

# Use-Case, Runtime and Deployment Views for Developing Subsidies Mobile Wallet

Dina Darwish

Assistant Professor, International Academy for Engineering and Media Science - Egypt  
e-mail: gdarwish@mcit.gov.eg

## ABSTRACT

*Subsidizing is usually controlled and provided by governments and NGOs in many countries, and is considered as a strategic service. Subsidizing is important because it provides a means for needy customers to obtain certain services, they cannot afford paying their original prices. The revolution in the technology used in mobile phones and the wide use of these mobile phones, have led to the idea of distributing subsidies using mobile phones. Solutions were proposed for a flexible **Subsidies Mobile Wallet (SMW)** through mobile phones and wireless networks to satisfy the technical requirements necessary for enhancing the subsidies distribution. This paper proposes complete design solution for the **Subsidies Mobile Wallet (SMW)** in terms of Use-Case View, Runtime View, and Deployment View. The discussion of the **Subsidies Mobile Wallet (SMW)** is illustrated by examples to show how the system can be used in the real world.*

**Keywords:** Mobile Wallet, Runtime View, Deployment View, Mobile technology, Subsidies.

## 1. Introduction

The number of people suffering from hunger and malnutrition all around the world increases every year, and this poses a threat to all the countries. Because fighting poverty is an important tool to fight other problems, such as: illiteracy, crimes, and diseases. The current situation proves that all efforts done to reduce poverty and provide food security are not enough. The international and national organizations are aware of food crisis, and have done a lot of efforts and provided a lot of food aid[1-4] to a huge number of people worldwide, but the level of action is quite far from meeting the needs of the increasing numbers of hungry people.

Subsidizing is considered as a strategic tool adopted by governments and NGOs to fight poverty. Food subsidies try to minimize the poverty, if they are distributed correctly to the people in need. The distribution of subsidies poses a problem for all governments to ensure food delivery with equity, and to avoid theft and bad distribution of food aid. Subsidies are classified in many different ways according to the reason behind them, the recipients of the subsidy, the means of distributing the subsidy, and the source of the funds (government, NGO's, general tax revenues, etc). [5-10].

Some economic analysts argue that direct subsidies (cash benefits) are more beneficiary than indirect subsidies (such as trade barriers). Because direct subsidies ensure the delivery of subsidy to people considered needy directly by governments, but indirect subsidies does not mean that the benefit would reach the intended people. However, the choice to enact a subsidy is a political choice taken by governments. Distribution of subsidies constitutes a nightmare to governments and international organizations. There are many challenges facing the delivery of subsidies, such as: preventing corruption in cash transfers, addressing people of wide range of education, age, capabilities, skills, and constraints, finding innovative delivery mechanisms in conjunction with private sector partners, delivering the subsidy to the intended beneficiaries many of whom may live in remote areas [11-14].

Due to the wide spread of the use of Information and Communications Technologies (ICTs), bringing new technology in the delivery of food cash has been a crucial idea. There are many new technologies to determine the identity beneficiaries and to secure transfers of food cash to remote areas, such as: fingerprint recognition, smartcards, cell phones and 3G networks, and so

International Journal of Computer Science & Information Technology (IJCSIT), Vol 3, No 1, Feb 2011 on. To ensure correct beneficiary identification. Taking the emerging mobile technology into consideration, a new system architecture is developed [15], that is mobile-based, to solve the problem of delivering food subsidies in a more efficient way, and at the same time, to overcome previous problems resulted from the old subsidies distribution methods. The proposed system architecture is called subsidies mobile wallet (SMW). The Subsidies Mobile Wallet (SMW) is described in the previous paper in details from the Logical View, Architectural Design, and the Process View. In this paper, the SMW system design is proposed in terms of the Use-Case, Runtime, and Deployment Views.

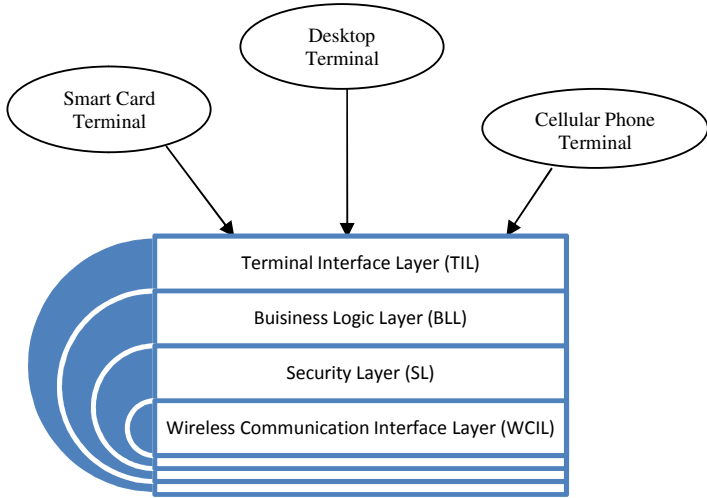
The rest of this paper is organized as follows. In Section 2, the developed subsidies mobile wallet (SMW) architecture is described briefly. Section 3 gives a detailed description of the System Use-Case View operations. Section 4 gives a detailed description of the System Runtime View operations. Section 5 gives a description of the System Deployment View. Finally, conclusion for this paper is given in section 6.

**2. Proposed Subsidies Mobile Wallet (SMW) Architecture**

In [15] high-level operational view of the SMW system. Was developed as shown figure 1, where the system consists of four entities interconnected via mobile network infrastructure. The four entities are the client (customer), the merchant, the third-party entities (subsidizing entities), and the servers (hosted at relevant stakeholders’ sites). The scope of this section is to present a proposed architecture for the design of both customer and merchant devices, which are similar from the *architectural perspective*.

The proposed architecture of the SMW [15-19] solution presents the structure of the system that comprises software components, the externally visible properties of those components, and the relationships among them. The proposed architecture adopts the well-known layered architectural style in order to ensure flexibility and modularity. Generally speaking, the basic structure of a layered architecture consists of a number of different layers, each accomplishing operations that progressively become closer to the low level physical operations. At the outer layer, components service user interface operations. At the lower layer, components perform physical system operations. Intermediate layers provide utility services and application software functions.

The summary of business operations to be performed by the SMW in the developed SMW architecture are grouped into four main categorizes:



**Fig. 1.** The proposed layered architecture for SMW

- *User Account Management operations:* These operations deal with the registration of the customers and merchants to the SMW system. The operations also cover various activities related to the management of the created account, such as the change of the user PIN.
- *Subsidize-Out operations:* These operations present the core of the SMW solution. They cover the activity of claiming subsidized items from a merchant. The Subsidize-Out operations include the customer request, merchant approval, and customer approval processes.
- *Management and Control operations:* these operations focus on the activities related to the supply of subsidized items to the merchants and monitoring of the status of these items throughout their life-cycle from their arrival to the merchant until they are claimed by consumers.
- *Enquiry-related operations:* these operations cover all relevant enquires that are useful for various stakeholders in the system. This includes, for example, operations about enquires related to merchant information, customer available balance, list of customer transactions, etc.

### 3. Description of the System Use-Case View Operations

From a business point of view, the proposed software architecture should be able to support the business and functional requirements. Through various use-cases the requirements, shows how the software architecture can fulfil and realize the business needs.

A use case describes how the system should respond under various conditions to a request from one of the stakeholders to deliver a specific goal. This is primarily done in the form of a scenario that describes a sequence of steps. Each use case captures a “contract” for the behaviour of the System.

#### Actors

An actor is anyone or anything with behaviour. The primary actor is the stakeholder that interacts with the system to achieve a specific goal. The primary actors in the SMW are:


- Consumers
- Merchants
- Organization Users

As described below, an actor is a role rather than a specific person, job title or thing.

#### Consumers

The system stores a large database of consumers representing the eligible citizens whom subsidy is intended to support. Consumer is assumed to interact with system through a Java enabled mobile that runs a simple application.

#### Merchants

The system stores a large database  of merchants that have to be registered in the system in order to be able to distribute goods and services. Merchant concept also includes all other forms of services. Merchants may include medical service providers, railway station offices, or merchants for food delivery. Merchant is assumed to interact with system through a Java enabled mobile that runs a simple application.

## Organization Users

To aid privacy and security for the system, each organization has a number of users that administer all of the organization activities. Each user is allocated a role of operation (e.g., Administration, reading, updating, or inserting).



## Scenarios

The main part of a use case is its scenario. A scenario describes a sequence of steps that are executed in order to fulfil the goal the use case is supposed to deliver. The most important scenario of the SMW system is the ration delivery to consumers. Figure 2 shows consumer journey.

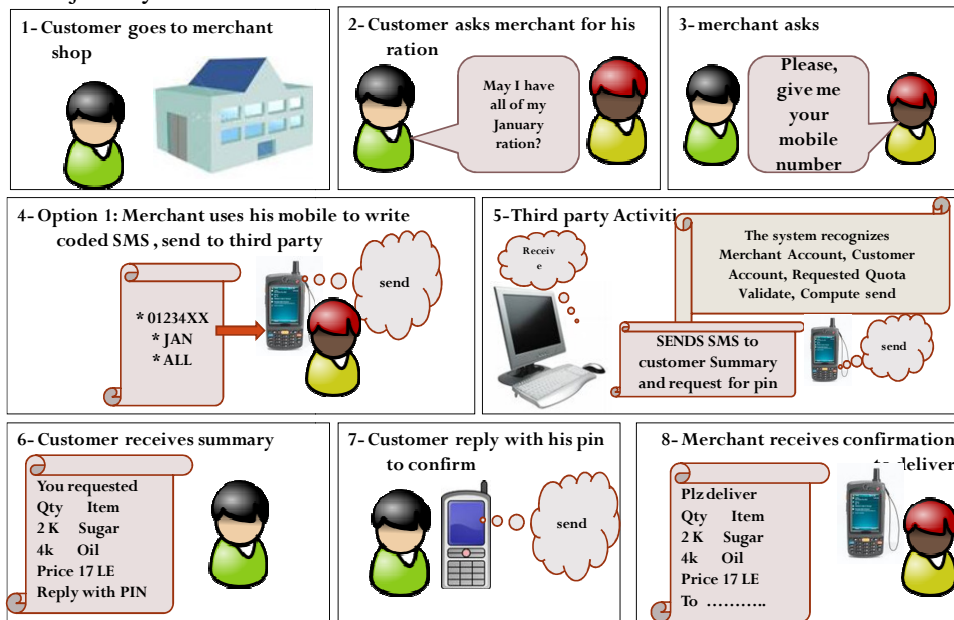


Figure 2: Conceptual view of quota delivery to consumer

The customer quota delivery is converted to a logical architectural view as shown in figure 3 .

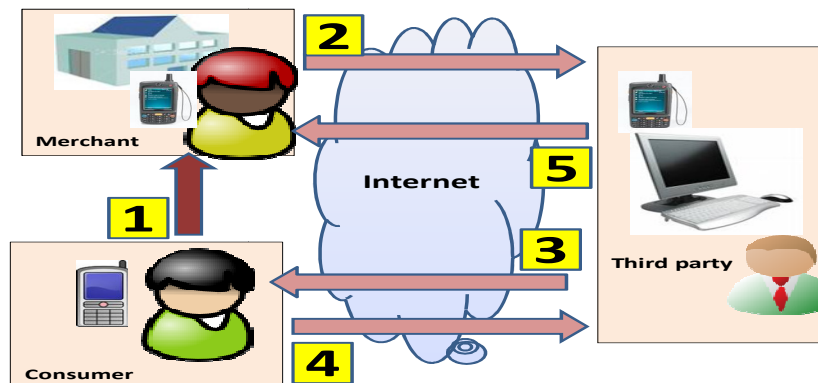


Figure 3: Logical architecture view of quota delivery

**The assumptions of this scenario are:**

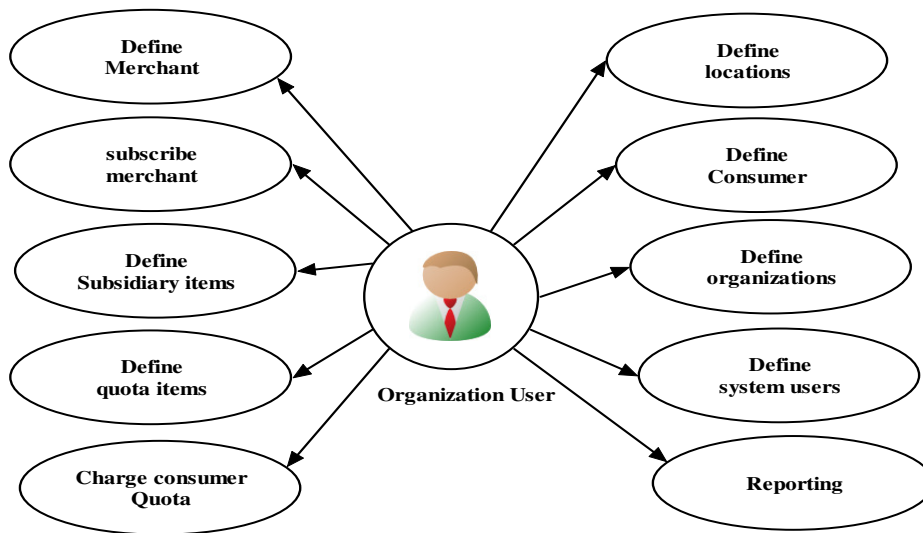
1. Customer requests all or part of his quota package.
2. Customer owns java enabled Mobile to run simple application.
3. Merchant owns java enabled Mobile to run simple application.
4. Third party has a computer connected to a mobile.
5. Merchant has some IT skills to run a mobile application.
6. Customer can read confirmation messages, and can reply with his pin code.

**Use Cases**

Figure 4 is a high level description of each use case identified in the SMW design.

**Organization Use Cases**

**Business Description**



**Figure 4: Organization use cases**

**Define organizations**

Organization represents third-party that owns and run subsidies distribution system. Currently, many organizations - governmental and non-governmental- inject many subsidiary aids into society. However, there is no estimate of the total amount of these subsidiaries, due to random distribution strategies followed by each organization. The current system allows many organizations to share a common consumer data base, to deliver their aids. Therefore, the amount of subsidiaries delivered to the society is accurately estimated. Each organization is characterized by its name, address, contact person, etc.

**Define Consumer**

The system stores a large database of consumers representing the eligible citizens whom subsidy is intended to support. To ensure that each consumer is stored once, the primary key is selected to be his/her national id. This also helps integration with national databases in future phases of the project. Other main features of a consumer are his/her name, mobile phone number, pin code, and number of persons associated with each consumer; for group-based quota to be calculated correctly.

**Define Locations**

International Journal of Computer Science & Information Technology (IJCSIT), Vol 3, No 1, Feb 2011  
To help analysis of the fair subsidiary distribution over rural-urban or middle-poor areas, geographical information is associated with consumers.

### **Define Merchant**

The system stores a large database of merchants that have to be registered in the system in order to be able to distribute goods and services. To ensure that each merchant is stored once, the primary key is selected to be his/her national id. This also helps integration with national databases in future phases of the project. To help analysis of the fair subsidiary distribution over rural, urban or cities, geographical information is associated with merchants.

### **Define subsidiaries delivered by organization**

The system allows each governmental and nongovernmental organization to offer many subsidiary quotas. For example nongovernmental organizations can offer clothes, blankets, or even 'Adahi' meat quotas. Governmental sector such as Petroleum Ministry offers gas canisters quota. Ministry of Social Solidarity offers food quotas.

### **Personal and group-based quota**

The quota can be personal or group-based quota. Personal quota is computed for only one person- the consumer. Single personal quota is suitable for many subsidiary assistance forms such as gasoline, and non-governmental assistances. Group-based quota is calculated for all members represented by the consumer. The group-based quota is the normal figure of food subsidiary delivery in any country.

### **Define Periodical forms of quota**

Quota is also characterized by its time periodical forms, i.e., yearly, monthly, weekly, daily, or once. The monthly-based quota is the normal subsidiary form in any country for food. However, the system allows much other timely-based assistance. For example bread subsidiary can be daily, railway subscription assistances for students and employees may be yearly based, and non-governmental subsidiary may be given only once.

### **Define Quota Items**

The system allows third party to determine dynamically the quota components. It may differ from month to month. For each item, the cost, merchant price, and consumer prices are determined. Organization sets quantities per person for each item (for example ½Kgm of oil, 1Kgm of sugar).

### **Charge consumer Quota**

The system allows automatically notifying consumers when their balance is credited with new rations. Since this feature is suitable for governmental quota, the system allows third parties to determine whether consumers should be notified for charging consumer accounts or not. When setting the features of "Regular" and "consumers should be notified?" to yes,, the system charges all consumers accounts with the specific quotas at right time (according to period setting), and notifies them. Normally, these features are set to yes when all of the consumers stored in the database are targeted by rations.

### **Subscribe Merchant as quota delivery**

The system allows each merchant to deliver many quotas as long as he/she is subscribed as a quota provider. The constraint is that a unique mobile number for a merchant is assigned for each subscription.

### **Quota Delivery Use Cases**

Figure 5 shows quota delivery use cases

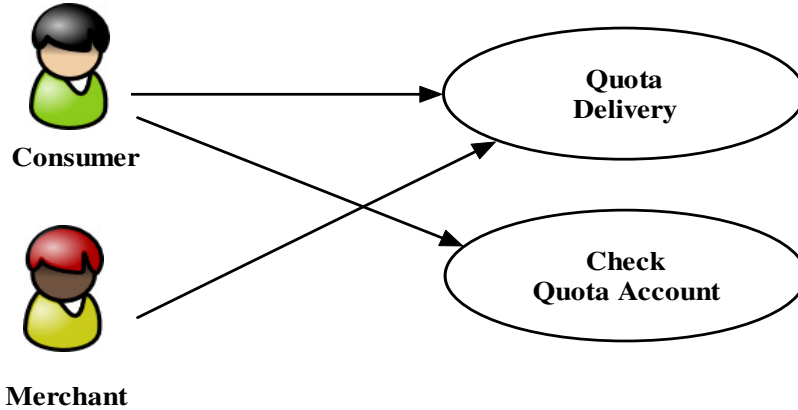


Figure 5: Quota Delivery Use Cases

### Business Description

The system records delivery of quota items into customer voucher header and details. The header carries the customer id, quota id, and merchant id. Also, the header includes total quota prices for merchant and total quota price for consumer. This allows computation of the merchant profit, and helps for informing consumers with their share price of quotas. Other attributes are delivery period name, which can be Feb, March, etc, year, and actual delivery date. The details of the voucher includes the items and their rated (Formal) quantities as defined in quota detail by third party organization. Also, actual delivered quantity is stored to allow partial delivery of rations. The price of each item is also stored from merchant and customer points of views are registered to compute totals stored at the head.

### Merchant Use Cases

Figure 6 shows merchant use cases

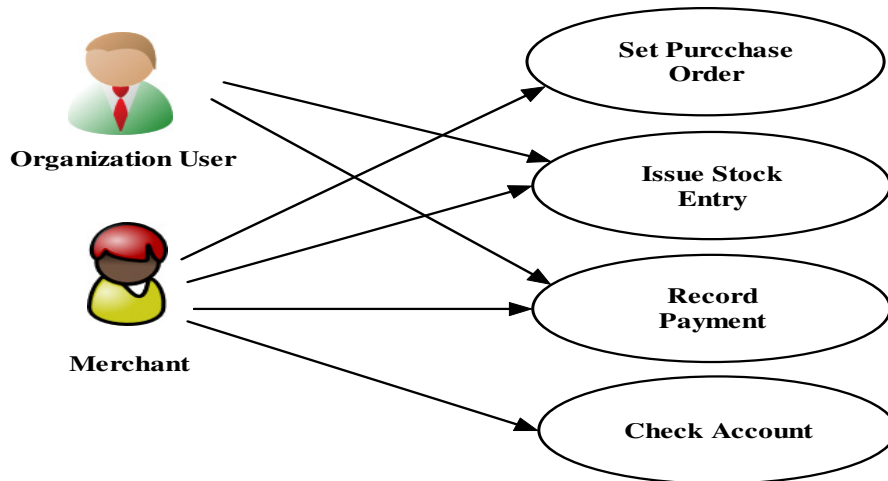


Figure 6: Merchant Use Cases

### Merchant Purchase and payment

Each merchant has one and only one store for each subscription. The store include quota items.

#### 4. Description of the System Runtime View Operations

In order to reduce the technical risks associated with hypothetical software architecture early during the software life cycle when it is the most cost effective to introduce change, the software engineering process utilized requires the creation of an architectural proof-of-concept. An architectural proof-of-concept is an actual software application constructed in order to test and validate the proposed software architecture prior to the creation of the software architecture document. Figure 7 shows the architecture design of SMW run time view system.

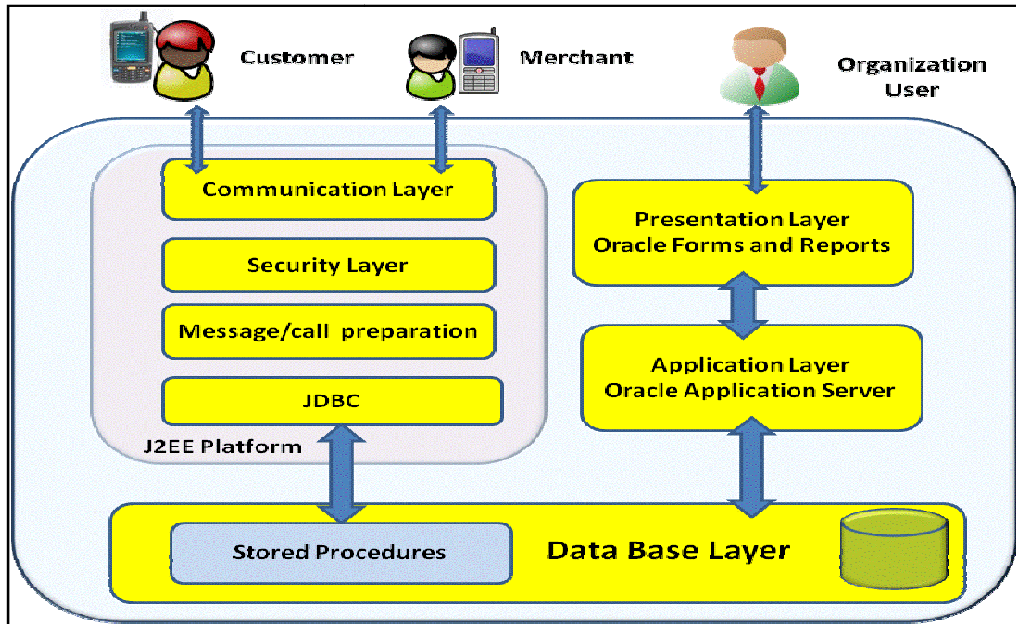


Figure 7: SMW run-time view

#### Figure 7 : SWM Architecturally Significant Features

The architecturally significant features that have been followed include the following:

- Object-Oriented Software Development Methodology
- Layering
- Relational Database Management System
- SMW Technical Architecture
  - Windows 2003 Server
  - Oracle RDBMS
  - JAVA J2EE platform
  - Oracle Forms and Report Server
  - Oracle Forms and Report builder
  - JDBC for oracle
  - Mobile .....

These decisions have directly or indirectly influenced the design of the SMW System and they are reflected as appropriate in the architectural view representation of the SMW software architecture.



### **(A) Object-Oriented Software Development Methodology**

The SMW System is being developed using an object-oriented development methodology; a methodology that is based on the concepts of classes, objects, data abstraction, encapsulation, messages, and inheritance. The various objects that comprise a software application have relationships, and collaborate with each other, to perform the work of the system through message passing. One of the principal advantages of an object-oriented development methodology is the ability to change existing objects or add new objects to the software system with minimal impact to the other objects that comprise the system.

SMW System is being developed using an object-oriented development methodology. J2EE platform is used to implement communication layer, security layer, message preparation and passing, and data base connections.

### **(B) Layering**

Layering is an architectural design pattern that structures applications so they can be decomposed into groups of subtasks such that each group of subtasks is at a particular level of more manageable abstraction. A layer represents a slice through the software architecture, with each layer representing a grouping of related functionality. Layers restrict inter-system dependencies with the goal being to design a system that is more loosely coupled and thus easier to maintain. The software component within a given layer should ideally access only components within its own layer or components in the layers beneath it. A common application of the layers design pattern organizes and defines the various layers within the problem domain based upon the responsibilities assigned to each layer. The system is partitioned into the following layers:

- **Presentation Layers**

Presentation layer provides support for the interactions between the actors, or the users of the system, and the software system itself through the presentation of user interfaces. It is the topmost level of the application. It communicates with other tiers by outputting results to the browser/client and mobile application SMW tiers. In the current design, two presentation layers do exist. The first layer resides within thin client running normal web browser for subsidiary control at organization's side. The second presentation layer is implemented with java to run over merchant and consumer mobile phones.

- **Business Logic Layer**

The logic tier is pulled out from the presentation tier. It controls an application's functionality by performing detailed processing. This middle layer provides support for application specific business processes, as well as, the application and enforcement of business and data integrity rules.

- **Data Access Layer**

Data access layer tier consists of Database Servers. This bottom layer provides support for data access and persistence of the relational database. Here information is stored and retrieved. This tier keeps data neutral and independent from application servers or business logic.

- **Communication Gateway layer**

In addition to the above traditional layers, additional communication gateway layer is required to provide the services specific to fulfil merchants and customer requests. This layer provides an intermediate layer between the presentation layer at the mobile phones of merchants and customers from one side, and the stored procedures that respond to actor requests from the other side. It includes communication layer, security layer, message interpretation, and JDBC.

### (C) Relational Database Management System

Relational databases are currently the predominant choice in storing financial records, manufacturing and logistical information, personnel data and much more. The SMW System will utilize an Oracle RDBMS and relational databases created within this environment to store the project's persistent information including consumer data, merchant data, quota details, and other data.

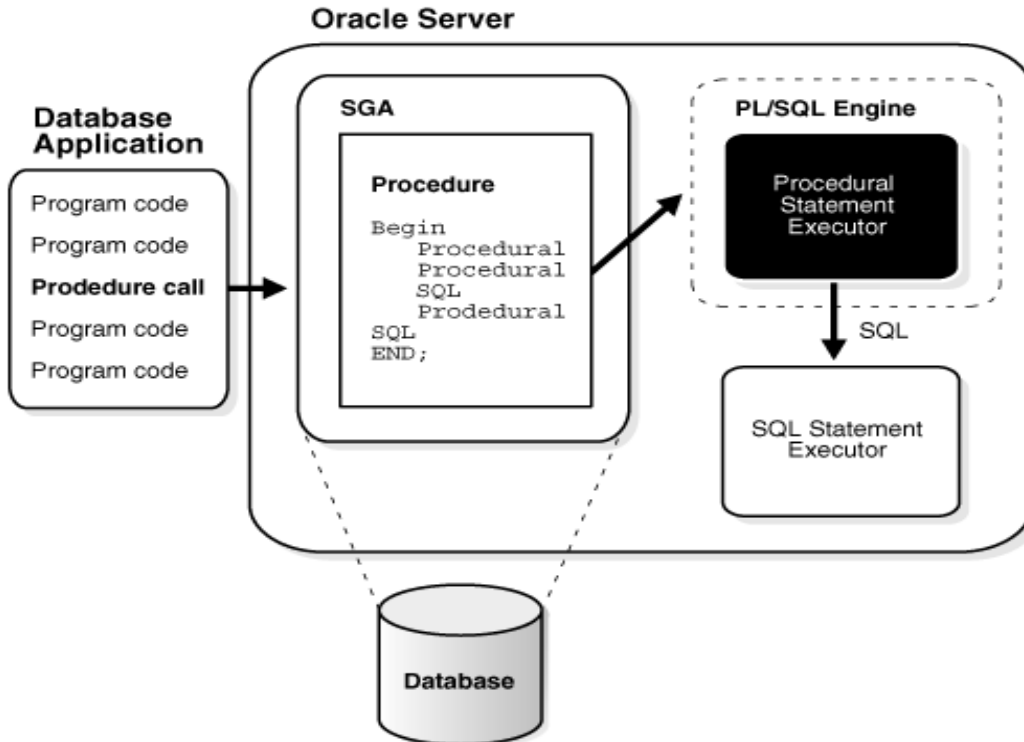


Figure 8: The PL/SQL Engine and the Oracle Database Server

Oracle has been designed with focus on certain key development areas such as Application Development, Internet Content Management, Data Integration, Availability, Manageability, and Security. Oracle offers development platform for Internet and traditional application development. This includes XML, Enterprise Java Engine, SQL and PL/SQL procedures. These platforms are very important during implementation stage.

### (D) SMW Technological Architecture

#### Stored Procedures

A stored procedure is executable code that is associated with, and generally stored in, the database. Stored procedures usually collect and customize common operations, like inserting a tuple into a relation, gathering statistical information about usage patterns, or encapsulating complex business logic and calculations. Frequently they are used as an application programming interface (API) for security or simplicity.

Implementations of stored procedures on SQL DBMSs often allow developers to take advantage of procedural extensions (often vendor-specific) to the standard declarative SQL syntax.

Stored procedures are not part of the relational database model, but all commercial implementations include them.

International Journal of Computer Science & Information Technology (IJCSIT), Vol 3, No 1, Feb 2011  
 A stored procedure performs one or more specific task. This is similar to a procedure in other programming languages. A procedure has a header and a body. The header consists of the name of the procedure and the parameters or variables passed to the procedure. The body consists or declaration section, execution section and exception section similar to a general PL/SQL Block. A procedure is similar to an anonymous PL/SQL as shown in figure 8

**JDBC Architecture**

The JDBC API contains two major sets of interfaces: the first is the JDBC API for application writers, and the second is the lower-level JDBC driver API for driver writers. JDBC technology drivers fit into one of four categories. Applications and applets can access databases via the JDBC API using pure Java JDBC technology-based drivers, as shown in figure 9:

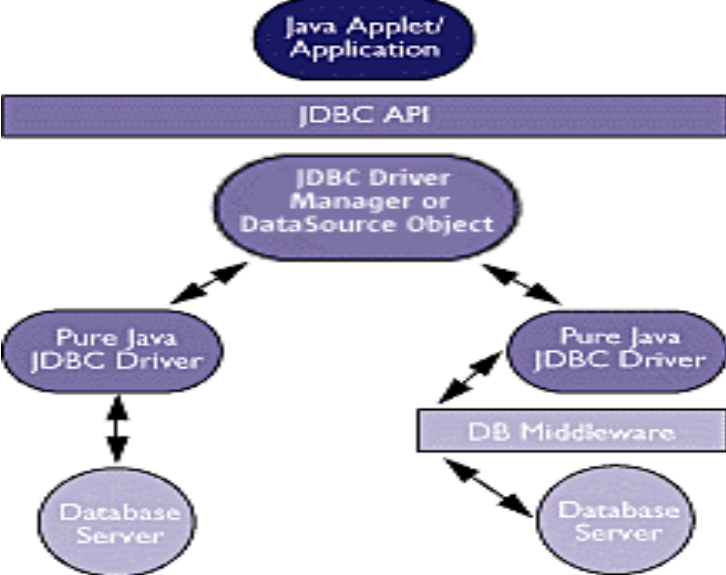


Figure 9: Java JDBC technology-based drivers

**Oracle Forms Server architecture**

With Forms Server, complex applications can run over the Internet, without compromising either functionality or richness of interface. The Forms Server consists of a Java Client that is downloaded automatically to the end user and three components in the middle tier “Application Server:” Figure (10) illustrates how a form runs on the web.

**Oracle Developer Forms**

Oracle Developer Suite Oracle Forms and Oracle Application Server Forms Services provide a complete application framework for optimal deployment of Forms applications on the Internet. Together they deliver a Rapid Application Development (RAD) environment and application deployment infrastructure to ensure that your Internet applications automatically scale and perform over any network.

Rather than deploying a series of separate and independent applications, companies are seeking an integrated solution -- applications that can work together. Through its openness and integration with Oracle Application Server, Oracle Forms provides an extensible framework to access other technologies, like Java and XML, to help meeting the needs to build applications that can be deployed in today's complex network and Web environments.

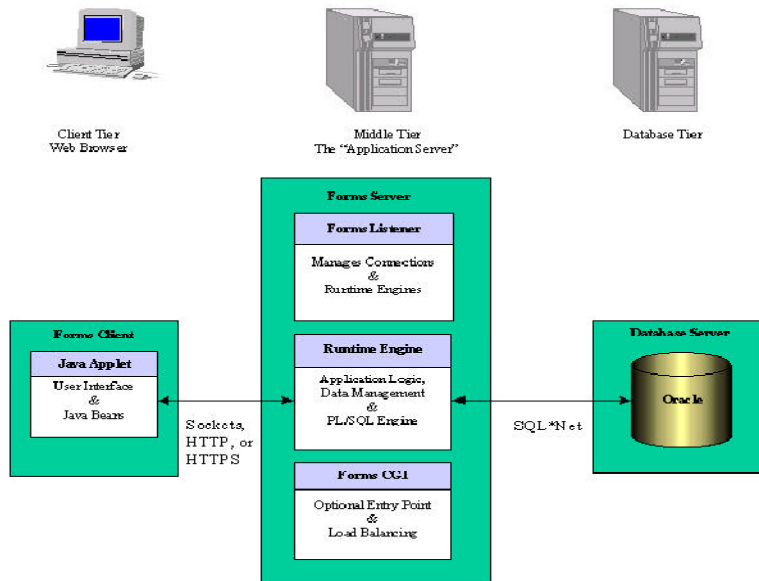


Figure 10: How forms run on the web

## 5. Description of the System Deployment View operations

The system consists of four gross components, namely,

- Customer
- Merchant
- Third Party
- SMW server

The deployment diagram in figure 11 shows the structure of each component together with the interconnections between components.

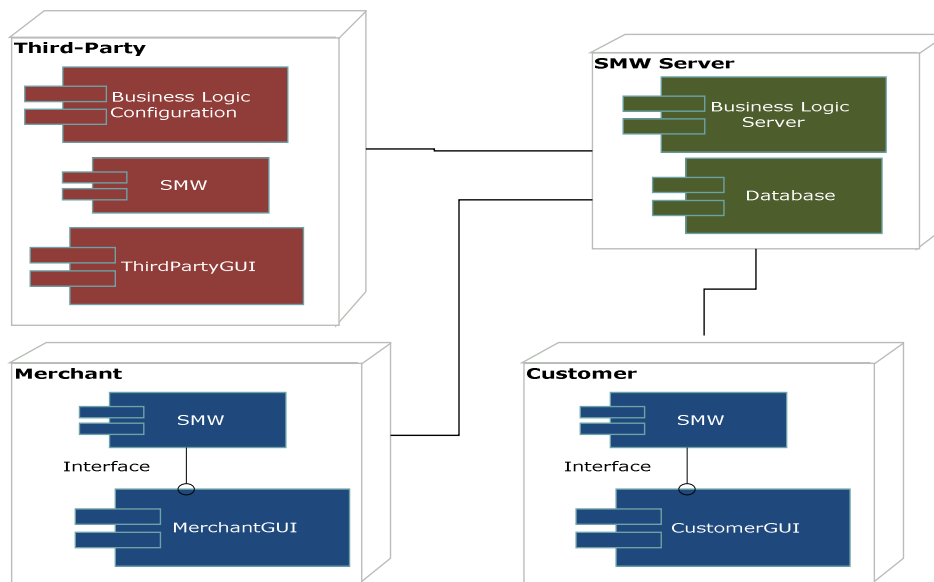


Figure 11: Deployment diagram of the proposed system

## 6. Conclusion

The purpose of developing a Subsidies Mobile Wallet (SMW) is to make the delivery of subsidies more easy through the use of mobile phones and the wireless networks. The SMW design tries to treat the limitations of previous techniques used in delivering subsidies, and to provide an efficient distribution of subsidies for both governments and NGOs. In [15], an innovative solution for a flexible **Subsidies Mobile Wallet (SMW)** through mobile phones and wireless networks was proposed to find a way to satisfy the technical requirements necessary for enhancing the subsidies distribution. Also, in [15], the Logical View, the Architectural Design, and the Process View of the **SMW** were described in details. In this paper, design solution for **SMW** is proposed from different views. These new views are: the System Use-Case View Operations, the System Runtime View Operations, and the System Deployment View. Each view illustrates the relationships between the subsidy distribution actors and elements. In the System Use-Case View Operations, through various use-cases, the requirements show how the software architecture can fulfil and realize the business needs. A use case describes how the system should respond under various conditions to a request from one of the stakeholders to deliver a specific goal. The different use-cases mentioned in this paper are: Organization Use Cases, Quota Delivery Use Cases, and Merchant Use Cases. In the System Runtime View, trials were done to reduce the technical risks associated with hypothetical software architecture during the software life cycle, because it is more cost effective to solve these risks early. The software engineering process utilized requires the creation of an architectural proof-of-concept. An architectural proof-of-concept is an actual software application constructed in order to test and validate the proposed software architecture prior to the creation of the software architecture document. The architecturally significant features that have been followed include the following: Object-Oriented Software Development Methodology, Layering, Relational Database Management System, SMW Technical Architecture (Windows 2003 Server, Oracle RDBMS, JAVA J2EE platform, Oracle Forms and Report Server, Oracle Forms and Report builder, JDBC for oracle, Mobile). These decisions have either direct or indirect influence on the design of the SMW System and they are reflected as appropriate in the architectural view representation of the SMW software architecture. In the System Deployment View, the system consists of four gross components, namely, Customer, Merchant, Third Party, and SMW server, and the structure of each component together with the interconnections between components are illustrated.

## Acknowledgement

The author would like to acknowledge the financial support of the project No. C2/S1/145 of EU - Egypt Innovation Fund . Where this paper has been produced with the financial assistance of the European Union. The contents of this paper are the sole responsibility of the author and can under no circumstances be regarded as reflecting the position of the European Union.

## References

- [1] [http://en.wikipedia.org/wiki/Food\\_security](http://en.wikipedia.org/wiki/Food_security)
- [2] Natalia Carolina Maya Ortiz, “ Analyzing Market Reforms and Food Distribution Systems in Accra, Ghana: Lessons and Implications for Reforming the Bazarro Market of Cartagena, Colombia ” , Massachusetts Institute of Technology, UAS, January 2010.

- International Journal of Computer Science & Information Technology (IJCSIT), Vol 3, No 1, Feb 2011
- [3] Joseph C. Salvacruz, “ Competitiveness of the United States and the ASEAN in the International Agricultural Market ”, Journal of Food Distribution Research, February 1996.
- [4] Eileen T. Kennedy And Harold H. Alderman , “ Comparative Analysis Of Nutritional Effectness Of Food Subsidies And Other Food-Related Interventions “ , International Food Policy Research Institute, Massachusetts, U.S.A., March 1987.
- [5] <http://www.reference.com/browse/subsidy>
- [6] Debra P. Steger, Thomas & Partners, Ottawa, Canada, “ The WTO Doha Round Negotiations on Subsidies and Countervailing Measures: Issues for Negotiators, Symposium on Economic Restructuring in Korea “ , Seoul, Korea, May 2003.
- [7] Antonio La Vina, Lindsey Fransen, Paul Faeth, Yuko Kurauchi, Reforming, “ Agricultural Subsidies “No Regrets” Policies for Livelihoods and the Environment “ , World Resources Institute, 2006.
- [8] <http://law.irank.org/pages/4201/Agriculture-Subsidies.html>
- [9][http://www.servinghistory.com/topics/subsidized::sub::Types\\_Of\\_Subsidies](http://www.servinghistory.com/topics/subsidized::sub::Types_Of_Subsidies)
- [10] Report of the Secretary-General, “Policy options and actions for expediting progress in implementation: Agriculture”, United Nations, Economic and Social Council, Commission on Sustainable Development, Seventeenth session, May 2009
- [11] Experiences in Targeting Nutrition Programs, FAO Corporate Document Repository  
<http://www.fao.org/DOCREP/004/Y1329E/y1329e07.htm>
- [12] National Nutrition Networks Conference,  
<http://nrha.ruralhealth.org.au/otherconferences/docs/NNNC08DraftRecommendations.pdf>
- [13] “ Recovering from Economic and Financial Crisis: Food Security and Safety Nets “ , Background Document Prepared Jointly By UNDP/UNFPA, UNICEF and WFP, January 2010.
- [14] J.S. Pettersson (ed.), Proceedings of 1st International Conference on M4D Mobile Communication Technology for Development, General Tracks of M4D 2008, Center for Human IT, Karlstad University, Sweden, December 2008.
- [15] Gamal Darwish, Ibrahim Eleter and Dina Darwish, “ Developing a Subsidies Mobile Wallet through Mobile Phones and Wireless Networks “ , Proceedings of the International Conference on Computer, Electrical, and Systems Science and Engineering – ICCESSE, Paris – France, June 2010, pp. 2201-2211.
- [16] C. Hofmeister, R. L. Nord, and D. Soni, “ Describing Software Architecture in UML” Proceedings of the First Working IFIP Conference on Software Architecture (WICSA1), Kluwer Academic Publishers, Boston, Dordrecht, London 1999, pp. 145-159
- [17] I. Sommerville, “Software Engineering” 8<sup>th</sup> eds., Addison-Wesley, Pearson Education 2008.
- [18] I.F. Cruz And H. Xiao, “ Using a Layered Approach for Interoperability on the Semantic Web “ , Proceedings of the Fourth International Conference on Web Information Systems Engineering (WISE03). 2003.
- [19] J. Savolainen and V. Myllärniemi, “ Layered architecture revisited - Comparison of research and practice “ , Joint Working IEEE/IFIP Conference on Software Architecture, Cambridge, UK, September 2009.