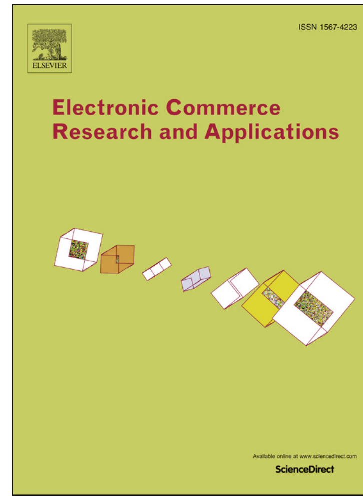
Accepted Manuscript



CE-GMS: A Cloud IoT-enabled grocery management system

Pankaj Deep Kaur, Jasleen Kaur

PII: S1567-4223(18)30008-5
DOI: https://doi.org/10.1016/j.elerap.2018.01.005
Reference: ELERAP 759

To appear in: *Electronic Commerce Research and Applications*

Received Date: 12 January 2018
Revised Date: 14 January 2018
Accepted Date: 14 January 2018

Please cite this article as: P.D. Kaur, J. Kaur, CE-GMS: A Cloud IoT-enabled grocery management system, *Electronic Commerce Research and Applications* (2018), doi: https://doi.org/10.1016/j.elerap.2018.01.005

CE-GMS: A CLOUD IOT-ENABLED GROCERY MANAGEMENT SYSTEM

Pankaj Deep Kaur (corresponding author) and Jasleen Kaur

Computer Science, GNDU Regional Campus
Jalandhar, India

Last revised: January 14, 2018

ABSTRACT

With the advent of the Fourth Industrial Revolution and cyber-physical systems, the multi-disciplinary area of *mechatronics* has become pervasive in "smart industry." The infusion of the *cloud IoT model* to the business processes has broaden the horizon for the grocery business. This study puts forward an innovative design of a retail business model and its implementation, using *sensor-based measurement containers* (SBMCs) and an Android application, which contributes new ideas for the international community. SBMCs containers are employed in the system to mitigate the pressure related to a *cloud IOT-enabled grocery management system* (CE-GMS) for the management of necessary household commodities, by sensing their quantities on hand using an ultrasonic sensor. It allows an alert to be generated when the item quantities reach the reorder point. The proposed CE-GMS consists of four automated entities: production units, warehouses, stores and customer sites – all of which are equipped with SBMCs of varying physical capacities. Cloud computing aids the systems to communicate with each other for better coordination. The automated information flow from a customer's SBMCs is handled by the nearest store. The data in the store's containers are received by the warehouses, and the data in the warehouse's SBMCs is sent to the production units. This communication among entities occurs with the support of GPS and GSM modules. The entity which receives information has the responsibility for supplying the necessary commodity products, by sensing the demand of the entity regarding commodity depletion. Further, the articles discusses the locking mechanism in SBMCs, inventory monitoring and container tracking for interaction in CE-GMS to validate the grocery market's operations using an IoT-compliant layer.

Keywords: Cloud IoT-enabled grocery management (CE-GMS); cloud IoT; cyber-physical systems; food commodities; Fourth Industrial Revolution; grocery industry; grocery reordering; Internet of Things (IoT); mechatronics; procurement; sensor-based measurement containers (SBMCs)

---

1. INTRODUCTION

The convergence of two technologies, cloud computing and the Internet of Things (IoT), has brought a revolution for smart industry by creating the basis for the Factory-of-Things and automated manufacturing units (Tramboulidis and Christoulakis 2016). Besides IoT and cloud computing, public Wi-Fi, 4G and 5G networks, and smartphones are the key elements of smart retail market (Obaidat and Nicopolitidis 2016).

The *smart grocery* approach leverages a sensor-based marketing model, which makes use of *sensor-based measurement containers* (SBMCs) that are placed at every subscribing site. The market is based on subscriptions, in which the customers are the kernel around which the whole system revolves, to provision them with the automated facilities of grocery purchases in their homes and delivery to their door steps.

The conditions for triggering the reorder quantity are detected by IoT-enabled sensors and actuators autonomously relay data input to the smart grocery system, which supports the functionality of a partially-automated retail grocery business process. The traditional retail system has been based on placing manual orders, which involves the challenge to the seller of winning the trust of the client. But in e-commerce, the sales growth entirely depends on the performance of the logistics. Supply chain management involves vigilance for consignment of commodities, the delivery at the destination address, and the return of products in case of an unexpected fault (Yu et al. 2017). The existing logistic system is automated to the extent that orders can be traced until they have reached their destination (Ganzha et al. 2016).

Sensor-based approaches have helped delivery services and shipping agents that use *global positioning* systems (GPS) to track the routes of moving products, including the commodity products (e.g., flour, oil, grain, etc., which we refer to as *commodities* in this article) sent to customers and the vehicles in which they are shipped. SBMCs represent an added component in the existing logistics process. The containers must be able to communicate with other nodes in a *cloud IOT-enabled grocery management system* (CE-GMS) in real-time to manage information and capital flows related to the commodities sent or returned.

Logistic managements in existing product delivery systems is based on the *asset-process-performance framework* (Subramanian et al. 2014). State-of-the-art smart logistics include the intelligent identification and location sensing of commodity products which are sent by e-grocers. They include GPS, networking, data processing, and billing based on the distance of the recipient and the weight of the commodity. In CE-GMS, the necessary data are related to commodities like the unique identification of each container. These can be checked by e-grocers, commodity product delivery service providers, and consumers. Other parameters include quantity, source address, destination address, product name, date-of-request, date-of-receipt, and so on. They are encapsulated in a single row of the logistics database in the cloud.

The objectives of CE-GMS are: (1) to reduce the overhead of shopping thereby aiding working professionals to manage homes and perform related house chores in less duration of time; to mitigate anxiety and the overhead of food shopping; and (3) to enhance existing the business model for groceries to a smart system. Section 2 present the cloud IoT approach, followed by the CE-GMS' characteristics in Section 3. Section 4 covers technologies for the containers in CE-GMS. Section 5 discusses the design of the system, while Section 6 discusses the physical paradigm associated with its operations. Section 7 lays out the approach used to implement the system, before we conclude

2. RELATED WORK

A systematic data-driven CE-GMS model contribute to the development of a smart communities. This model is composed of components that reduce the burden of household activities involving grocery purchases. We outline an archetype for an automated grocery market, a novel concept in the smart community movement. Earlier, in click-oriented e-commerce system, only items that are ordered online are supplied to customers. In this research, a new paradigm for an automated supply system is introduced, emphasizing the automatic supply of commodities for a particular site. The supplier automatically also receives a message if the quantity of any commodity is equal to or below the reorder point. This supports customized, and automatic purchase and supply, which mitigates the overhead of managing household activities by aiding in grocery purchases.

2.1. Cyber-Physical Systems and Information and Communication Technologies

Cyber-physical systems bring together physical systems and information and communication technology (ICT) (Segura et al. 2016). Cloud computing offers users ubiquitous resources via the network, using on-demand and pay-per-use access to a shared pool of configurable computing resources, such as networks, servers, storage, applications, and services (Mital et al. 2015). IoT networks, in turn, will eventually offer billions of nodes, each of which in principle will be uniquely addressable (Diaz et al. 201). The IPv6 standard offers a solution by providing a 128-bit address field, making it possible to assign a unique address to any possible node in the IoT network.

Other technologies are used to create a novel system of e-commerce, which makes use of IoT and cloud computing to satisfy the service demand of customers. In addition, based on service-oriented architecture (SOA) and service-oriented computing (SOC), resources such as platforms, software and infrastructure are allocated temporarily to clients who are subscribers (Chao 2015). SOA supports scalability and interoperability for different technologies using a four-layered architecture. It includes: a sensing layer to sense things, a network layer to gather data from hardware infrastructure and connect things together, a service layer to manage and create services for consumers, and an interface layer to handle interactions between users and applications (Buyya and Dastjerdi 2016, Khajenasiri et al. 2015). These services are released to the pool of resources when clients no longer need them.

Today, many industries are exploring how to use IoT to broaden its business horizons (Ju et al. 2016). The domain of e-commerce intertwined with the New Economy, which relies on information and communication rather than physical labor and natural resources (Ayden and Kavakliogl u 2011).

The transition to the Internet from a network of human-operated devices to non-human operated devices involves semi- and fully-automated systems. They use peer-to-peer and client-server based decision models for communication (Macaulay et al. 2016). Intelligent environments that use IoT are created through the use of various cloud computing capabilities: infrastructure-as-a-service (Iaas); platform-as-a-service (PaaS); and software-as-a-service (SaaS). The IoT devices have constraints in terms of battery life, processing power, and memory though (Bello and Zeadally 2014).

2.2. The Cloud IoT Paradigm and E-Commerce Business Models

The term *cloud IoT* – sometimes written as *cloudIT* to reflect their convergence (Kefalakis et al. 2016) comprises two technologies: cloud computing and the Internet of Things. cloud computing facilitates services for the consumer, with on-demand hardware and software resources, and the Internet of Things supports the identification and utilization of physical and virtual resources connected over the cloud. Massive data are generated by connected IoT devices to carry out well-defined communication (Karkouch et al. 2016).

In the cloud IoT model, data producers and consumers are viewed as smart objects (Karkouch et al. 2016). Due to technological advancement, voluminous data are reaped every moment in such environments. IoT technology is a catalyst for the proliferation of big data (Botta et al. 2016). Moreover, as the network of cloud IoT-enabled devices grows, traditional database management systems will become increasingly inefficient for storing and processing data. Recent technologies for higher storage capacity, faster processing and scalability have been emerging to handle these challenges (Diaz et al. 2016). Also, since data management is the epicenter of the IoT paradigm, it has become essential to manage data generated by IoT nodes, where security and the privacy of data streams are thmajor concerns (Mineraud et al. 2016). In sensor-based network systems, data security in terms of authenticity, integrity and confidentiality is essential, and requires strong trust among the various actors in the system (Grico et al. 2014). Further, sensitive IoT data are encrypted at the time of transmission, and later decrypted where it is stored (Jayaraman et al. 2017). Read/write permissions for data and access to appropriate techniques will aid in the handling of raw data sent from one network node to another (Mineraud et al. 2016).

The contemporary trend towards the utilization of online services and grocery purchases is changing (Elms et al. 2016). Earlier, the practice of purchasing online groceries was quite limited. The "death of distance" view triggered consumer interest in e-stores. But the stumbling block for its success in the online grocery market was logistics problems, including slow response times, limited product availability, ordering application interoperability, and so on. Today though, interoperability is ensured by SOA. Further, real-time *quality of service* (QoS) monitoring systems are now available to continuously check for *service-level agreement* (SLA) violations and ensure customer needs are satisfied (Alodib 2016). SLA agreements between parties involved function to mitigate the risks related to the exchange of resources (Petri et al. 2012, Lu et al. 2016). Any system without some QoS-aware capabilities will cause numerous customer orders to be delivered after the time the products should have been received, or food commodities in our case (Alodib 2016). As a result, it is necessary to include a QoS-awareness service component in a CE-GMS to ensure it will perform well.

2.3. Towards Smart Grocery Markets

Table 1 compares the various grocery market purchasing channel capabilities and their contrasts.

INSERT TABLE 1 ABOUT HERE

There are four market approaches covered: the traditional physical channel, the click-oriented e-commerce channel, the multi-channel approach, and the new CE-GMS approach. Traditional commerce employs offline marketing and the selling of products. It involves interpersonal relationships that must be developed so customers are convinced to purchase the products they need. Differentiation strategy is used in this approach, as Johansson and Kask (2017) have pointed out. They also characterize the approaches in the other channels and business models.

Click-oriented e-commerce is common these days, with Amazon, Flipkart, Snapdeal, e-Bay and so forth. Its business model utilizes an online approach for attracting customers with impersonal communications, but shares detailed information with consumers on the quality of the products and services via websites. The products that customers wish to purchase are clicked and supplied by e-grocers with the assistance of a delivery logistics partner. The business strategy is cost leadership. The multi-channel approach uses both online and offline markets, mixed personal and impersonal communication to achieve its results. Differentiation is emphasized in this retail format too.

Finally, the CE-GMS approach employs online and sensor-based approaches. Impersonal communication initiates the consumer relationship via a website, but a consumer's requests are serviced automatically later using sensor technology. The business strategy is differentiation.

2.6. Complications: Trust, RFID and UML

Another aspect that affects novel technologies is trust. Three factors are key: consumer, firm, and website infrastructure and interaction characteristics (Oliveira et al. 2017). There also are issues related to the costs for cloud computing-based smart services. Their success relies on the creation and deployment of cloud ecosystem capabilities through various stakeholder. They include: application service providers, cloud service providers, utility and mobile service providers, and community administration and payment gateways (Mital et al. 2016).

RFID offers many applications in the domains of retail and supply chain management (Gubbi et al. 2013). Three different kinds of RFID tags are: (1) *passive RFID*, employed in item-level tracking and supply chains; (2) *active RFID*, which possess their own battery supplies, and whose usage is in the domains of manufacturing, remote-sensing, IT asset management, etc.; and (3) *semi-passive RFID* tags (Lee and Lee 2015). Their main purpose is to locate or identify things, so the information can be shared over the Internet and data can be transferred related to what they sense (Fosso Wamba et al 2008). RFID technology is integrated with wireless devices and wireless web portals, and support e-commerce processes, such as customer service, order management, and real-time alert monitoring.

The use of a UML profile to represent system functionally for to the industrial Internet Things, involving component with IoT-like interfaces that are ready to be integrated into the modern IoT-equipped manufacturing environment. UML allows developers to generate IoT-compliant interfaces for cyber-physical components and automatedevelopment (Thramboulidid and Christoulakis 2016).

3. SYSTEM CHARACTERISTICS

The smart system we propose emphasizes the automatic management of inventory through the use of ultrasonic sensors, GPS and global system for mobile (GSM) modules in SBMCs using commodity interface devices. (See Figures 1 and 2.)

INSERT FIGURES 1 AND 2 ABOUT HERE

Up-to-the-minute information on a customer's stock is updated at the nearest CE-GMS store, which in turn responds by replenishing the commodity product items that are needed. The store-level SBMCs are connected to warehouses in the supply chain, and they are replenished by the production units. Automatic alerts for reorder are broadcast for the system to be informative and function well. Tracking elements are embedded in every physical container to monitor their location and identify them in the organization's CE-GMS.

Customers cannot add their own commodity product choices into the containers. Only the delivery staff can refill them, and to do so, they require authentication passwords that are known only to a store's staff. The consumption of the commodity product is monitored through a sensor located at the bottom of the container. When the commodity products are consumed by the customer and the remaining level matches the reorder point, an SMS alert is sent to a registered phone number of the retailer to supply the desired amount of the commodity. Similarly, when the commodity products have been depleted in stores or warehouses, a similar alert is sent to the warehouse or the production unit to automatically supply the required commodities.

5. DESIGN OF CE-GMS

5.1. UML representation of CE-GMS

Figure 3 represents seven classes in the system: Goods, Store, Customer, ProduceUnit, Supplier-Warehouse, Order and Container.

INSERT FIGURE 3 ABOUT HERE

Classes represent the blocks of code required to realize the CE-GMS model. The Customer class contains the following data members: customer_Id; customer_Name; customer_Address; payment-Mode, which may be cash on delivery or online transaction; and qtyLimit is intended to describe the Customers according to their consumption level.

The member function, addCust(), is used to add a new Customer to the business process, while any Customer who is inactive or wants to discontinue the service can be removed from the list using the removeCust(). The function detectQty() detects the available quantity for the container in which product items are stored. The reorder quantities in the Containers are sent to grocery Stores to automatically inform the staff to supply the required product items. The methods such as communicateS- tore() and requestStores() automatically shares essential information regarding the remaining quantity of products or any other related information to the Store class.

Suppose a Customer places an order, as illustrated in Figure 3. The dashed line with an arrow pointed towards the Store shows the dependency of the Customer entity on the Store entity. A hollow diamond and solid line indicates that the Customer has Goods in her sensor-based Container. Store contains storeId, storeLocation, Address, good_Id for the products allocated to the Store. Also good_Name and good_Qty incorporate the quantity of the product corresponding to good_Id, and good_Price. Further, lowQtyLimitStore indicates when there is a low quantity of the product items available in any Store. This results in an automated message that is sent to the Warehouse to supply essential items to a particular Store location.

The subroutines, detectLowQtyStore() and communicateSupplier(), use the attribute lowQtyLimitStore to pass message calls to the warehouse whereas the procedure detectLowQtyCustomer() and communicateCustomer() acts on the low quantity of items at the consumer end. Moreover, the class Store depends upon the class SupplierWareHouse, and it illustrates the relationship in aggregate with the Container and Goods classes. It also depicts the association with Order class so as to support the placing of orders.

The SupplierWareHouse entity includes supplier Id, denoted by s_Id, s_Name, s_Address, qtyOrder, order_Id, good_Id, good_Name, good_Qty, goodPrice, date, amountRemaining, and lowQtyWareHouse. The latter contains the reorder commodity quantity which, when reached, sends a buzzer message to the ProduceUnit(). The methods which implement the essential functionality of the SupplierWareHouse class are addSupplier(), to add new suppliers into the list, and removeSupplier() to remove suppliers in case the services of any supplier are terminated with respect to all the stores. updateSupplier() updated the suppliers in order to allocate them to different stores or to assign new information about them. detectLowQtyStore() and communicateStoreContainers() contain instructions to initially detect the present quantities in the Containers of a Store, and if there is an indication of a low quantity coming close to the reorder limit, then a message will be generated to communicate this to the ProduceUnit.

In this scenario, the SupplierWareHouse entity captures its dependence on ProduceUnit for its supply of products. The aggregation relationship of the SupplierWareHouse class with the Goods and Container classes shows that SupplierWareHouse has goods and Containers available to store commodity products. The association with the Order class suggests that the SupplierWareHouse can place orders. The ProduceUnit class has attributes, including: good_Id, good_Name, producedQty, suppliedQty, s_Id, s_Address, date and productionQtyLowLimit. Procedures such as communicateWarehouse() and qtyDetectWarehouse() detect the quantity of the commodity product available at a Warehouse and communicate about it as needed. The methods qtyDetectionPU(), Produce() and Update() query the ProduceUnit to determine the amount of a commodity that is available. Otherwise a message will be generated to inform the ProduceUnit to make the commodity product. The ProduceUnit has a 'Has A' relationship with the Goods and Container classes.

5.2. Information Flows in CE-GMS

Figure 4 represents the information flows in sequential order.

INSERT FIGURE 4 ABOUT HERE

The figure shows that the control flow is initiated by a warehouse with index 1, with connections via subscriptions to the Production Units. Immediately upon the receipt of a new subscription, SBMCs are allocated to the Warehouse's Containers to store the products. The low quantity limit for reordering is fed into the SBMCs which generate alert messages for the Production Unit to supply additional items to a particular Warehouse. Later, a payment is made on receipt of the order.

Stores also goes through similar steps at another level. Initially, Stores subscribe to warehouses, which in turn allocate Containers to the Stores for several different items. A minimum reorder quantity is sent to the Containers. Just above that quantity, the system will generate alert messages for the warehouse to supply goods again, and eventually, payments will be made.

Customers experience all of the phases (Subscription, SBMC allocation, SBMC alerts, Supply and Payments), as depicted in Figure 4. A repository includes a database to keep a record of all of the transactions. Mutual communication among the entities (Production Unit, Warehouse, Store and Customer) are also stored in the database as an audit trail for recovery.

5.3. A Representative Example

An example of CE-GMS is offered to enhance the reader's understanding. Customers may have commodity products such as oil, flour, grain and so on, in SBMCs with defined reorder points, say, 2 units. When the reorder point is reached, an SBMC deliver a message to the Store to replenish the needed commodity product for the home without intervention by the Customer. This eliminates the need to visit an e-commerce website to order resupply quantity. Meanwhile, a Stores' products are detected and managed by Warehouse suppliers. The Warehouse is maintained by the Production Unit with the help of the SBMC and cloud IoT network. The cloud IoT network is responsible for communication and maintaining security on behalf of the various entities.

In addition, Stores receive the commodity products from Warehouses. The Stores are allocated bigger SBMCs according to their need to store the commodity products and will inform the Warehouses about the remaining quantities they have on hand. According to the sales of various goods in Stores, reorder limits are assigned to SBMCs by a Store staff. As with Customers, who benefit from automatic replenishment of their commodity products from Stores, Stores also will be automatically resupplied with commodity products from the Warehouses. And, automated Warehouses will resupply Stores located at different places as well.

Another tier of the system involves the Warehouses, which possess bigger SBMCs than those at the Stores to store commodity products. Warehouses receive data feeds on the reorder points when there are low quantities in the SBMCs for the various commodity products. The Production Unit detects the Containers assigned to the various suppliers and their own Containers. When SBMCs detect threshold quantities at the Warehouse, the Production Unit will become active to inform its staff via their mobile phones through a cloud-based Android app to ship certain commodity products to the Warehouses or to commence their production, if a Production Unit has reached the threshold quantity of the commodity. Thus, SBMC plays a vital role in the automation of the grocery management system, and Production Units produce commodity products based on Customer demand.

6. THE PHYSICAL PARADIGM OF CE-GMS

Figure 5 represents the various CE-GMS market components which are connected with the use of cloud IoT technology. Communication takes place via the different cloud services – IaaS, Paas, and SaaS. Data travel from one node to another through networking. Numerous Production Units, Warehouses, Stores and Customers are able to subscribe in this service model.

In Figure 6, a set of containers belonging to various Customers [C11, C12, …, C1n] in a particular region send requests to Store S1, which in turn receives commodity products from different Warehouses [W11, W12, …, W1n]. The Warehouses obtain freshly-produced commodities from various Production Units [P11, P12, …, P1n]. The containers belonging to another set of Customers of another region [C21, C22, …, C2n] and [Cn1, Cn2, …, Cnn] send threshold quantity alarms to Store entities S2 and Sn. The Store entities collect the required commodities from the Warehouses [W21, W22, …,W2n] and [Wn1, Wn2, …, Wnn] for replenishment. In addition, the Warehouses set [W21, W22, …,W2n] and [Wn1, Wn2, …,Wnn] to obtain their commodities from the Production Units [P21, P22, …, P2n] and [Pn1, Pn2, …, Pnn].

7. IMPLEMENTATION OF CE-GMS

7.1. Design of the Mobile Application

The four entities (Customer, Store, Warehouse and Production Unit) use four separate user- specific interfaces with shared databases. The foremost task of the Customers, the Stores, the Warehouses and the Production Units is to subscribe to the Android application of CE-GMS to participate in the automated retailing system. The attributes of ***Registration*** (Activity 1) are: *Name, e-mail, Password, Confirm Password, Mobile, Address and Select Nearest Store/Warehouse/Production Unit*, as sensed via GPS (Activity 2) for those who want to subscribe. Another interface of Android is its ***Login*** from which the Customers, Stores, Warehouses and Production Units can enter the system.

Customers, Stores, or Warehouse can only make manual orders, as with Amazon, Flipkart, Snapdeal, and Myntra, the e-commerce giants. The ***Manual Order*** activity includes *Container Type, Item Order, Item Quantity* and *Reorder Level*. The Customer can place an order by specifying its category, like a 5kg container, or the commodity product by mentioning the product quantity to a subscribed store. Stores can maintain the records of SBMCs. There is a section called ***Automated Container Management***, which contains *Container ID, Container Type* and *Reorder Count*, with buttons having the functionality to add more containers, delete defective ones, and update and view container details (Activity 4). Additionally, it is programmed to count the total number of containers available and the reorder quantity threshold of the containers, such that when the total available container quantity matches the reorder quantity, an SMS alert will be generated and sent to the corresponding Warehouse or Production Unit to supply the required containers at a specific site, place orders to subscribed Warehouses, and Warehouses can place orders to subscribed Production Units.

The ***Manual Order Reception*** activity displays the first or manual order of the Customer, Store or Warehouse to the Store, Warehouse or Production Unit. The attributes of this activity are: *Name, e-mail ID, Mobile, Address, Container Type, Container ID, Item Name, Quantity* and *Reorder quantity.*

When the containers are reached at the sites where the CE-GMS is implemented, the automated process gets started. The attributes of the ***Automated Response*** (Activity III) are: *Name, e-mail, phone, Address, Container Type, Container ID, Item Name, Quantity Available* (through the ultrasonic sensor)*, Quantity to Deliver, Reorder Level,* and *Tracked Container Location* (via GPS). Now, when the available quantity of the container equals the reorder quantity (Activity IV), an SMS alert is generated to the Store (for the Customer, to the Warehouse (for a Stores) or to a Production Unit (for a Warehouse) to inform them that a particular Customer requires a specific commodity product to be delivered. Afterwards, the Store, Warehouse, or Production Unit will confirm if the Customer, Store, or Warehouse, respectively, want to receive the item as the alert suggests.

The *Goods* record is stored in the repository. ***Automated Goods Management*** (Activity 5) has: *Goods Name, Quantity and Reorder Level*. The Store, Warehouses and Production Units can add, update, delete, or view goods according to what is required in order to manage their stocks. While supplying goods to customers, when the quantity of some commodity product becomes equal to the reorder quantity, an automated SMS alert is sent to the Warehouse so it can replenish the commodity based on the request of the Store. This is similar for Warehouses as well: when a commodity product is depleted in a Warehouse, an auto-alert will be sent to a Production Unit. Then it will be responsible for sending the commodity product to the Warehouses with quantity available equal to the reorder quantity. This gives the new system its automatic capability to serve the Store. In addition, the Stores automatically recognize the needs of their customers who have subscribed, by sending them the required commodity at the right time. Warehouses never let the commodity products at the Stores become depleted, because the Production Units automatically make deliveries when any commodity product quantity is equal to the threshold reorder level.

7.2. Database Schema

The web server database contains four separate databases for four entities of this system. The Customer database contains these tables: **CustRegister** (CustID, CustName, CustEmail, CustPasswd, CustMobile, CustAddress, NearestStore), **CustLogin** (CustEmail, CustPasswd), and **CustManualOrder** (CustContId, CustContType, CustItemName, CustItemQty, CustReOrderQty).

The Store database has: **StoreRegister** (StoreID, StoreName, StoreEmail, StorePasswd, Store-

Mobile, StoreAddress, NearestWareHouse), **StoreLogin** (StoreEmail, StorePasswd), **StoreManualOrder** (StoreContId, StoreContType, StoreItemName, StoreItemQty, StoreReOrderQty), and **AutomatedReceiveRqstFrmCust** (CustGpsLoc, CustLevelSenseQty), **StoreContainers** (ContId, ContType, ReorderCount).

The Warehouse database includes: **WarehouseRegister** (WareID, WareName, WareEmail, WarePasswd, WareMobile, WareAddress, NearestProdUnit), **WarehouseLogin** (WareEmail, WarePasswd), **WarehouseManualOrder** (WareContId, WareContType, WareItemName, WareItemQty, WareReOrderQty), **AutomatedReceiveRqstFrmStore** (StoreGpsLoc, StoreLevelSenseQty), and **WarehouseContainers** (ContId, ContType, ReorderCount).

Finally, the Production Unit database covers: **ProdUnitRegister** (ProdID, ProdName, ProdEmail, ProdPasswd, ProdMobile, ProdAddress), **ProdUnitLogin** (ProdEmail, ProdPasswd), **AutomatedReceiveRqstFrmWarehouse** (WareGpsLoc, WareLevelSenseQty), and **ProdUnitContainers** (ContId, ContType, ReorderCount).

7.3. Case Study

The implementation of a trial smart grocery system was carried out at a city store with 50 customers who had 5kg capacity SBMCs for storing flour. The local Store was furnished with a container of 40kg capacity, which also contained flour. Up-to-the-minute information was acquired by the Store's staff trained to use the mobile app. Whenever the automatic request alert was received by the staff that an SBMC with a particular ID (e.g., COT1266558BT) reached its reorder point, a call was made to the customers to verify if the commodity product needed to be sent to them.

In the next step, the Store was connected with a Warehouse, whose role was to replenish the Store's SBMC with flour when it reached reorder point. The Warehouse has containers of 100kg capacity. After some days, when the commodity product was sold to Customers and the available quantity reached the reorder level, then an alert was sent automatically to the Warehouse, which then called the Store to confirm if it required the commodity product to be delivered. After this confirmation occurs, consignment of the commodity is completed by the Warehouse.

The Warehouse sent an alert to a Production Unit, indicated its SBMC had a 150kg capacity. When the quantity of a commodity product reached the threshold quantity at a Warehouse, the Production Unit started producing more flour.

7.4. Results

After some time to use the system, customers were requested to send feedback on their satisfaction level. A feedback activity was appended to the mobile app with three categories: accuracy, usability and satisfaction ratings. The latter was used by customers to report their satisfaction levels. The study was implemented for three types of customers:

- Case I: Customers in families with two working parents
- Case II: Customers in extended families with working parents and retired grandparents

- Case III: Customers with one parent working

Case I customers. We observed that 75% of the customers rated accuracy with four stars, 13% it 3 stars for accuracy, and 12% rated accuracy with 2 stars. Moreover, 80% of customers valued usability at 4 stars, 12% with 3 stars and 8% with 2 stars. Additionally, 82% of customers rated convenience with 5 stars, 8% with 4 stars, 5% with 3 stars, 3% with 2 stars, and 2% with 1 star.

Case II customers. In contrast, this group reported different results. 60% of customers rated accuracy with 3 stars, 13% with 2 stars, and 27% with only 1 star. Additionally, 58% of customers rated usability with 3 stars, 24% with 2 stars, and 18% with 1 star. Moreover, 17% of customers valued convenience with 5 stars, 12% with 4 stars, 52% with 3 stars, 11% with 2 stars, and 8% with 1 star only.

Case III customers. Among the consumer who replied in this group, 13% of customers with one parent working rated accuracy with 3 stars, 17% with 2 stars, and 70% with 1 star. Additionally, 23% of the customers rated usability with 3 stars, 44% with 2 stars, and 33% with 1 star. In addition, 8% of the customers rated convenience with 3 stars, 48% with 2 stars, and 44% with 1 star.

Thus, we infer that this system was acceptable for families with working parents, with the least interests shown by customers with only one working parent. Either way, this kind of household food procurement support will be a valuable service for the contemporary world.

8. CONCLUSION AND FUTURE WORK

In this study, we designed a smart business model – a cloud IoT-enabled grocery management system (CE-GMS) – which automatically senses the household grocery needs of customers for commodity products such as flour and oil, among other things. This is an extension of the existing click-oriented e-commerce business model. CE-GMS integrates the functionality of cloud computing and the Internet of Things to implement a mobile application which processes data from sensors to identify when it is appropriate for reordering to occur. We evaluated this system in a grocery store with customers, and found that made grocery procurement easier for the family, especially working couples, and gave assistance for the management of kitchen commodities. In future research, we will work to enhance system interoperability and the networking of SBMCs in CE-GMS. Interoperability research will focus on security issues and security implementation, to enhance trust in the business model, so more customers will be willing to participate.

REFERENCES

Alodib, M. 2016. QoS-Aware approach to monitor violations of SLAs in the IoT. Journal of Innovation in Digital Ecosytems, 3, 2, 197-207.

Aydin, E., Kavaklioglu, S. 2011. A study of superiority of e-trade compared to traditional methods of commerce in overcoming crises: A case study of Kitapix.com. Procedia - Social and Behavioral Science, 24, 123–137.

Bello, O., Zeadally, S. 2014. Intelligent device-to-device communication in the Internet of Things. IEEE Intelligent Systems, 10, 3, 1172-1182.

Botta, A., De Donato, W., Persico, V., Pescapé, A. 2016. Integration of cloud computing and Internet of Things : A survey. Future Generation Computer Systems, 56, 684–700.

Buyya, R., Dastjerdi, A.V. (eds.) 2016. Internet of Things: Principles and Paradigms. Morgan Kauffmann, Burlington, MA.

Chao, K. 2015. E-services in e-business engineering, Electronic Commerce Research and Appl., 14, 3, 145-149.

Díaz, M., Martín, C., Rubio, B. 2016. State-of-the-art, challenges, and open issues in the integration of Internet of Things and cloud computing. Journal of Network and Computer Applications, 67, 99–117.

Elms, J., de Kervenoael, R., Hallsworth, A. 2016. Internet or store? An ethnographic study of consumers' Internet and store-based grocery shopping practices, Journal of Retailing and Consumer Services, 2, 234–243.

Fosso Wamba, S., Lefebvre, L.A., Bendavid, Y., Lefebvre, É. 2008. Exploring the impact of RFID technology and the EPC network on mobile B2B eCommerce: A case study in the retail industry. International Journal of Production Economics, 112, 2, 614–629.

Ganzha, M. Paprzycki, W. Pawlowski, P. Szmeja, K. Wasielewska, 2017. Semantic interoperability in the Internet of Things: An overview from the INTER-IoT perspective. J. Network and Computer App., 81, 111-124.

Grieco, L.A., Rizzo, A., Colucci, S., Sicari, S., Piro, G., Di Paollo, D., Boggia, G. 2014. IoT-aided robotics applications: Technological implications, target domains and open issues. Comp. and Comm., 54, 1, 32-47.

Gubbi, J., Buyya, R., Marusic, S., Palaniswami, M. 2013. Internet of Things (IoT): A vision, architectural elements, and future directions, Future Generation Computer Systems, 29, 7, 1645–1660.

Jayaraman, P.P., Yang, X., Yavari, S., Georgakopoulos, D., Yi, X. 2017. Privacy preserving Internet of Things: From privacy techniques to a blueprint architecture and efficient implementation. Future Generation Computer Systems, 76, 540-549.

Johansson, T., Kask, J. 2017. Configurations of business strategy and marketing channels for e-commerce and traditional retail formats: A qualitative comparison analysis in sporting goods retailing, Journal of Retailing and Consumer Services, 34, 326–333.

Ju, J., Kim, M., Ahn, J. 2016. Prototyping business models for IoT service. IT and Quant. Mgmt., 91, 882–890.

Kabanda, S., Brown, I. 2017. A structuration analysis of small and medium enterprise (SME) adoption of e-commerce: The case of Tanzania. Telematics and Informatics, 34, 4, 118-132.

Karkouch, A., Mousannif, H., Al Moatassime, H., Noel, T. 2016. Data quality in Internet of Things: A state-of-the-art survey. Journal of Network and Computer Applications, 73, 57–81.

Kefalakis, N., Petris, S., Georgoulis, C., Soldatos, J. 2016. Open source semantic web infrastructure for managing IoT resources in the cloud. Chapter 2 in R. Buyya and A.V. Dastjerdi (eds.), Internet of Things: Principles and Paradigms. Morgan Kauffmann, Burlington, MA. pp. 29-48.

Khajenasiri, I., Estebsari, A., Verhelst, M., Gielen, G. 2017. A review on Internet of Things solutions for intelligent energy control in buildings for smart city applications. Energy Procedia, 111, 770–779.

Lee, I., Lee, K. 2015. The Internet of Things (IoT): Applications, investments, and challenges for enterprises. Business Horizons, 58, 4, 431–440.

Lu, K., Yahyapour, R., Wieder, P., Yaqub, E., Abdullah, M., Schloer, B., Kotsokalis, C. 2016. Fault-tolerant service level agreement lifecycle management in clouds using actor system. Future Generation Computer Systems, 54, 247–259.

Macaulay, T., Schneck, P., Brown, G. 2016. RIOT Control: Understanding and Managing Risks and the Internet of Things. Elsevier Science & Technology Books, Amsterdam, Netherlands.

Mineraud, J., Mazhelis, O., Su, X., Tarkoma, S. 2016. A gap analysis of Internet-of-Things platforms. Computers and Communication, 89–90, 5–16.

Mital, M., Pani, A.K., Damodaran, S., Ramesh, R. 2015. Cloud based management and control system for smart communities: A practical case study. Computers in Industry, 74, 162-172.

Obaidat, M.S., Nicopolitidis, P. 2016. Smart Cities and Homes: Key Enabling Technologies. Morgan Kauffmann, Burlington, MA.

Oliveira, T., Alhinho, M., Rita, P., Dhillon, G. 2017.Modelling and testing consumer trust dimensions in e-commerce. Computers and Human Behavior, 26, 5, 857-869.

Petri, I., Rana, O.F., Silaghi, G.C. 2012. Service level agreement as a complementary currency in peer-to-peer markets. Future Generation Computer Systems, 28, 8, 1316–1327.

Segura, D.M., Kaur, N., Whittow, W.G., Conway, P.P., West, A.A. 2016. Robotics and computer-integrated manufacturing towards industrial Internet of Things: Crankshaft monitoring, traceability and tracking using RFID. Robotics and Computer-Integrated Manufacturing, 41, 66–77.

Subramanian, N., Ning, K., Edwards, D. 2014. Product delivery service provider selection and customer satisfaction in the era of Internet of Things: A Chinese e-retailers' perspective. International Journal of Production Economics, 159, 104-116.

Thramboulidis, K., Christoulakis, F. 2016. UML4IoT-A UML-based approach to exploit IoT in cyber-physical manufacturing systems, Computers in Industry, 82, C, 259–272.

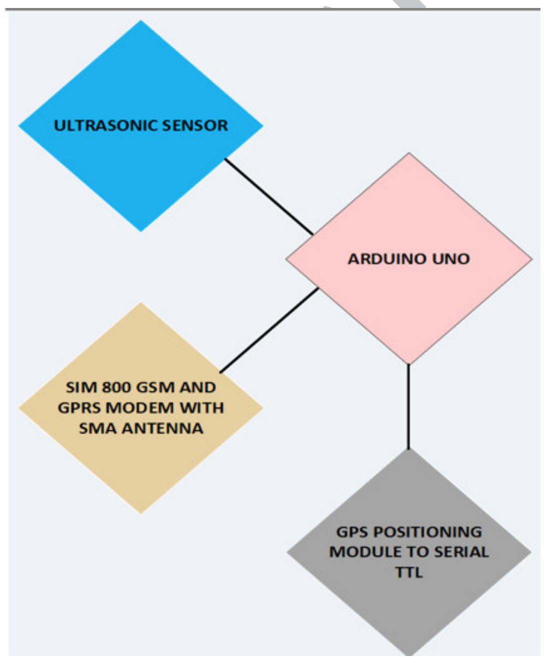
Yu, Y., Wang, X., Zhong, R.Y., Huang, G.Q. 2016. E-commerce logistics in supply chain management: Practice perspective, Procedia CIRP, 52, 179–185.

Table 1. Comparison of traditional, Internet based, multichannel and fully automated GMS

Channels	Retail Format	Marketing Approach	Strategy
Traditional	Offline	Involves individual interactions with harmonious overtones to attract customers and establish short- or long-term relationships	Differentiation
Clicks	Online	Impersonal mode of communication takes place in this category which involves rhetoric publishing on the web	Cost leadership
Multi-channel	Online, offline	Use. impersonal and interpersonal communication	Differentiation
Cloud IoT	Online, sensors	Initial impersonal interaction at time of subscription, then requests and responses become eautomatic with sensors use	Differentiation

Note: Refer to Johansson and Kask (2017) on the contents of this table.

Figure 1. The Sensor-Based Measurement Container (SBMC)



Notes: The container tracker and reordering process requires inexpensive commodity interfaces, such as the Arduino Uno products noted above. The other components ensure GPS tracking and GSM telecommunication, as well as reorder quantities for reorder point-driven requests.

Figure 2. Technologies Used in the Containers

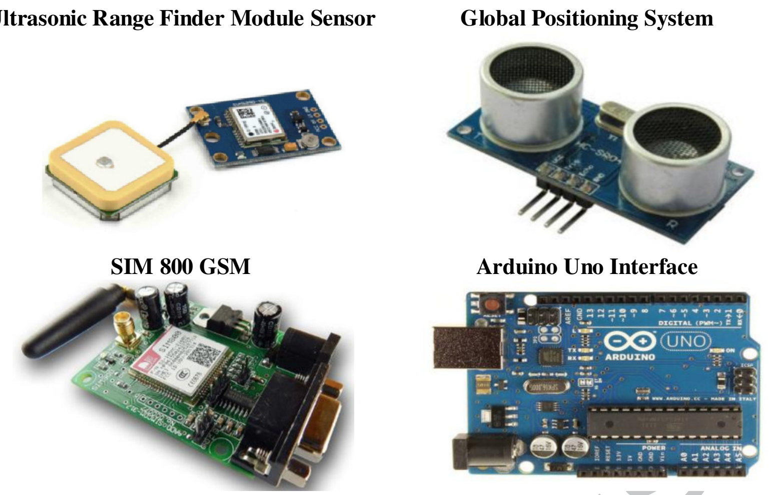


Figure 3. Sys-ML Representation of CE-GMS

Figure 4. Information Flows in CE-GMS

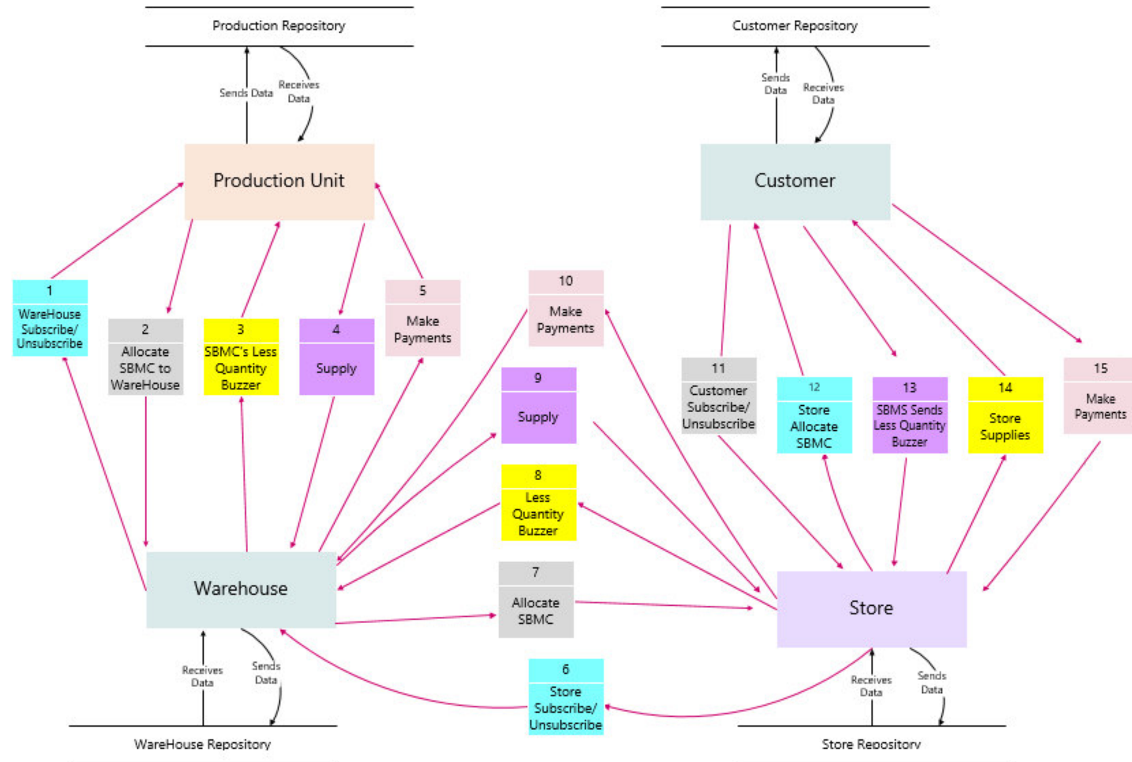


Figure 5. Various Units in the Cloud IoT-Enabled Grocery Management System Components

Figure 6. Extended Cloud IoT-Enabled Grocery Management System

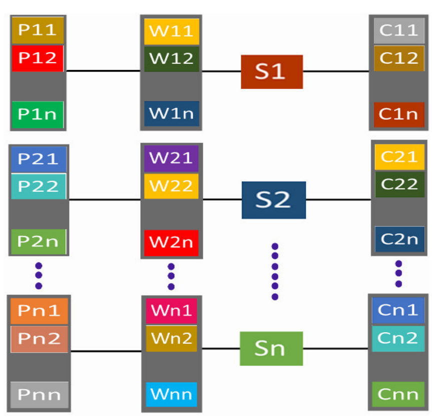


Figure 7. The Design of the Mobile Application

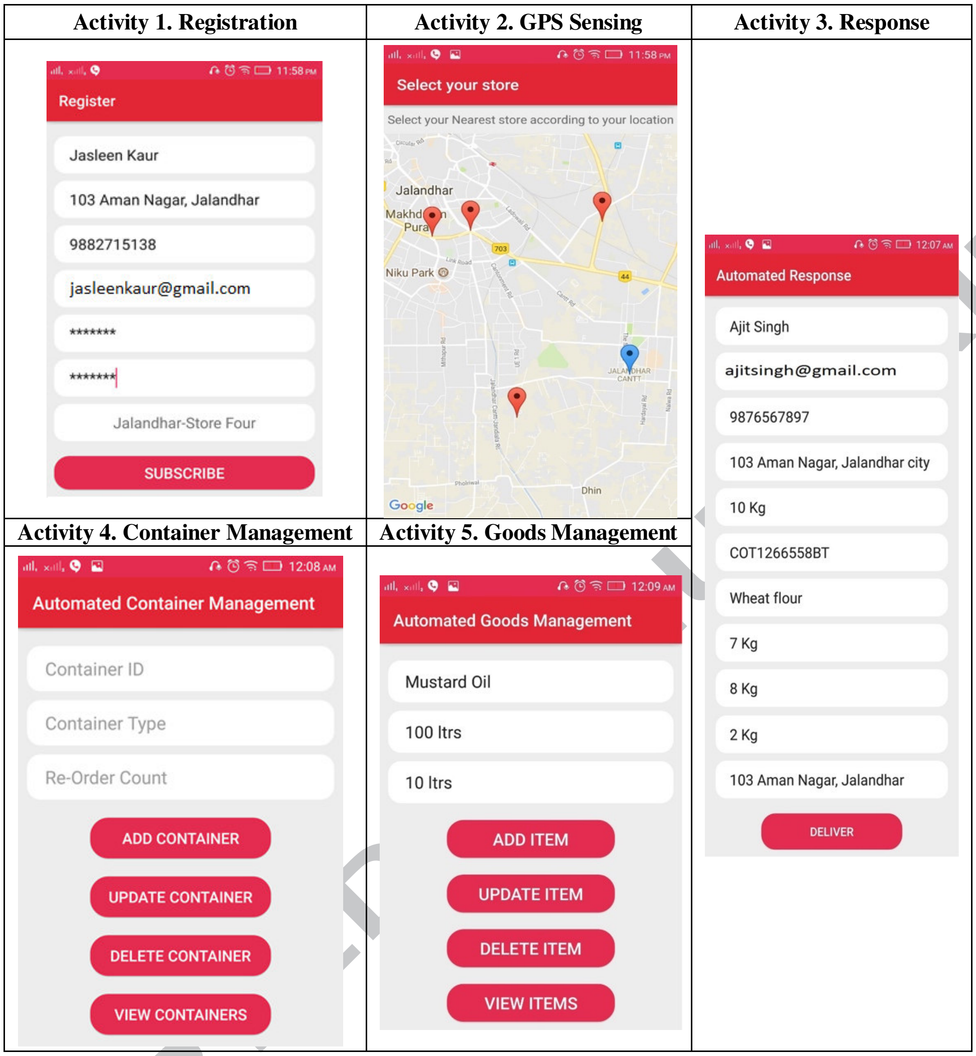


Figure 8. Accuracy

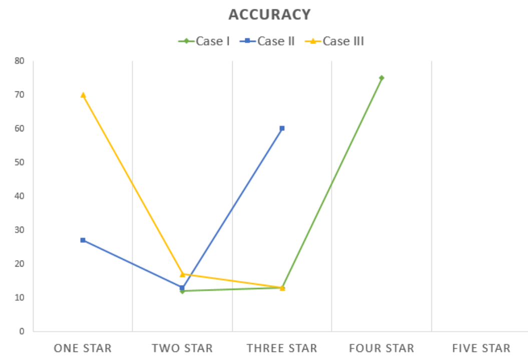


Figure 9. Usability

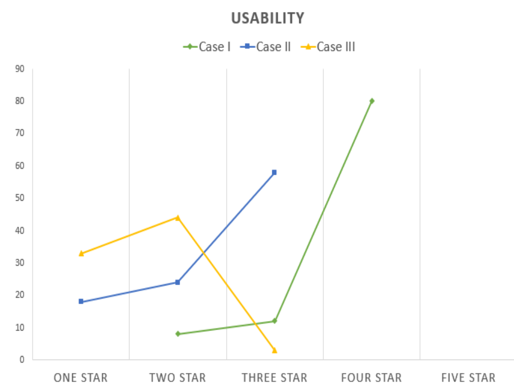
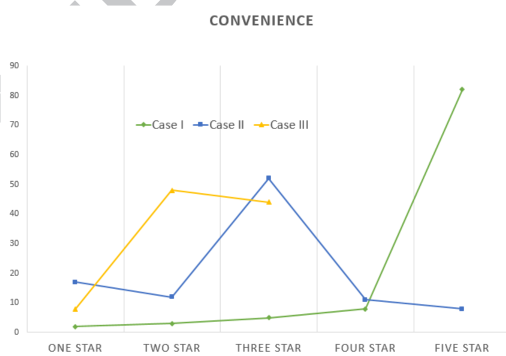


Figure 10. Convenience



Highlights

Cloud IoT-enabled grocery management system (CE-GMS) model is presented

Sensor-based measurement containers (SBMCs) manage household commodities by sensing the quantity of commodity products.

A smart system can manage inventory automatically by the use of an ultrasonic sensor, and GPS and GSM modules in the SBMCs.

Reorder quantities for commodity food products (flour, rice, oil, etc.) can be tracked and reorder requests made

CE-GMS was built using an Android-based mobile app

A case study was performed to evaluate the system in terms of accuracy, usability and customer satisfaction