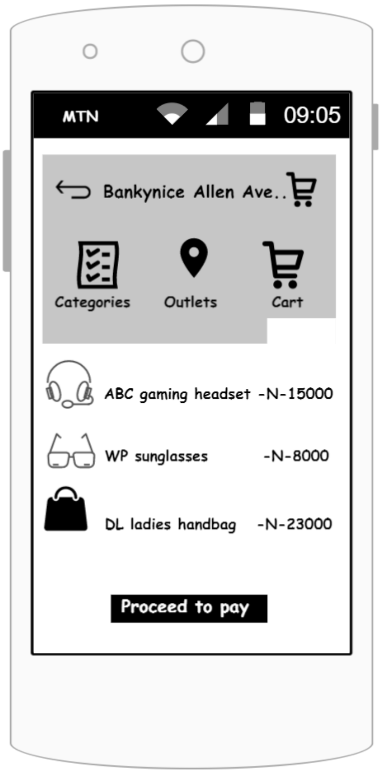
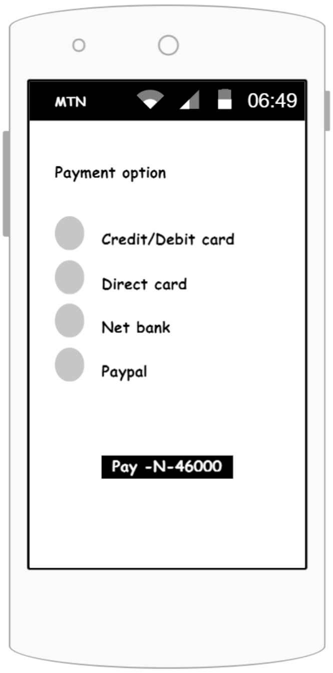


Figure 25. Virtual shopping cart I.

Figure 26. Virtual shopping cart II.

The next page gives the customer different options to make the payment with and a button totaling the price of all the products added to the virtual shopping cart as depicted in figure 27. A payment information page appears after clicking the “proceed to pay” button. In this page the saved payment card is displayed and an option to remove and replace the saved card if the customer so wishes as demonstrated in figure 28. The PayPal option directs the customer to PayPal where payment will be verified and redirected to the application with approval or denial. This applies to other payment merchants as well e.g. Visa, MasterCard etc. When the customer initially started shopping as guest a dedicated page meant for guess checkout appears with the relevant information need to process the payment as illustrated in figure 29.



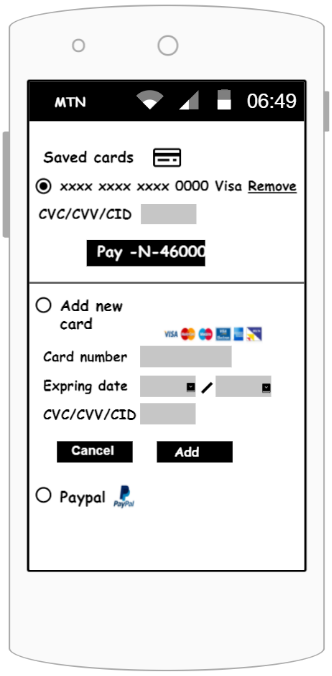
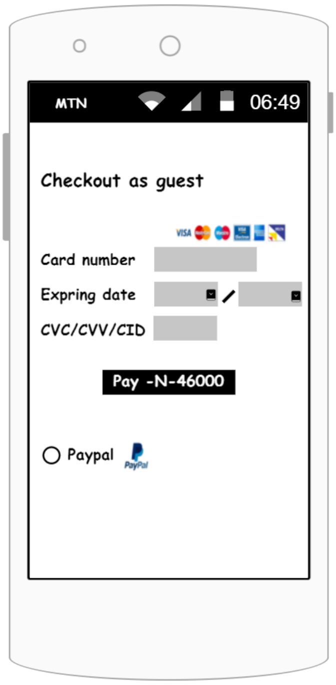


Figure 27. Payment option page.

Figure 28. Payment information page.



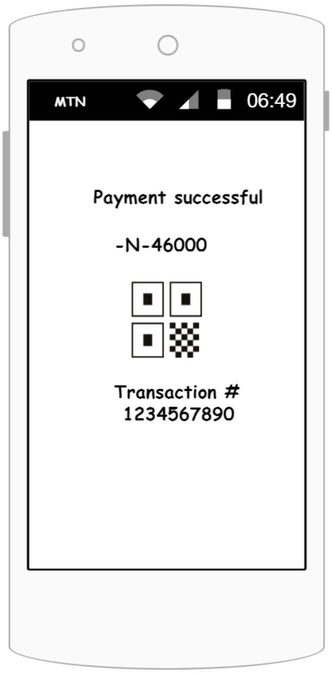


Figure 29. Guest checkout page

Figure 30. Receipt page

When the payment of a customer with an account or those shopping as a guest is verified, the receipt page appears displaying the amount paid, QR code and the transaction number as shown in figure 30.

5 Conclusions and recommendation

The aim of the thesis was to propose a functional model for a mobile self-checkout application that will be developed for a client's retail store. The features of the application were clearly expounded by highlighting the probable chain of actions that may occur between the key actors of the application i.e. virtual shopping cart, store, employee and the customer and the mobile self-checkout application.

The sequence of events owing to the interaction of an actor or collection of actors with the application were distinctly illustrated with UML diagrams. In some rare case where events did not happen as premeditated, reasons were given as well as helpful tips on how to remedy such occurrence.

This study also developed a user interface prototype to show the possible appearance of the mobile self-checkout application. A recommendation for the frontend, backend technology and implementation best suitable for the development of the application is given later in this section.

It is clear the application is going to be an immense asset to the client's business venture. Congestion currently experienced during peak period of sales at the cashier checkout point will be a thing of the past. The shopping process will be much faster, efficient and less strenuous for all parties involved. Moreover, from the store perspective, documentation and tasks e.g. store inventory, sales reports, product restock orders will be effortlessly implemented.

Considering that the mobile self-checkout application will be developed for a store, MySQL should be used to build the database system (server) at the backend application and stored in a remote location i.e. the cloud. MySQL is a free open source database that has Excel in key areas such as security, performance, query optimization and so forth. Undeniably, it suits the task at hand. The next step would be to develop an intermediary medium using PHP to serve as an interface between the mobile devices and the database system. They help retrieve data from the server and transfer to the mobile devices or store data into the database coming from a mobile device.

For the frontend application, to avert complexity during application development a hybrid application which is accessible to both IOS and Android platforms should be built at the frontend. This can be achieved by using photoshop to create PSD files are split into smaller components which will be in PNG format. ICONIC with HTML files inherent in it uses the created PNG components to design the UI of the application.

Finally, all the design elements should be incorporated into a code and functions programmed to come up as a working application using AngularJS. This programming language is preferable since a hybrid application is developed at the frontend application. If developed, the retail store will be able to significantly improve the payment process at the checkout.

References

1. Margaret Rouse, Don Vlack, Sue Huhn, E. Bud Gillaspie, Robert Kramer, “Self-scanning checkout (self-checkout)” [Online]. September 2005. Available from: http://searchcrm.techtarget.com/definition/self-scanning-checkout [Accessed 17 August 2017].

2. Matt Lake, “HOW IT WORKS; The Self-Checkout: Lots of Swiping, No Stealing” [Online]. 6 June, 2002. Available from: http://www.nytimes.com/2002/06/06/technology/how-it-works-the-self-checkout-lots-of-swiping-no-stealing.html [Accessed 21 August 2017].

3. Coursera.org. [online]. Available from: https://www.coursera.org/learn/client-needs-and-software-requirements/lecture/bZNCr/3-2-4-use-cases [Accessed 30 August 2017].

4. Javaengineeringprograms.com. [online]. Available from: http://www.javaengineeringprograms.com/activity-diagram-for-inventory-management-system/ [Accessed 5 September 2017].

5. Uml-diagrams.org. [online]. Available from: http://www.uml-diagrams.org/examples/point-of-sales-uml-use-case-diagram-example.html [Accessed 5 September 2017].

6. Quora.com. [online]. Available from: https://www.quora.com/What-are-some-reasons-products-get-discontinued [Accessed 9 November 2017].

7. Creately.com. [online]. Available from: https://creately.com/diagram/example/hm2e9u8a/Inventory%20System-Use%20Case [Accessed 9 November 2017].

8. Smashingmagazine.com. [online]. Available from: https://www.smashingmagazine.com/2013/03/designing-a-better-mobile-checkout-process/ [Accessed 9 November 2017].