USAID | DELIVER PROJECT
FROM THE AMERICAN PEOPLE

HEALTH COMMODITY MANAGEMENT INFORMATION SYSTEM (HCMIS) SOFTWARE DOCUMENT

FOR FACILITY AND WAREHOUSE (HUB) EDITIONS

JULY 2010

This publication was produced for review by the United States Agency for International Development. It was prepared by the USAID | DELIVER PROJECT.

HEALTH COMMODITY MANAGEMENT INFORMATION SYSTEM (HCMIS) SOFTWARE DOCUMENT

FOR FACILITY AND WAREHOUSE (HUB) EDITIONS

USAID | DELIVER PROJECT
John Snow, Inc.
1616 Fort Myer Drive, 11th Floor
Arlington, VA 22209 USA
Phone: 703-528-7474
Fax: 703-528-7480
E-mail: askdeliver@jsi.com
Internet: deliver.jsi.com

The authors' views expressed in this publication do not necessarily reflect the views of the United States Agency for International Development or the United States Government.

Contents

Figures

Tables

Abbreviations

- LMIS - Logistics Management Information System
- FEOS - First to Expire First Out
- PFSA - Pharmaceutical Fund and Supply Agency
- HCMIS - Health Commodity Management Information System
- ARV - Anti RetroViral
- STV - Stock Transfer Voucher
- IM – Inventory Management
- DBMS – Database Management System
- IDE – Integrated Development Environment
- ORM – Object Relational Management

1 | INTRODUCTION

Objectives

This chapter deals with the following issues:

- ❒ General Overview of the HCMIS System
- ❒ Why a New (Automated) System is required
- ❒ Purpose and Goals of the HCMIS System
- ❒ Scope of the HCMIS System
- ❒ HCMIS Design Principle from Ethiopian Context
- ❒ Technology Selection and Software Licensing Principles of the HCMIS System
- ❒ The HCMIS Software Document Organization

1.1. Overview

An efficient inventory control and warehouse management system plays a pivotal role in a Logistics Management Information System (LMIS). A properly crafted inventory control system would help generate important and timely stock status reports that would assist in a proper decision making. It can also be more useful for supporting the supply chain of the logistics management. The warehouse management module could also help efficiently track the exact location of items within the warehouse. This is essential for enforcing standard working procedures in a pharmaceutical logistics. Some of the standards that are better supported by the system include First to Expire First Out (FEFO), batch and expiry tracking.

Currently, USAID-DELIVER in collaboration with different other partners is providing technical assistance to Ethiopian Ministry of Health in the implementation of an efficient pharmaceutical management system. Pharmaceutical Fund and Supply Agency (PFSA) handles various processes at the warehouse (i.e. hub) as well as facility levels using a very minimal manual system. In the manual system the commonly used bin or stock card systems are put in place. However, due to the complexity of the transactions and lack of properly setup system, the manual system fails to provide the much desired timely reports. Moreover, lack of a systematic record keeping system, that can help efficiently manage daily transactions at the warehouse as well as facility levels, is among the main problems to mention. Besides, generating appropriate reports, that can help make sound decisions, is difficult.

Taking these facts into account, a new computerized system that is referred to as Health Commodity Management Information System (HCMIS) is designed and developed to alleviate problems that are inherent in the current manual system. The HCMIS is an inventory control plus warehouse management software developed using local talent. A deployment of this software at different facilities enables the facilities to enforce accepted standard practices. Some of these features include efficient management of processes to expiry dates and batch tracking.

Detail illustration of the HCMIS system will be discussed in the forthcoming sections.

1.1. Why a New (Automated) System?

The manual system that is being in use experienced a number of limitations. The fact that the existing manual system was so disorganized lead to lack of proper record keeping mechanisms. Even if there is a record keeping procedure, it's mainly for accounting purposes. Besides, generating appropriate reports that are essential to make sound decisions regarding the buying of necessary supplies, checking the expiry dates of unused supplies, and locating items in the warehouse and/or facilities is complex. Moreover, the manual system has specific limitations on handling the following tasks:

- properly locating items in the warehouse and/or facilities
- properly handling the transaction of receiving and/or issuing medical supplies
- Managing the movement of items with in the warehouse or facilities
- Efficiently controlling expiry dates
- Wastage of time, labor and money looking for inventory
- Controlling over stock and minimum stock levels
- Increasing inventory accuracy
- Generating various reports

1.2. Purpose and Goals

The HCMIS system is designed and developed in order to alleviate problems that are experienced in the hub as well as facility levels. In this document, the approaches that are put in place to efficiently tackle the limitations of the current system are outlined.

The approaches include requirement analysis, design, implementation and all other related phases that are vital for the successful implementation of the HCMIS system. Moreover, the general purpose of this document is to portray or document how issues like increasing inventory accuracy, batch processing, expiry date tracking (i.e. using FEFO method), keeping inventory from being out of balance, keeping track of important transactions, and fast and flexible item search can be effectively implemented.

These features will help maintain inventory without overstock or expiration, improve staff productivity, centrally plan and manage, reduce order entry errors, enhance management processes, and eliminate manual, paper-based processes.

1.3. More Than Just an IM System

HCMIS is designed as an inventory management system that helps health facilities track their commodities. However, the application is more than just an inventory tracking system. It includes major components of the Logistics Management Information System in which it provides the user site ability to track consumption rates, manage re-order levels, enforce FEFO as well as provide many advanced logistics reports that allow proper facility and warehouse management rules to be enforced by the facility.

1.4. Scope

The Health Commodities Management Information System, HCMIS, is an inventory management and LMIS application designed to operate both at the Facility Level (branded as 'Facility Edition') such as Hospitals and large Health Centers as well as the Hub Warehouses (branded as 'Hub Edition').

While HCMIS is presented to the end user either as a Facility or Hub Edition and each have their own separate User Manuals, the System Documentation is common. Architecturally, both Editions come from a single source package; share a single database, design elements as reflected in the common features and functionality.

Moreover, the analysis, design and implementation of the HCMIS software components and their details are briefly described in this document. Generally, this document focuses primarily on describing the different phases in the analysis, design and implementation of the HCMIS system.

Specifically, this HCMIS system covers the following aspects:

- properly locating items in the warehouse or facilities
- Handling the transaction of receiving and/or issuing pharmaceuticals
- Managing the movement of items with in the warehouse or facilities
- Efficiently managing expiry dates using the FEFO system
- Applying the batch and recall methods
- Efficient data management and backup
- Security of the system

1.5. Design Principle and Ethiopian Context

The Health Commodities Management Information System, HCMIS, is designed with the aim to develop a system that was as ‘simple as possible’ application but yet meet the Ethiopian Government’s requirement of proper LMIS. This vision required a system that can be implemented for the country that was best fit for the type of facilities, warehouses as well as the skill level of those who are expected to operate the system.

Therefore, while the system provides all the requirements in order to operate a facility and/or a warehouse, deliberate decision was made to not include advanced features that would normally be found in commercially available inventory management tools. The user is therefore advised not to read this document from the lenses of such commercial level tools but only from the expectation of the government of Ethiopia requirement.

1.6. Local Need and Local Development

The entire work of HCMIS was developed in Ethiopia from inception, design, development, testing and deployment to ensure the project was as localized as possible.

1.7. Technology Selection

The technology used in HCMIS was picked after a careful deliberation of various factors which included 1) availability of skilled and qualified developers/programmers, 2) adoption and acceptance of technology in other sectors and the local IT job market 3) cost effectiveness of technology expenditure requirements

1.8. Software Licensing Principles

HCMIS was designed with a principle to ensure the deployed product has no end user licensing requirements attributed to it and both the application as well as the backend database system had to be license free.

1.9. Document Organization

This document is organized into six sections and various subsections. Section 1 highlights some introductory remarks of the system where as section 2 describes the current system environment which includes details about the current system's processes and limitations.

Section 3 gives the insight of the requirements and analysis of the HCMIS system. In this section, therefore, the functional and non-functional requirements, and the system models are briefly illustrated.

Under section 4, issues related to the architecture of the system, subsystem decomposition, hardware/software mapping and object relational mapping are explained in detail. Moreover, section 5 covers the implementation issues of the HCMIS system which comprehends the details of the development tools used and description of the user interface components together with users' access control.

Finally, general remarks are given under section 6 of this document.

2 | THE CURRENT SYSTEM ENVIRONMENT

Objectives

This chapter deals with the following issues:

- ❐ Brief Description of the Current System
- ❐ Limitations of the Current System
- ❐ Identifying the Current System Processes

2.1. The System Description

As described in the previous sections, the current system is a manual system that is implemented for both the hub and different facilities. With all its limitations, the current system is used to manage different tasks in the hub as well as facilities.

The two primary tasks in the hub are the warehouse management and inventory control. The warehouse management wields the movement and storage of stock in the hub. Overseeing and recording deliveries and pickups, maintaining inventory records and tracking, determining appropriate places for storage and controlling inventory levels are some of the several crucial functions in the hub. The relatively small amount of stock maintained at the facility stores makes the warehouse management less of a concern. However, inventory control is a major functionality at the facility stores. Hence, paramount importance is given to the receiving and issuing of pharmaceutical supplies, managing the movement of items, and looking for a specific item in the facilities.

Currently, the different transactions at the hub as well as facility levels are being dealt with bin and/or stock cards. The bin or stock cards are expended to manage the daily records of the transactions. The major transactions are comprised of receiving and issuing stock, updating the bin or stock cards and performing inventory counts.

2.2. Limitations of the Current System

The existing system has major problems in terms of providing readily available and accurate reports regarding stock in the warehouse as well as facilities. The reasons for such inability are manifold. However, in this section, only some of the major limitations of the current system that are identified in the previous sections will be briefly explained.

□ PROPERLY LOCATING ITEMS IN THE WAREHOUSE

The warehouse and/or the facilities host quite large collections of items. However, the current system is insufficient to provide appropriate mechanisms to easily locate items in the warehouse. In the large warehouses, thousands of different drug types can be stored. In the current manual system, however, knowing the location of items that are about to be issued is left to the mercy of the memory of the inventory controller. The whereabouts of the items is forgotten at times and this in more than one occasion causes the items to have passed their expiry dates. As a result, locating items in the warehouse has been a difficult endeavor that causes the wastage of time, labor and money.

□ PROPERLY HANDLING THE TRANSACTION OF RECEIVING AND/OR ISSUING MEDICAL SUPPLIES

Receiving and issuing pharmaceutical supplies are the primary and routine activities in the warehouse as well as facilities. Failure to properly keep track of these major tasks can have a drastic effect on the rest of the activities within the warehouse and facilities. Currently, the manual system has flaws to properly maintain records of pharmaceutical supplies received and/or issued. Recording these transactions manually may lead to inventory inaccuracy.

□ MANAGING THE MOVEMENT OF ITEMS WITH IN THE WAREHOUSE OR FACILITIES

Managing the location and movement of items within the warehouse and/or facilities is one of the major limitations of the current manual system. Moreover, the current system lacks a reliable mechanism for tracking the movement of unused items, items that are close to expire, and items that are delicate and require too much care within the warehouse as well as facilities.

□ ENFORCING FEFO RULE IS NOT EASY

Failure to properly track expiry dates of the pharmaceutical supplies can have a hazardous impact. However, the current manual system has insufficient expiry date management. There is no proper mechanism that helps the inventory controller to enforce First to Expire First Out rule.

□ BATCH TRACKING IS NOT ENFORCED

Even though it is a strong recommendation and wish of the PFSA, the manual system doesn’t accommodate batch tracking at the hub and facility stores.

□ CONTROLLING OVER STOCK AND MINIMUM STOCK LEVELS

Stock level management is one of the major tasks in inventory control. Effective inventory control system requires appropriate stock level management. This in turn plays a pivotal role in increasing inventory accuracy. However, in the contrary, the current manual system fails short to provide suitable over stock and minimum stock level managements. Besides, lack of sufficient reporting system contributes a great deal to the problem of determining the minimum re-order level for the pharmaceutical supplies.

At time of over stock, there is a chance of disclosing fund tie and the problem of determining the emergency order point. It is also inevitable that the over stocked items may hold large sum of money which is important for ordering highly demanded pharmaceuticals.

□ GENERATING VARIOUS REPORTS THAT CAN HELP MAKE INFORMED DECISIONS

Due to deficiency of proper and systematic record keeping mechanisms, the current system lacks to generate vital reports that can help make sound decisions regarding the buying of necessary supplies, checking the expiry dates of supplies, and consumption levels and analysis across all facilities.

2.3. Current System Processes

The current manual system maintains records of pharmaceutical supplies and their routine transactions within the warehouse (i.e. hub) as well as facilities. The record keeping is ensured by the commonly used manual record keeping mechanisms; bin/stock cards. The different transactions are being recorded on the bin/stock cards for later reference.

Here are some of the major processes of the current system:

□ RECEIVING STOCK

- ✧ On receipt of drug supplies, goods receiving voucher is prepared
- ✧ The goods are inspected and put away by the inventory controller
- ✧ The inventory controller updates the stock card with the incoming amounts

□ ISSUING STOCK

- ✧ Depending on the program the drugs will be issued for, a stock transfer voucher may be prepared. For example, if the drug is an ARV medication, it is free item. Hence, a stock transfer voucher is prepared.
- ✧ For non-free items, a cash or credit invoice is prepared
- ✧ The items are picked using the numbers on the STV or invoice.
- ✧ The stock card is updated

□ PERFORMING INVENTORY COUNTS

- ✧ On year End, physical inventory count is performed and the bin card/stock card is updated with the loss and adjustment. The current year balance will be passed as a beginning balance for the next fiscal year.

3 | THE PROPOSED SYSTEM

Objectives

This chapter deals with the following issues:

- ❒ Identifying the Functional Requirements of the HCMIS System for the Warehouse (Hub) and Facility Editions
- ❒ Non-Functional Requirements of the HCMIS System
- ❒ Designing the System Model which includes the Use Case Diagram with its brief descriptions and the Class Diagram

In the previous sections, brief descriptions of the current system are given. These descriptions illustrate how the current manual system works, its limitations, and expected functionalities. Due to its limitations and lack of prominent functionalities, the current manual system fails short fulfilling the successful management of the warehouse and facilities. Hence, a new automated system (i.e. HCMIS) has to be put in place.

In this section, therefore, the functional as well as non-functional requirements of the proposed system are briefly described. Furthermore, system models such as use cases and class diagrams will be used for illustration purposes.

3.1. Functional Requirements

The proposed system should support the effective management of both the warehouse (hub) and facilities. The system to be implemented at the hub level should carry on the warehouse management as well as inventory control. However, at the facility level, only inventory control will be put in place. Taking this fact into account, the basic functionalities listed out in the next section are classified as those that apply to the warehouse (hub) and/or facilities.

Table 3-1: Functional Requirements that Applies to Warehouses (hubs) and/or Facilities

No	Functional Requirements	Applies to	
		Warehouse/Hub	Facilities
1.	Managing user accounts	☒	☒
2.	Managing system settings	☒	☒
3.	Editing pipeline information	☒	☒
4.	Adding/editing/deleting supplies list	☒	☒
5.	Adding/editing/deleting drug's information	☒	☒
6.	Managing hub information	☒	☒
7.	Customizing drug list	☒	☒
8.	Setting up warehouse profile including number of racks, pallet locations dimensions etc.	☒	□
9.	Maintaining manufacturer	☒	□

10.	Maintaining dimensions of packages for items per manufacturer	☒	□
11.	Maintaining different Storage types	☒	□
12.	Maintain preferred storage type and storage location for items in the warehouse	☒	□
13.	Maintain separate logical stores for different programs in the same warehouse	☒	☒
14.	Setting/selecting storage location for pallets and non palletized items	☒	□
15.	Confirming if items are actually put on their planned locations	☒	□
16.	Recording item requests from facilities	☒	□
17.	Approving amount to be issued for facilities	☒	□
18.	Generating pick list including the location, item name, batch number and expiry date of items to be issued	☒	☒
19.	Confirming if the issue items are actually picked up and issued, generating Stock Transfer Voucher	☒	□
20.	Consolidate non full pallets to save space	☒	□
21.	Palletizing items on receipt	☒	□
22.	Managing internal movements of items	☒	□
23.	Handling loss/adjustment	☒	☒
24.	Managing Pick Face replenishment	☒	☒
25.	Handling inventory control	☒	☒
26.	Controlling receive transaction activity log	☒	☒
27.	Controlling issue transaction activity log	☒	☒
28.	Controlling loss/adjustment log	☒	☒
29.	Controlling inventory log information	☒	☒
30.	Handling database backup and restore	☒	☒

31.	Exporting data to PFSA in PDA, Server and Excel formats	☒	☒
32.	Generating reports regarding the stock status, over stock items, stock out items, issues by receiving unit, expired products, near expiry products, and storage status	☒	☒
33.	Generating summary reports that include summary chart, stock expiry status and cost summary	☒	☒

1.1. Non-Functional Requirements

Among other things, the HCMIS System is designed by considering functionality, reliability, cost, ease of use, and time parameters. Better than what off-the-shelf systems could possibly provide, the HCMIS System (designed as custom system) gives careful consideration to the Ethiopian specific practices by discarding the cluttered and zillions of inventory and warehouse management practices that are not suitable for the Ethiopian context. In line with this, the system is designed considering the fact that it can be easily installed and maintained by the very limited number of relatively non-expert and non-IT personnel with low maintenance cost.

Moreover, the system conforms to the donor's assisted programs with multi-faceted interventions and commodities managed and accounted for specific health intervention programs. In addition to this, once the HCMIS System is ready for use, it is free to install in as many sites as needed. However, commercial applications will require per site/per user license and licensed database for each site that makes a country-wide deployment virtually out of question because of potentially millions of dollar of budget requirement.

This section, therefore, elaborates the non-functional requirements that are given due attention in the design and implementation of the system.

□ USER INTERFACE (UI)

The HCMIS shall be designed flexible and user friendly. Besides, it shall be designed as a graphical user interface that is easy to understand and use. The system shall also enable users to perform their business in effective and efficient way.

□ HARDWARE CONSIDERATIONS

As far as the existing hardware infrastructure is concerned, [for the HCMIS] there is no need of acquiring a special purpose hardware device. This makes the HCMIS affordable and less costly. Hence, the new system shall be designed to support single machine installation at the facilities.

□ BACKUP AND RECOVERY

The HCMIS shall support data backup and recovery. Moreover, the backup shall be taken from the warehouse and facilities on regular basis. Taking backups on regular basis will help restore/recover the HCMIS at times of system failure.

□ EXTENSIBILITY

The new system shall be designed in such a way that it supports the addition of new features and customizations at next major version upgrade.

□ ERROR HANDLING AND EXTREME CONDITIONS

The HCMIS shall have an automatic error handling mechanism which shall leave users informed about the errors that they have committed and so that they can rectify the problems.

□ PLATFORM COMPATIBILITY

The system shall be designed to run on Windows XP and/or later operating systems that integrate the .Net 3.4 runtime framework.

□ RELIABILITY

One of the pertinent non-functional requirements of the system is reliability. Hence, the HCMIS shall be designed and developed in such a way that it produces accurate and timely outputs (such as reports).

- □ SECURITY ISSUES

The HCMIS system shall require user name and password of users to grant them access to the system in general and to create, modify and delete information in particular. Moreover, the HCMIS shall require administrator ID and password from system administrators in order for them to undertake necessary system administration tasks. The HCMIS shall also be developed and maintained in compliance with internationally established guidelines and standards for protecting computer systems, networks and information.

- □ USABILITY BY TARGET USER COMMUNITY

User's acceptance of the system is one of the many significant things that the new system shall fulfill. Hence, the HCMIS shall be acceptable by the target user community. To this end, the system shall be designed and developed in such a way that it envisions all the requirements of users.

3.2. System Model

In this section of the document, users' requirements will be best illustrated using system models that can help visualize the functional requirements, the interaction of users with the system and the persistent data of the system. Therefore, use case and class diagrams are used.

1.1.1. Use Case Diagram

Figure 3-1shows the use case diagram of the HCMIS system which presents users' perception of the functionality and behavior of the system from an external point of view. Moreover, the requirements identified in section 3.1 are put together as use cases based on their common purpose and functionality.

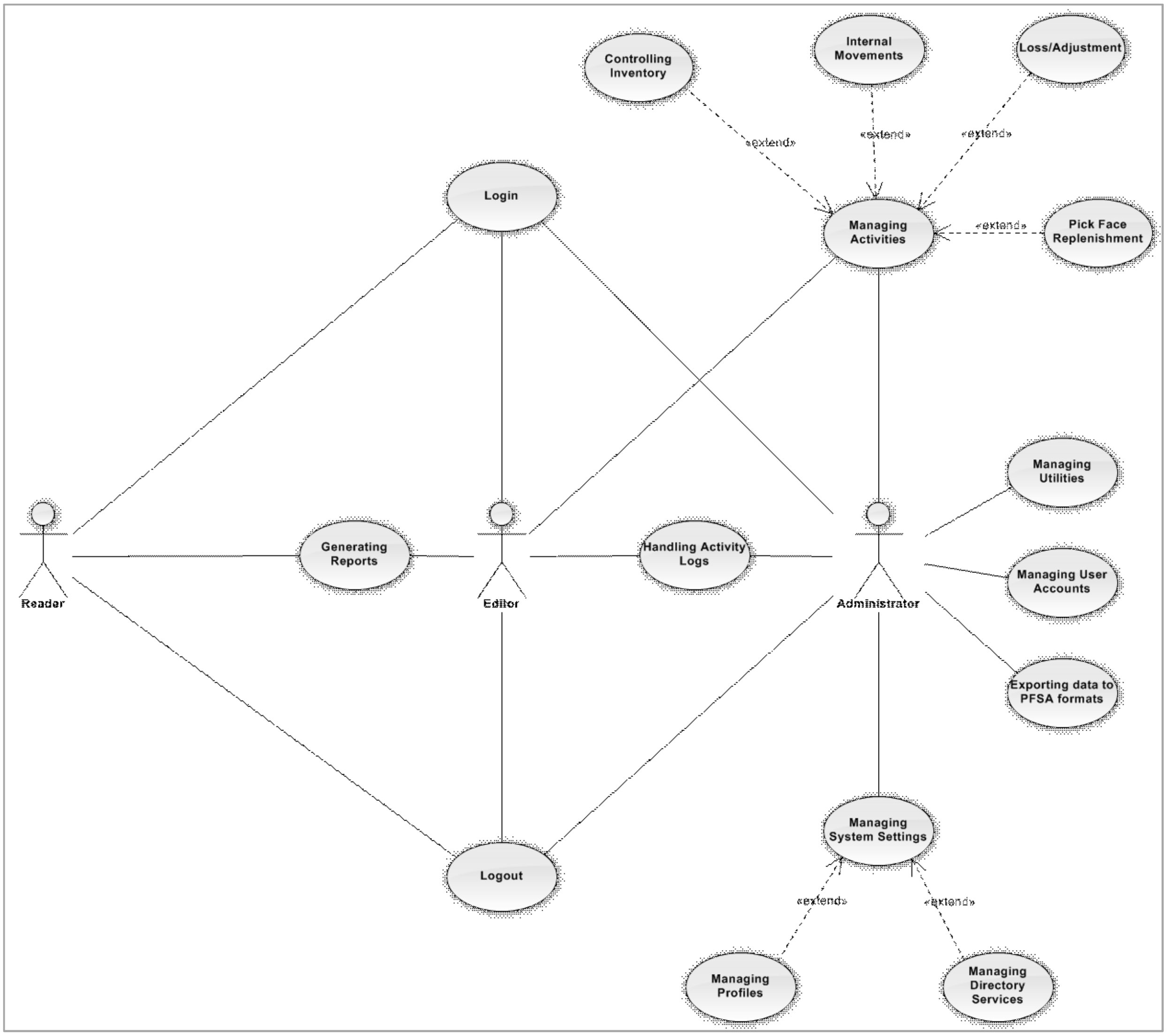


Figure 3-1: Use Case Diagram

□ USE CASE DIAGRAM DESCRIPTION

Use Case Id	UC-01
Use Case Name	Login
Actors	Administrator, Editor, Reader
Description	Allows to log into the HCMIS system
Pre-condition	The user should have an active account
Post-Condition	The system displays the main screen of the system
Includes:	None
Extends:	None
Basic Course of Action	1. The user activates the HCMIS software 2. The user enters user name password, and submits (Alternate Course A: *Incorrect user name or password*) 3. The system displays the main screen of the HCMIS system 4. The use case ends
Alternate Course A: *Incorrect user name or password*	A.2 The system displays the "Incorrect user name or password" error message A.3 The use case ends

Use Case Id	UC-02
Use Case Name	Managing User Accounts
Actors	Administrator
Description	Provides options to create new accounts, edit and delete already existing accounts to the administrator
Pre-condition	The administrator has an active account and logged into the system
Post-Condition	User Accounts will be managed successfully
Includes:	None
Extends:	None
Basic Course of Action	1. The administrator activates “User Accounts” option 2. The administrator selects the already existing user account to edit/delete (Alternative Course A: *Creating New User Account*) 3. The system displays the list of user accounts (modified/created) 4. The system displays acknowledgment message 5. The use case ends
Alternate Course A: *Creating New User Account*	A.2 The administrator selects “Add New User” option A.3 The administrator enters the details of the account A.4 The use case ends

Use Case Id	UC-03
Use Case Name	Managing System Settings
Actors	Administrator
Description	Allows the administrator to manage system profiles as well as directory services
Pre-condition	The administrator has an active account and logged into the system
Post-Condition	System profile and/or directory services information will be created, edited and/or deleted
Includes:	None
Extends:	Managing Profiles, Managing Directory Services
Basic Course of Action	1. The administrator activates "System Settings" option 2. The administrator selects the already existing profile and/or directory service to manage (UC-04, UC-05) 3. The system displays list of modified/created profile/director services information 4. The use case ends
Alternate Course	None

Use Case Id	UC-04
Use Case Name	Managing Profiles
Actors	Administrator
Description	Provides options to effectively manage system profiles
Pre-condition	The administrator have selected the “System Settings” option
Post-Condition	System profile information will be created and/or edited
Includes:	None
Extends:	None
Basic Course of Action	1. The administrator activates the appropriate option to add/edit profile information 2. The administrator adds/edits the profile (Alternative Course A: *Blank Field*) 3. The system displays list of created/modified profiles 4. The use case ends
Alternate Course A *Blank Field*	A.2 The system highlights the blank field A.3 Use case ends

Use Case Id	UC-05
Use Case Name	Managing Directory Services
Actors	Administrator
Description	Provides options to effectively manage directory services
Pre-condition	The administrator have selected the “System Settings” option
Post-Condition	Directory information will be created and/or edited
Includes:	None
Extends:	None
Basic Course of Action	1. The administrator activates the appropriate option to add/edit directory information 2. The administrator adds/edits the directory information (Alternative Course A: *Blank Field*) 3. The system displays list of created/modified directories 4. The use case ends
Alternate Course A *Blank Field*	A.2 The system highlights the blank field A.3 Use case ends

Use Case Id	UC-06
Use Case Name	Managing Utilities
Actors	Administrator
Description	Used to change the administrator's password and manage database backup
Pre-condition	The administrator has an active account and logged into the system
Post-Condition	The administrator backups the database and/or effectively changes the password
Includes:	None
Extends:	None
Basic Course of Action	1. The administrator activates the "Utilities" option 2. The administrator selects the "Database" option (Alternative Course A: *Administrator Selects "Change Password" option*) 3. The administrator selects "Backup/Export Database" option (Alternative Course B: *Administrator Selects "Restore/Import Database" option*) 4. The administrator selects a location to store the database backup 5. The system displays an acknowledgement message 6. The use case ends
Alternate Course A *Administrator Selects "Change Password" option*	A.2 The administrator enters the password in old, new and confirm password text entry boxes, and submits the data A.3 The system verifies the data (Alternative Course C: *Invalid Old Password Entry*) A.4 The system displays an acknowledgement message A.5 Use case ends
Alternate Course B *Administrator Selects "Restore/Import" option*	B.3 The administrator selects a location from which to restore/import the database B.4 The system displays an acknowledgement message B.3 Use case ends
Alternate Course C *Invalid Old Password Entry*	C.2 The system displays an error message C.3 Use case ends

Use Case Id	UC-07
Use Case Name	Exporting data to PFSA formats
Actors	Administrator
Description	Used to export data to PFSA for PDA, Server and Excel formats
Pre-condition	The administrator has an active account and logged into the system
Post-Condition	The administrator exports the data for PDA, Server or Excel formats
Includes:	None
Extends:	None
Basic Course of Action	1. The administrator activates the "Export to PFSA" option 2. The administrator selects the "PDA", "Server" or "Excel" option 3. The administrator selects the data to export 4. The system displays an acknowledgement message 5. The use case ends
Alternate Course	None

Use Case Id	UC-08
Use Case Name	Managing Activities
Actors	Administrator, Editor
Description	Used to control inventory, internal movements, loss/adjustment, and pick face replenishment of items
Pre-condition	The administrator/editor has an active account and logged into the system
Post-Condition	The administrator/editor effectively manages inventory, internal movements, loss/adjustment and pick face replenishment
Includes:	None
Extends:	Controlling Inventory, Internal Movements, Loss/Adjustment, Pick Face Replenishment
Basic Course of Action	1. The administrator/editor activates the “Activities” option 2. The administrator/editor selects an option from the list (UC-09, UC-10, UC-11, UC-12) 3. The use case ends
Alternate Course	None

Use Case Id	UC-09
Use Case Name	Internal Movements
Actors	Administrator/Editor
Description	Used to control manual stock movements, suggested pallet movements, other stock movements and consolidation
Pre-condition	The "Activities" option is selected
Post-Condition	The internal movements of items will be effectively managed
Includes:	None
Extends:	None
Basic Course of Action	1. The administrator/editor activates the "Internal Movements" option 2. The administrator/editor selects the "Manual Stock Movement" option (Alternative Course A: *"Suggested Pallet Movements" option selected,* Alternative Course B*: Other Stock Movements Option selected,* Alternative Course C*: "Consolidation" option selected*) 3. The administrator/editor specifies the storage type, rack, and the old and new locations of the item 4. The item will be assigned a new location and the system displays a confirmation message 5. The use case ends
Alternate Course A *"Suggested Pallet Movements" option selected*	A.2 The system generates list of possible pallet locations A.3 The user selects and confirms the new location A.4 The system displays an acknowledgement message A.5 Use case ends
Alternate Course B *"Other Stock Movements" option selected*	B.2 The system displays other possible new item locations B.3 The user selects and confirms the new location B.4 The system displays an acknowledgement message B.5 Use case ends
Alternate Course C *"Consolidation" option selected*	C.2 The system displays items, source pallet location and destination pallet location C.3The user selects the item, source and destination pallet information and confirms C.4 The system displays an acknowledgment C.3 Use case ends

Use Case Id	UC-10
Use Case Name	Loss/Adjustment
Actors	Administrator/Editor
Description	Used to handle the loss and adjustments
Pre-condition	The "Activities" option is selected
Post-Condition	The loss/adjustment will be effectively handled
Includes:	None
Extends:	None
Basic Course of Action	1. The administrator/editor activates the "Loss/Adjustment" option 2. The administrator/editor selects the "Loss and Adjustment" option (Alternative Course A: *"Quarantine" option selected*) 3. The administrator/editor selects the item and moves to quarantine or adjusts 4. The item will be moved to quarantine or have adjusted value 5. The use case ends
Alternate Course A *"Quarantine" option selected*	A.2 The system displays all quarantined items A.3 The user selects the item and commits the quarantine A.4 The system displays an acknowledgement message A.5 Use case ends

Use Case Id	UC-11
Use Case Name	Pick Face Replenishment
Actors	Administrator/Editor
Description	Used to control the Pick Face
Pre-condition	The “Activities” option is selected
Post-Condition	The Pick Face Replenishment will be effectively managed
Includes:	None
Extends:	None
Basic Course of Action	1. The administrator/editor activates the “Pick Face Replenishment” option 2. The administrator/editor selects the “Pick Face” option (Alternative Course A: *“Pick Face Replenishment List” option selected*) 3. The administrator/editor specifies the item to replenish its pick face 4. The administrator/editor selects the “Replenish” option 5. The pick face will be replenished and the system displays a confirmation message 6. The use case ends
Alternate Course A *“Pick Face Replenishment List” option selected*	A.2 The system displays list of items to be replenished A.3 The administrator/editor selects an item to replenish and selects the “Confirm” option A.4 The pick face will be replenished and the system displays a confirmation message A.5 Use case ends

Use Case Id	UC-12
Use Case Name	Controlling Inventory
Actors	Administrator/Editor
Description	Used to control inventory
Pre-condition	The “Activities” option is selected
Post-Condition	Effective inventory control
Includes:	None
Extends:	None
Basic Course of Action	1. The administrator/editor activates the “Inventory” option 2. The system displays details of items and their inventory information 3. The administrator/editor prints or exports the data 4. The use case ends
Alternate Course A	None

Use Case Id	UC-13
Use Case Name	Handling Activity Logs
Actors	Administrator/Editor
Description	Used to manage the receive transaction, issue transaction, loss/adjustment, and inventory logs
Pre-condition	The administrator/editor has an active account and logged into the system
Post-Condition	Receive transaction, issue transaction, loss/adjustment, and inventory logs will be effectively managed
Includes:	None
Extends:	None
Basic Course of Action	1. The administrator/editor activates the "Activity Log" option 2. The administrator/editor selects the "Receive Transaction Log" option (Alternative Course A: *"Issue Transaction Log" option selected,* Alternative Course B*: "Loss/Adjustment" option selected,* Alternative Course C*: "Inventory Log" option selected*) 3. The administrator/editor specifies the filter options and the date range 4. The system displays the stock report 5. The use case ends
Alternate Course A *"Issue Transaction Log" option selected*	A.2 The administrator/editor specifies the filter options and the date range A.3 The system displays the stock report A.4 The use case ends
Alternate Course A *"Loss/Adjustment" option selected*	B.2 The administrator/editor specifies the filter options and the date range B.3 The system displays the stock report B.4 The use case ends
Alternate Course A *"Inventory Log" option selected*	C.2 The administrator/editor specifies the filter options C.3 The system displays the stock report C.4 The use case ends

Use Case Id	UC-14
Use Case Name	Generating Reports
Actors	Editor/Reader
Description	Allows the editor/reader to generate different kinds of reports
Pre-condition	The editor/reader has an active account and logged into the system
Post-Condition	Different kinds of reports will be generated
Includes:	None
Extends:	None
Basic Course of Action	1. The editor/reader activates the "Reports" option 2. The editor/reader selects the options "Stock Status", "Over Stock Items", "Stock Out Items", "Issues By Receiving Unit", "Expired Products", "Near Expiry", or "Storage Reports" option 3. The editor/reader specifies the filtering criteria 4. The system generates the respective report 5. The use case ends

Use Case Id	UC-15
Use Case Name	Log Out
Actors	Administrator, Editor, Reader
Description	Allows to log out of the HCMIS system
Pre-condition	The user has to be already logged in
Post-Condition	The HCMIS main screen exists and the login screen will be displayed
Includes:	None
Extends:	None
Basic Course of Action	1. The user activates the “File” menu and selects “Log Out” 2. The system terminates the HCMIS main screen 3. The system displays the login screen 4. The use case ends

3.2.2. Class Diagram

A class diagram is useful to capture the structure (attributes and associations) and behavior (method signatures) of a system. Moreover, a class diagram facilitates the abstractions of attributes and behavior of a set of objects. Thus, the class diagram in Figure 3-2 depicts the attributes, methods and associations of the classes in the HCMIS system.

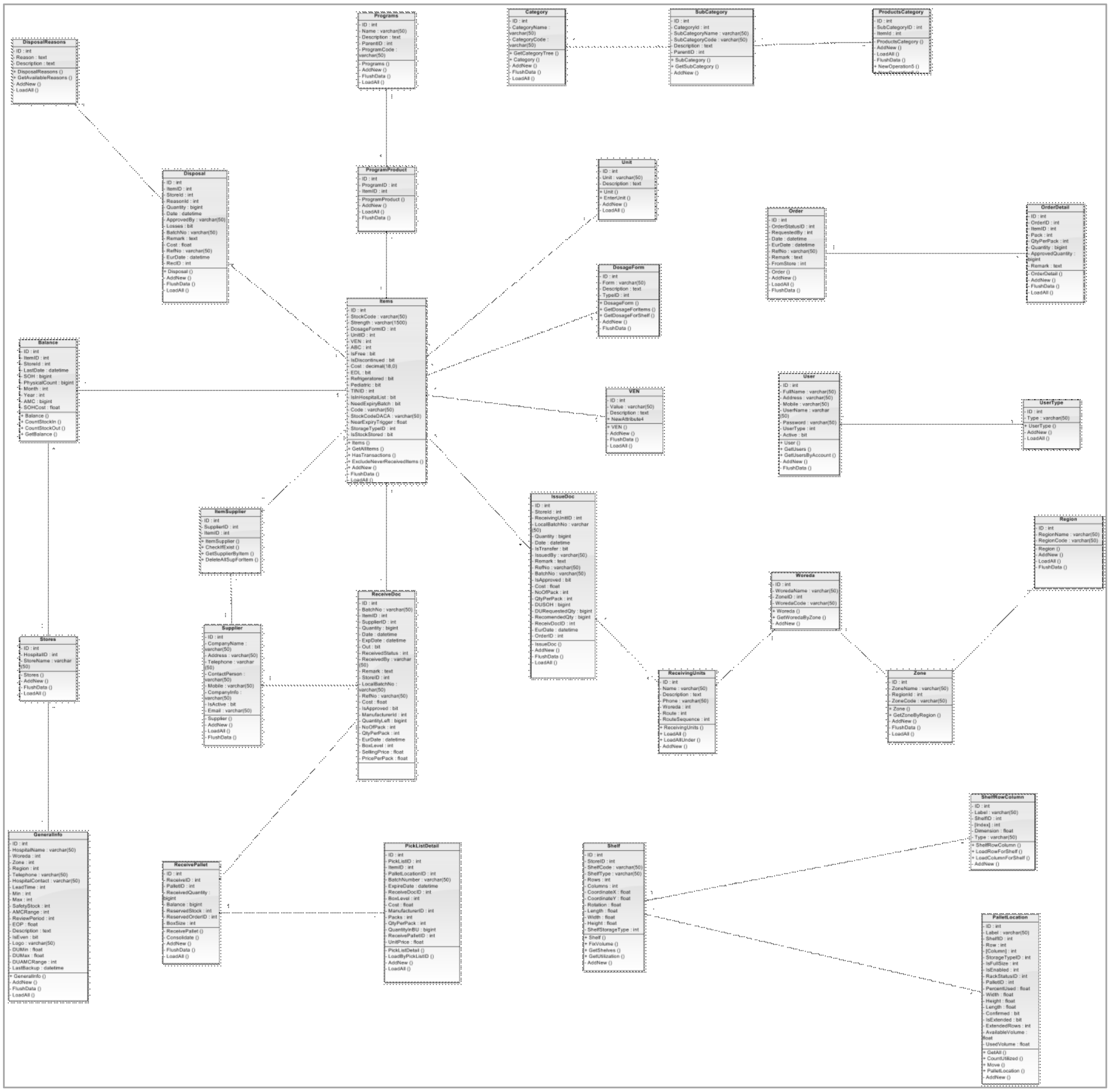


Figure 3-2: The class diagram of the HCMIS System

4 | SYSTEM DESIGN

Sections

This chapter deals with the following issues:

- ❒ The Assumptions, Dependencies and General Constraints of Designing the HCMIS System
- ❒ The System Environment
- ❒ The Architecture of the HCMIS System and its Design Patterns
- ❒ Deployment Management of the HCMIS System which includes both the Application Deployment and Database Deployment
- ❒ Persistent Data Management (Mapping)

In section 3 of this document the functional and non-functional requirements of the HCMIS system were briefly discussed. Furthermore, the functional requirements were identified and grouped based on their common purpose and functionality as use cases. Then after, a use case diagram was drawn to envisage the view and interaction of users with the system. Moreover, the pictorial representation of the detailed static view of the HCMIS system was designated by a class diagram.

In this section of the document, the design issues of the HCMIS system will be dealt with. Therefore, this section gives insight about the assumptions, dependencies and general constraints, system environment, system architecture, subsystem decomposition, deployment management, and persistent data management.

4.1. Assumptions, Dependencies and General Constraints

The HCMIS system is a graphical application software that requires users to have a computer with the right hardware and software components. Users are also required to have basic computer skills and familiarity with Graphical User Interfaces (GUI). Besides, the HCMIS system demands Microsoft .NET Framework 3.5 and Microsoft SQL Server later than 2005 to be properly installed and configured.

In addition to this, the HCMIS system is designed to be user friendly, reliable, error-free and secure. Therefore, it can be used by users with various backgrounds ranging from those with limited computing experience to advanced users and experts. Besides, the HCMIS system's code base shall be comprehensively commented, conventions explained, and ambiguities noted to ease the task of maintenance for future developers.

4.2. System Environment

The HCMIS system is designed to be compatible virtually with all Windows Operating Systems (OS) later than Windows XP. Besides, the following software packages are used for the successful design and implementation of the system:

- □ *Development*: Visual Studio 2010+DevExpress
- □ *Database*: Microsoft SQL Server Express Edition
- □ *Diagrams*: Software Ideas Modeler 3.30

☐ *Running Environment:*.Net Framework 3.5

4.3. System Architecture

The other significant issue to consider in the design of the system is the application (system) architecture. The architecture plays a prominent role in assisting the design and deployment of the HCMIS system by breaking up the application (based on functionalities and components) into layers (tiers). Separating the components of the system into different layers provides a model to create a flexible and reusable application. Besides, modification and/or addition of functionalities to a specific layer can be best managed rather than rewriting the entire application all over again.

As most Windows-based systems, the HCMIS system will follow two-layer architecture; the application and data layers.

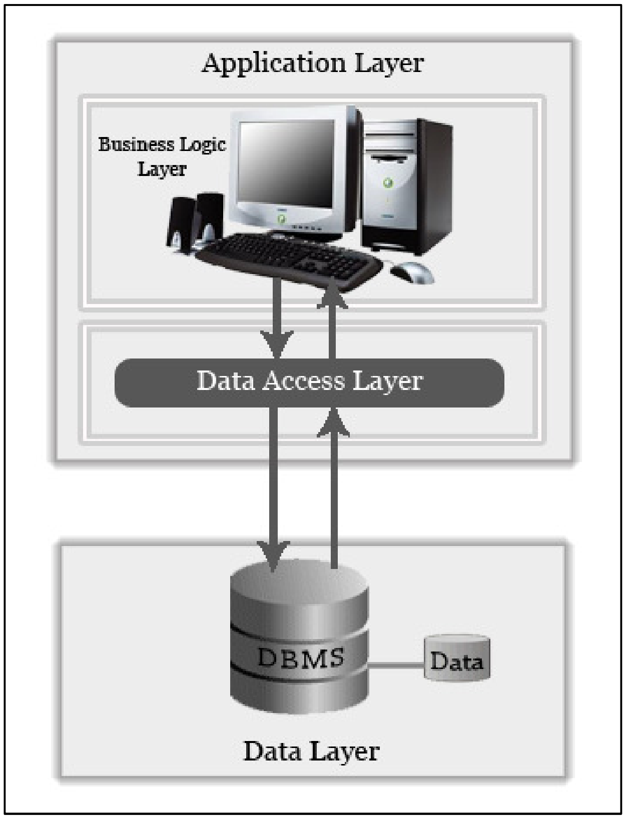


Figure 4-1: The Architecture of the HCMIS System

- □ APPLICATION (BUSINESS LOGIC) LAYER

This layer contains the working functionalities of the HCMIS system. All the business logics including the processing of information exchange between the database management system (DBMS) and the user interfaces. The Data Access sub layer provides handy option to establish database connection and management.

- □ DATA LAYER

The data layer of the HCMIS system architecture represents all the system data and the database server. Hence, this layer is responsible for managing the system data as well as the Microsoft SQL Server by providing the best option to separate the data and the application (business) logic.

1.1. Design Patterns

The system analysis as well as design of the HCMIS is handled by the Object Oriented (OO) system development approach using two-layer system architecture.

4.4. Deployment Management

As indicated in the previous sections, the HCMIS system is designed to be deployed on Windows OS platform (i.e. later than XP) with the Microsoft .NET Framework 3.5 and SQL Server available. The deployment process includes both the application as well as database deployments.

4.4.1. Application Deployment

Among other things, the HCMIS software is designed and developed so as to facilitate the deployment process. As a result, anyone with basic software installation skill can deploy the system. Users are only required to run the setup file from the CD-ROM where the HCMIS software is packaged and follow the screen instructions. Once the software installation is complete, the next step is to copy the database (which is found on the same CD-ROM) to the local disk. These procedures work for both the Warehouse (hub) and Facility editions.

4.4.2. Database Deployment

One of the software packages that are necessary for the HCMIS system to be operational is the Microsoft SQL Server. The SQL Server DBMS should be installed and properly configured for the

HCMIS system to work. For security purpose, appropriate user name and password should be provided to the database server (i.e. Microsoft SQL Server) and the database attached to the SQL Server.

Figure 4-2 shows the deployment of the application and data layers of the HCMIS system.

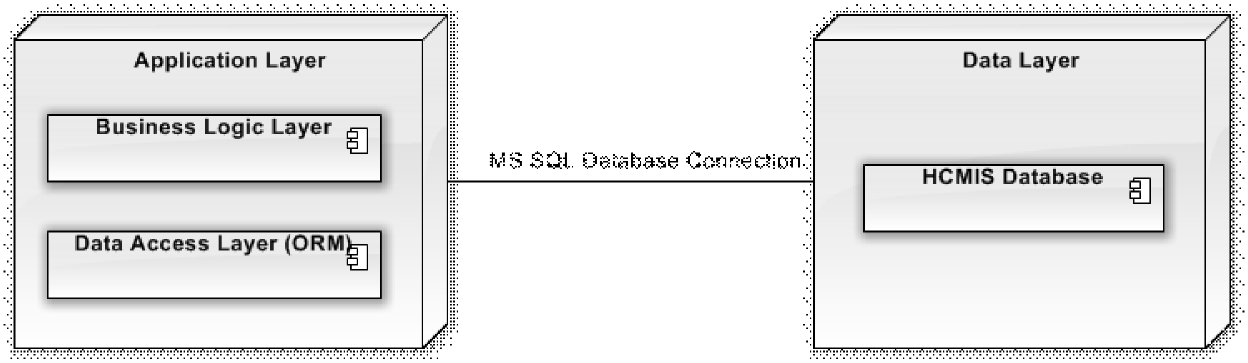


Figure 4-2: The Deployment Diagram of the HCMIS System

4.5. Persistent Data Management (Mapping)

This section gives insight about persistent data management of the HCMIS system. In section 3, potential classes of the HCMIS system were identified altogether their relationships. The classes hold attributes as well as methods. The attributes represent the behavior (structure) whereas the methods are the operations to be carried out.

The classes, therefore, should be converted into relations (tables) so that they can be represented in the relational DBMS (i.e. Microsoft SQL Server). Furthermore, the attributes of the classes are mapped to table fields and the relationships between the classes will be maintained between the tables.

Figure 4-3 depicts the Entity Relationship Diagram (ERD) of the HCMIS system.

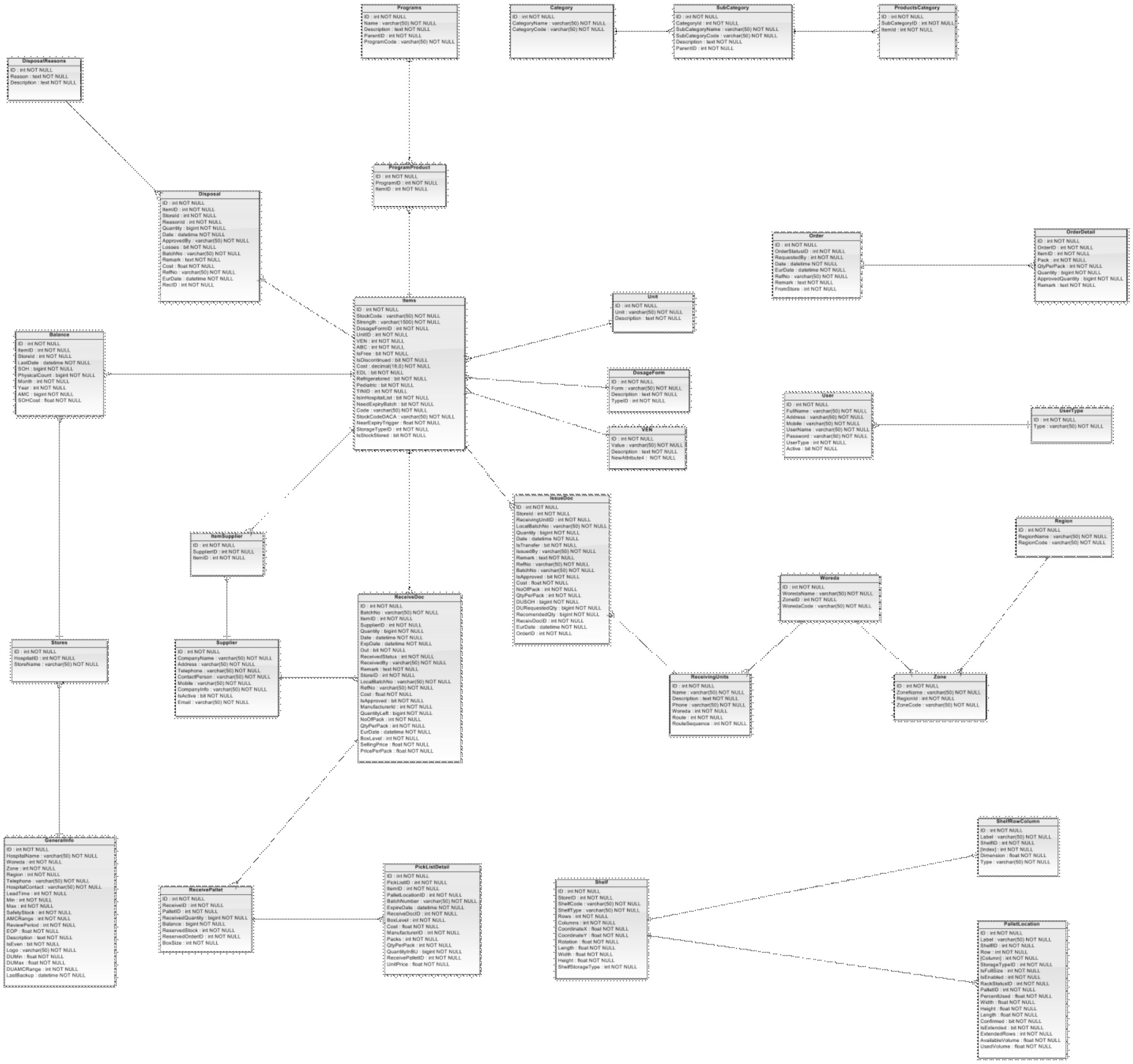

Programs
ID : int NOT NULL
Name : varchar(50) NOT NULL
Description : text NOT NULL
ParentID : int NOT NULL
ProgramCode : varchar(50) NOT NULL
Category
ID : int NOT NULL
CategoryName : varchar(50) NOT NULL
CategoryCode : varchar(50) NOT NULL
SubCategory
ID : int NOT NULL
CategoryId : int NOT NULL
SubCategoryName : varchar(50) NOT NULL
SubCategoryCode : varchar(50) NOT NULL
Description : text NOT NULL
ParentID : int NOT NULL
ProductsCategory
ID : int NOT NULL
SubCategoryID : int NOT NULL
ItemId : int NOT NULL
DisposalReasons
ID : int NOT NULL
Reason : text NOT NULL
Description : text NOT NULL
ProgramProduct
ID : int NOT NULL
ProgramID : int NOT NULL
ItemID : int NOT NULL
Disposal
ID : int NOT NULL
ItemID : int NOT NULL
StoreId : int NOT NULL
ReasonId : int NOT NULL
Quantity : bigint NOT NULL
Date : datetime NOT NULL
ApprovedBy : varchar(50) NOT NULL
Losses : bit NOT NULL
BatchNo : varchar(50) NOT NULL
Remark : text NOT NULL
Cost : float NOT NULL
RefNo : varchar(50) NOT NULL
EurDate : datetime NOT NULL
RecID : int NOT NULL
Order
ID : int NOT NULL
OrderStatusID : int NOT NULL
RequestedBy : int NOT NULL
Date : datetime NOT NULL
EurDate : datetime NOT NULL
RefNo : varchar(50) NOT NULL
Remark : text NOT NULL
FromStore : int NOT NULL
OrderDetail
ID : int NOT NULL
OrderID : int NOT NULL
ItemID : int NOT NULL
Pack : int NOT NULL
QtyPerPack : int NOT NULL
Quantity : bigint NOT NULL
ApprovedQuantity : bigint NOT NULL
Remark : text NOT NULL
Balance
ID : int NOT NULL
ItemID : int NOT NULL
StoreId : int NOT NULL
LastDate : datetime NOT NULL
SOH : bigint NOT NULL
PhysicalCount : bigint NOT NULL
Month : int NOT NULL
Year : int NOT NULL
AMC : bigint NOT NULL
SOHCost : float NOT NULL
Items
ID : int NOT NULL
StockCode : varchar(50) NOT NULL
Strength : varchar(1500) NOT NULL
DosageFormID : int NOT NULL
UnitID : int NOT NULL
VEN : int NOT NULL
ABC : int NOT NULL
IsFree : bit NOT NULL
IsDiscontinued : bit NOT NULL
Cost : decimal(18,0) NOT NULL
EDL : bit NOT NULL
Refrigeratored : bit NOT NULL
Pediatric : bit NOT NULL
TINID : int NOT NULL
IsInHospitalList : bit NOT NULL
NeedExpiryBatch : bit NOT NULL
Code : varchar(50) NOT NULL
StockCodeDACA : varchar(50) NOT NULL
NearExpiryTrigger : float NOT NULL
StorageTypeID : int NOT NULL
IsStockStored : bit NOT NULL
Unit
ID : int NOT NULL
Unit : varchar(50) NOT NULL
Description : text NOT NULL
DosageForm
ID : int NOT NULL
Form : varchar(50) NOT NULL
Description : text NOT NULL
TypeID : int NOT NULL
User
ID : int NOT NULL
FullName : varchar(50) NOT NULL
Address : varchar(50) NOT NULL
Mobile : varchar(50) NOT NULL
UserName : varchar(50) NOT NULL
Password : varchar(50) NOT NULL
UserType : int NOT NULL
Active : bit NOT NULL
UserType
ID : int NOT NULL
Type : varchar(50) NOT NULL
VEN
ID : int NOT NULL
Value : varchar(50) NOT NULL
Description : text NOT NULL
NewAttribute4 : NOT NULL
IssueDoc
ID : int NOT NULL
StoreId : int NOT NULL
ReceivingUnitID : int NOT NULL
LocalBatchNo : varchar(50) NOT NULL
Quantity : bigint NOT NULL
Date : datetime NOT NULL
IsTransfer : bit NOT NULL
IssuedBy : varchar(50) NOT NULL
Remark : text NOT NULL
RefNo : varchar(50) NOT NULL
BatchNo : varchar(50) NOT NULL
IsApproved : bit NOT NULL
Cost : float NOT NULL
NoOfPack : int NOT NULL
QtyPerPack : int NOT NULL
DUSOH : bigint NOT NULL
DURequestedQty : bigint NOT NULL
RecomendedQty : bigint NOT NULL
ReceivDocID : int NOT NULL
EurDate : datetime NOT NULL
OrderID : int NOT NULL
Region
ID : int NOT NULL
RegionName : varchar(50) NOT NULL
RegionCode : varchar(50) NOT NULL
ItemSupplier
ID : int NOT NULL
SupplierID : int NOT NULL
ItemID : int NOT NULL
Woreda
ID : int NOT NULL
WoredaName : varchar(50) NOT NULL
ZoneID : int NOT NULL
WoredaCode : varchar(50) NOT NULL
Stores
ID : int NOT NULL
HospitalID : int NOT NULL
StoreName : varchar(50) NOT NULL
Supplier
ID : int NOT NULL
CompanyName : varchar(50) NOT NULL
Address : varchar(50) NOT NULL
Telephone : varchar(50) NOT NULL
ContactPerson : varchar(50) NOT NULL
Mobile : varchar(50) NOT NULL
CompanyInfo : varchar(50) NOT NULL
IsActive : bit NOT NULL
Email : varchar(50) NOT NULL
ReceiveDoc
ID : int NOT NULL
BatchNo : varchar(50) NOT NULL
ItemID : int NOT NULL
SupplierID : int NOT NULL
Quantity : bigint NOT NULL
Date : datetime NOT NULL
ExpDate : datetime NOT NULL
Out : bit NOT NULL
ReceivedStatus : int NOT NULL
ReceivedBy : varchar(50) NOT NULL
Remark : text NOT NULL
StoreID : int NOT NULL
LocalBatchNo : varchar(50) NOT NULL
RefNo : varchar(50) NOT NULL
Cost : float NOT NULL
IsApproved : bit NOT NULL
ManufacturerId : int NOT NULL
QuantityLeft : bigint NOT NULL
NoOfPack : int NOT NULL
QtyPerPack : int NOT NULL
EurDate : datetime NOT NULL
BoxLevel : int NOT NULL
SellingPrice : float NOT NULL
PricePerPack : float NOT NULL
ReceivingUnits
ID : int NOT NULL
Name : varchar(50) NOT NULL
Description : text NOT NULL
Phone : varchar(50) NOT NULL
Woreda : int NOT NULL
Route : int NOT NULL
RouteSequence : int NOT NULL
Zone
ID : int NOT NULL
ZoneName : varchar(50) NOT NULL
RegionId : int NOT NULL
ZoneCode : varchar(50) NOT NULL
ShelfRowColumn
ID : int NOT NULL
Label : varchar(50) NOT NULL
ShelfID : int NOT NULL
[Index] : int NOT NULL
Dimension : float NOT NULL
Type : varchar(50) NOT NULL
GeneralInfo
ID : int NOT NULL
HospitalName : varchar(50) NOT NULL
Woreda : int NOT NULL
Zone : int NOT NULL
Region : int NOT NULL
Telephone : varchar(50) NOT NULL
HospitalContact : varchar(50) NOT NULL
LeadTime : int NOT NULL
Min : int NOT NULL
Max : int NOT NULL
SafetyStock : int NOT NULL
AMCRange : int NOT NULL
ReviewPeriod : int NOT NULL
EOP : float NOT NULL
Description : text NOT NULL
IsEven : bit NOT NULL
Logo : varchar(50) NOT NULL
DUMin : float NOT NULL
DUMax : float NOT NULL
DUAMCRange : int NOT NULL
LastBackup : datetime NOT NULL
ReceivePallet
ID : int NOT NULL
ReceiveID : int NOT NULL
PalletID : int NOT NULL
ReceivedQuantity : bigint NOT NULL
Balance : bigint NOT NULL
ReservedStock : int NOT NULL
ReservedOrderID : int NOT NULL
BoxSize : int NOT NULL
PickListDetail
ID : int NOT NULL
PickListID : int NOT NULL
ItemID : int NOT NULL
PalletLocationID : int NOT NULL
BatchNumber : varchar(50) NOT NULL
ExpireDate : datetime NOT NULL
ReceiveDocID : int NOT NULL
BoxLevel : int NOT NULL
Cost : float NOT NULL
ManufacturerID : int NOT NULL
Packs : int NOT NULL
QtyPerPack : int NOT NULL
QuantityInBU : bigint NOT NULL
ReceivePalletID : int NOT NULL
UnitPrice : float NOT NULL
Shelf
ID : int NOT NULL
StoreID : int NOT NULL
ShelfCode : varchar(50) NOT NULL
ShelfType : varchar(50) NOT NULL
Rows : int NOT NULL
Columns : int NOT NULL
CoordinateX : float NOT NULL
CoordinateY : float NOT NULL
Rotation : float NOT NULL
Length : float NOT NULL
Width : float NOT NULL
Height : float NOT NULL
ShelfStorageType : int NOT NULL
PalletLocation
ID : int NOT NULL
Label : varchar(50) NOT NULL
ShelfID : int NOT NULL
Row : int NOT NULL
[Column] : int NOT NULL
StorageTypeID : int NOT NULL
IsFullSize : int NOT NULL
IsEnabled : int NOT NULL
RackStatusID : int NOT NULL
PalletID : int NOT NULL
PercentUsed : float NOT NULL
Width : float NOT NULL
Height : float NOT NULL
Length : float NOT NULL
Confirmed : bit NOT NULL
IsExtended : bit NOT NULL
ExtendedRows : int NOT NULL
AvailableVolume : float NOT NULL
UsedVolume : float NOT NULL

Figure 4-3: Entity Relationship Diagram of the HCMIS System

A detailed description of the tables involved in the entity relationship diagram of the HCMIS System is provided in Appendix A.

5 | THE HCMIS SOFTWARE

Objectives

This Chapter deals with following issues:

- ❒ The development tools used for HCMIS System including the programming tools and IDE
- ❒ A brief description about the data access layer
- ❒ The DBMS that is put in place for the HCMIS software
- ❒ Other tools used (such as Software Idea Modeler)
- ❒ The description of the HCMIS program modules
- ❒ A brief description of the user interface Prototype with screen components for both the Warehouse (Hub) and Facility editions
- ❒ Installation guidelines and steps for the Warehouse (Hub) and Facility editions

This section of the document provides detailed description of the development tools that are put in place to implement the HCMIS system together with the user interface prototypes.

5.1. Development Tools

The development tools that are used to realize the HCMIS system include programming tools, Integrated Development Environment (IDE), Data Access Layer, and the Database Management System (DBM). The forthcoming sections will provide the details.

□ PROGRAMMING TOOLS AND IDE

The HCMIS system is designed in such a way that it can provide features that can help manage warehouses (hubs) as well as facilities. For this reason, the system has the facility as well as hub (warehouse) editions. These versions share some common features that include managing transactions, activity logs, and efficiently handling the settings of hubs and/or facilities. Besides, the system offers option to back-up the database and the system administrator with greater alternatives to manage the system.

As indicated in the previous sections, the HCMIS system is designed to run on the Windows OS environment. To this end, the Microsoft Visual Studio 2010 is used as an Integrated Development Environment (IDE). The IDE offers a conducive programming environment to develop the HCMIS system. Besides its popularity and handy feature for object oriented system development paradigm, the C# programming language altogether the DevExpress development tool is put in place so as to make the development process easy, integrated, effective and efficient.

□ THE DATA ACCESS LAYER

The data access layer of the HCMIS system is handled by the open source Object Relational Management (ORM) system - MyGeneration. The software package provides a flexible way of establishing connection with the DBMS and manipulating the database.

□ THE DATABASE MANAGEMENT SYSTEM (DBMS)

For the successful implementation of the design of the database depicted in section 4 of this document, appropriate, cost effective and convenient DBMS is required. For this reason, Microsoft SQL Server Express edition is selected and used as a database storage and management system. Being a free edition, Microsoft SQL Server Express, offers great features that are vital for the HCMIS database. These features may include flexibility to be integrated with the Visual Studio 2010 IDE and the data access layer.

- □ OTHER TOOLS

In addition to the development tools specified, other software packages are also used for documentation and modeling the HCMIS system. The use case, class and database diagrams are modeled using the Software Idea Modeler 3.30 software package. This software package is also applied to design the architecture and deployment diagrams of the system. For the purpose of creating documents and preparing templates, Microsoft Office Word 2010 is put in place.

5.2. PROGRAM MODULES

This section of the document lists out administrative files used to help maintenance of the Warehouse (Hub) and/or Facility versions of HCMIS System. The list includes tables to be updated, tables from which data is to be read and the purpose of the maintenance module. Besides, the list provides information regarding the version of the HCMIS System the module applies to.

Receive Item Module	
Update Table(s)	ReceiveDoc, ReceivePallet, Pallet, PalletLocation
Read-Only Table(s)	Item, Supplier, ItemPreferredLocation, ItemManufacturer
Applies To	Both Facility and Warehouse (Hub)
Purpose	This module is used to enter Item information into the system

Confirm Putaway Module	
Update Table(s)	**PalletLocation**
Read-Only Table(s)	Item, ReceiveDoc
Applies To	Warehouse (Hub)
Purpose	This module is applied to confirm the put away of received items

CDR Request Module	
Update Table(s)	Order, OrderDetail
Read-Only Table(s)	Item, Category, ReceivingUnit, Store
Applies To	Warehouse (Hub)
Purpose	This module is used to handle issue requests from the facilities

Hub Approval Module	
Update Table(s)	Order, OrderDetail
Read-Only Table(s)	ReceiveDoc, Item, ReceivingUnit
Applies To	Warehouse (Hub)
Purpose	This module is used to check stock status and decide the amount to issue

Generating Pick List Module	
Update Table(s)	Order, PickList, PickListDetail, ReceivePallet
Read-Only Table(s)	Item, ReceiveDoc, ItemManufacturer, PalletLocation, Pallet, PickFace, Manufacturer
Applies To	Both Facility and Warehouse (Hub)
Purpose	This module is applied to generate the pick list of approved quantities to be issued

Confirm Picking/Issue Module	
Update Table(s)	Order, OrderDetail, Issue, PickList, PickListDetail, ReceiveDoc, ReceivePallet, PalletLocation
Read-Only Table(s)	Item, ItemManufacturer, Manufacturer
Applies To	Both Facility and Warehouse (Hub)
Purpose	This module is used to enter the actual values of items issued

Internal Stock Movement Module	
Update Table(s)	PalletLocation
Read-Only Table(s)	Item, Pallet, Shelf, StorageTypes
Applies To	Warehouse (Hub)
Purpose	This module is used to manage the movement of items in the warehouse

Pallet Consolidation Module	
Update Table(s)	Pallet, PalletLocation, RecevePallet
Read-Only Table(s)	Item, ReceiveDoc
Applies To	Warehouse (Hub)
Purpose	This module is applied to control the arrangement of items on a pallet so that space can be efficiently managed.

Pick Face Replenishment Module	
Update Table(s)	PickFace, ReceivePallet
Read-Only Table(s)	ReceiveDoc, Item, PalletLocation, Store
Applies To	Both Facility and Warehouse (Hub)
Purpose	This module is used to replenish the pick face

Loss and Adjustment Module	
Update Table(s)	Disposal, ReceiveDoc
Read-Only Table(s)	Item, ReceiveDoc, IssueDoc
Applies To	Both Facility and Warehouse (Hub)
Purpose	This module is put in place to efficiently manage errors in data entry

Inventory Module	
Update Table(s)	ReceiveDoc, YearEnd
Read-Only Table(s)	ReceiveDoc, IssueDoc
Applies To	Both Facility and Warehouse (Hub)
Purpose	This module is applied to control inventory in the facilities and/or warehouses (hubs)

Stock Status Report Module	
Update Table(s)	None
Read-Only Table(s)	Category, Store, Item, ReceiveDoc, IssueDoc, Program, ReceivingUnit, Disposal, YearEnd, GeneralInfo
Applies To	Both Facility and Warehouse (Hub)
Purpose	This module is used to generate the Stock Status report

Over Stock Report Module	
Update Table(s)	None
Read-Only Table(s)	Category, Store, Item, ReceiveDoc, IssueDoc, Program, ReceivingUnit, Disposal, YearEnd, GeneralInfo
Applies To	Both Facility and Warehouse (Hub)
Purpose	This module is used to generate the Over Stock report

Stock Out Report Module	
Update Table(s)	None
Read-Only Table(s)	Category, Store, Item, ReceiveDoc, IssueDoc, Program, ReceivingUnit, Disposal, YearEnd, GeneralInfo
Applies To	Both Facility and Warehouse (Hub)
Purpose	This module is used to generate the Stock Out report

Issues By Receiving Unit Report Module	
Update Table(s)	None
Read-Only Table(s)	Category, Store, Item, ReceiveDoc, IssueDoc, Program, ReceivingUnit, Disposal, YearEnd, GeneralInfo
Applies To	Both Facility and Warehouse (Hub)
Purpose	This module is used to generate the Issues By the Receiving Unit report

Expire Products Report Module	
Update Table(s)	None
Read-Only Table(s)	Category, Store, Item, ReceiveDoc, IssueDoc, Program, ReceivingUnit, Disposal, YearEnd, GeneralInfo
Applies To	Both Facility and Warehouse (Hub)
Purpose	This module is used to generate the Expire Products report

Near Expiry Products Report Module	
Update Table(s)	None
Read-Only Table(s)	Category, Store, Item, ReceiveDoc, IssueDoc, Program, ReceivingUnit, Disposal, YearEnd, GeneralInfo
Applies To	Both Facility and Warehouse (Hub)
Purpose	This module is generates Near Expiry Products report

Summary Report Module	
Update Table(s)	None
Read-Only Table(s)	Category, Store, Item, ReceiveDoc, IssueDoc, Program, ReceivingUnit, Disposal, YearEnd, GeneralInfo
Applies To	Both Facility and Warehouse (Hub)
Purpose	This module is used to generate the Summary report

Storage Report Module	
Update Table(s)	None
Read-Only Table(s)	StorageType, Store, Shelf, PalletLocation, Pallet, ReceivePallet, ReceiveDoc, ShelfRowColumn
Applies To	Warehouse (Hub)
Purpose	This module is used to generate the Storage report

Activity Log Module	
Update Table(s)	ReceiveDoc, IssueDoc
Read-Only Table(s)	Item, IssueDoc, Store, ReceivingUnit, YearEnd
Applies To	Both Facility and Warehouse (Hub)
Purpose	This module is applied to manage receive, issue, loss and adjustment and inventory logs

Hub/Facility Setting Module	
Update Table(s)	GeneralInfo, Supplier, Store, Route, ReceivingUnit
Read-Only Table(s)	None
Applies To	Both Facility and Warehouse (Hub)
Purpose	This module is applied to manage the Hu/Facility settings

Customize Drug List Module	
Update Table(s)	Item, PreferredPalletLocation, PickFace, ItemManufacturer
Read-Only Table(s)	Category, Item, ItemManufacturer, Program, StorageType, ABC, VEN, PickFace, PreferredPalletLocation, Manufacturer
Applies To	Both Facility and Warehouse (Hub)
Purpose	This module is used to enter the prefer preferred location to put items in the facilities and/or warehouses (hubs)

Warehouse Setting Module	
Update Table(s)	Shelf, PhysicalStore, PalletLocation, ShelfRowColumn, ItemPreferredLocation
Read-Only Table(s)	None
Applies To	Warehouse (Hub)
Purpose	This module is used to manage the warehouse settings

5.3. User Interface Prototype

For a software system, the user interface plays a principal role in helping users visualize the system and provide them the options to interact with it. Taking this fact into account, due consideration is given to designing easy to use, friendly and highly integrated user interfaces for the HCMIS system. Besides, the user interfaces are designed in such a way that they can entertain future expansion and modification of the system.

As stated in the previous sections, the HCMIS system is composed of Facility and Warehouse (Hub) editions. The description of the user interface prototypes, therefore, will be for both the Facility as well as Warehouse (Hub) editions.

- □ USER INTERFACE COMPONENTS

The Facility and Warehouse (Hub) editions of the HCMIS system share a great extent of user interface components and features. These user interface components generally include screen layouts, menu options and many other features. However, in specific terms, the login and main screens, dialog boxes and other screen components of the Facility and Warehouse (Hub) editions share similar features.

When users try to start the HCMIS system, the login screen will be displayed prompting them to enter user name and password. The system grants access if the credential of users is valid and currently active. Figure 5-1 shows the login screen of the HCMIS system.

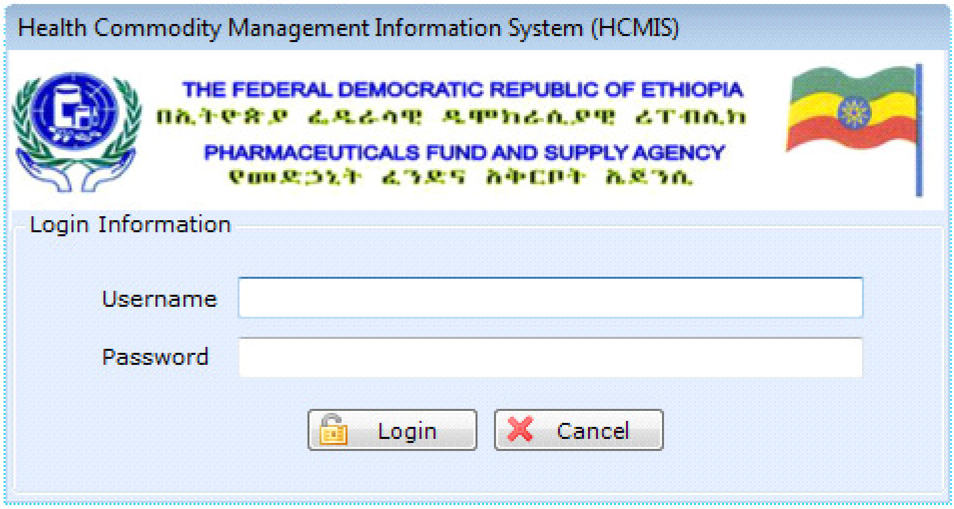
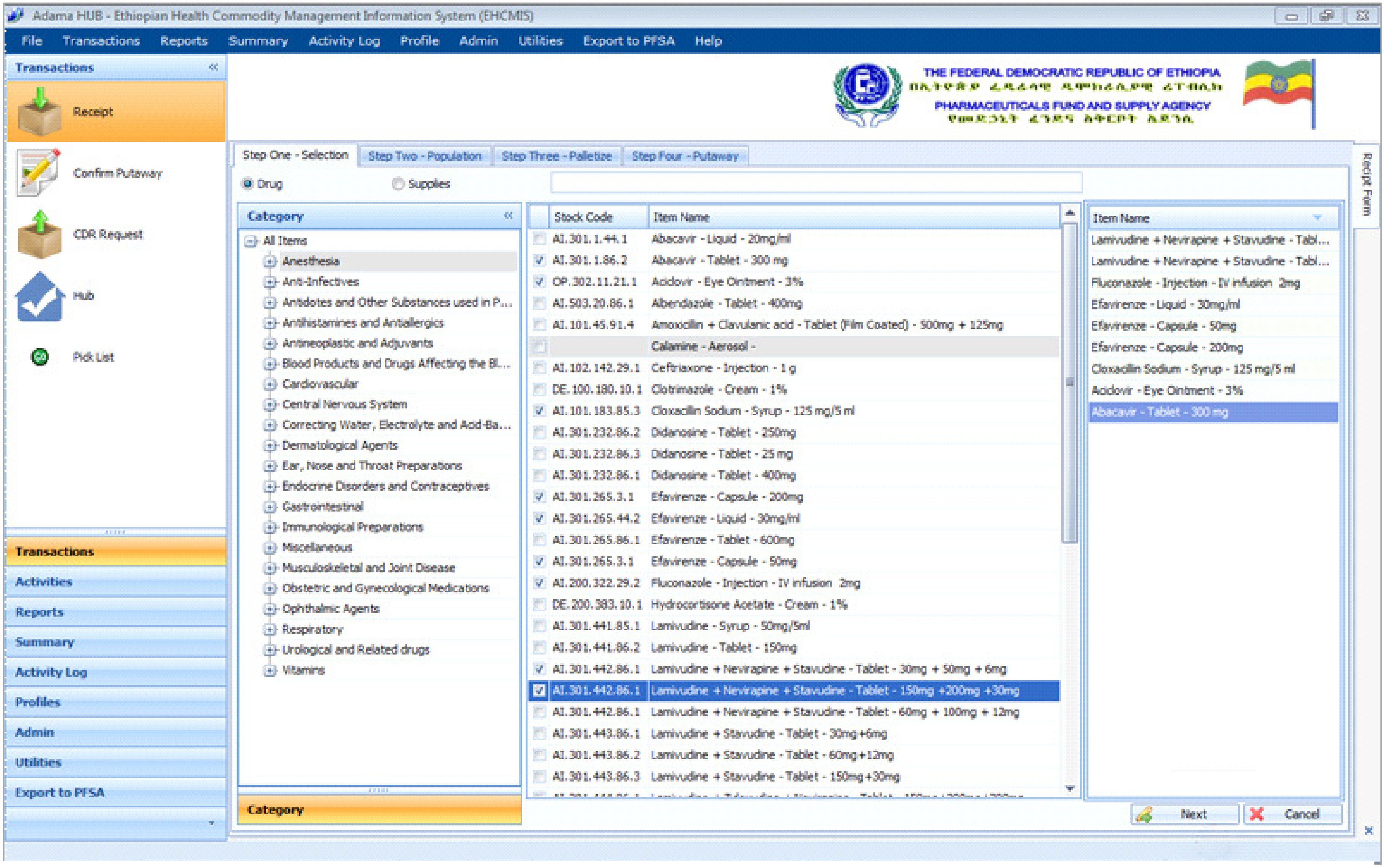

Health Commodity Management Information System (HCMIS)
THE FEDERAL DEMOCRATIC REPUBLIC OF ETHIOPIA
በኢትዮጵያ ፌዴራላዊ ዲሞክራሲያዊ ሪፐብሊክ
PHARMACEUTICALS FUND AND SUPPLY AGENCY
የመድኃኒት ፈንድና አቅርቦት ኤጀንሲ
Login Information
Username
Password
Login
Cancel

Figure 5-1: The Login Screen

Immediately after the user passes the security check, the main screen (window) of the system will be displayed. This screen contains different features for the Warehouse (Hub) and Facility editions. The main screen for the Warehouse (Hub) edition provides additional menus such as Activities, and features that are not available in the Facility edition. Moreover, the sub menus in the main screen also vary with respect to the software editions. Figure 5-2 and Figure 5-3 show the main screen for the warehouse (hub) and facility versions respectively.



● Drug ○ Supplies

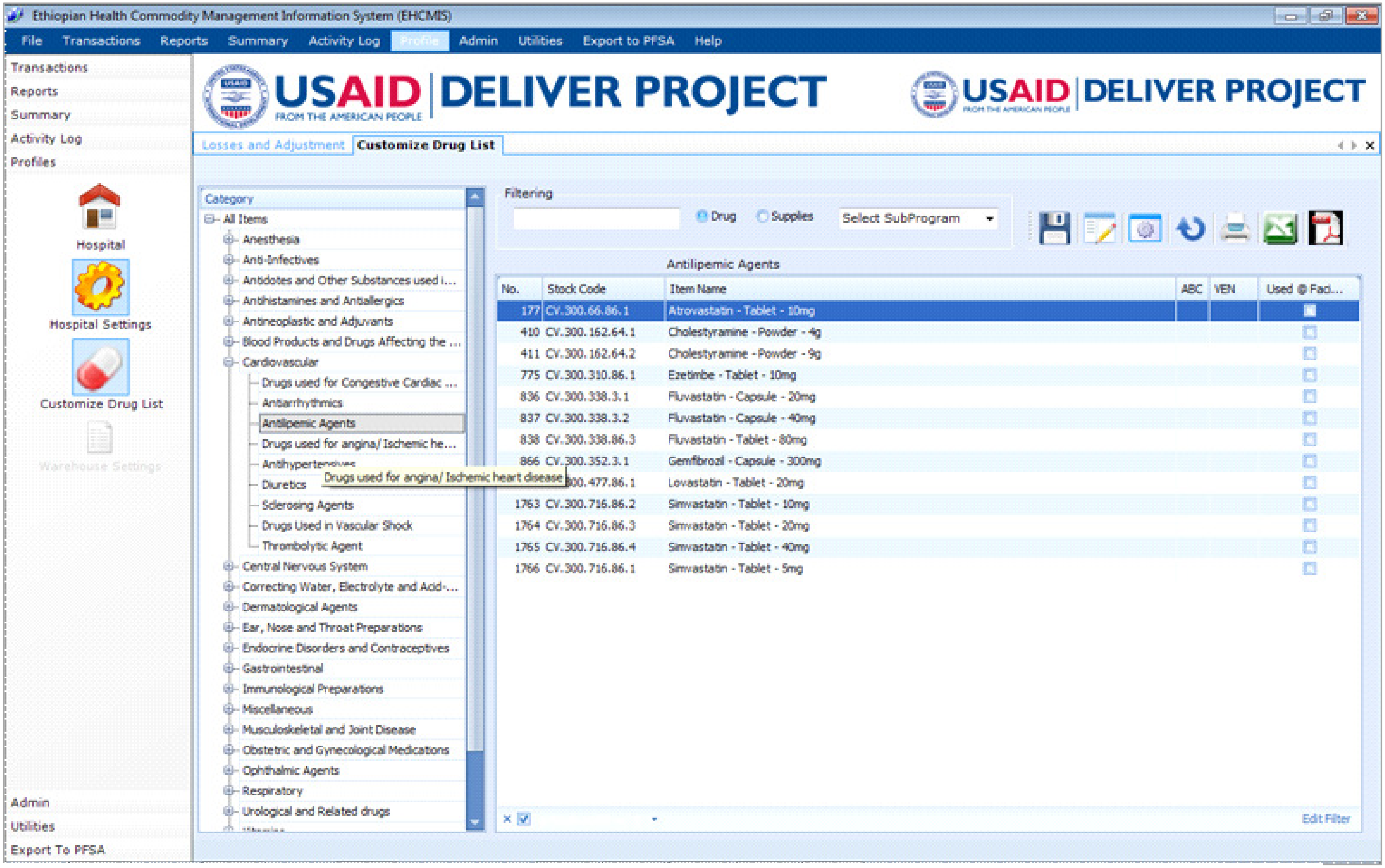
Category

- All Items
  - Anesthesia
  - Anti-Infectives
  - Antidotes and Other Substances used in P...
  - Antihistamines and Antiallergics
  - Antineoplastic and Adjuvants
  - Blood Products and Drugs Affecting the Bl...
  - Cardiovascular
  - Central Nervous System
  - Correcting Water, Electrolyte and Acid-Ba...
  - Dermatological Agents
  - Ear, Nose and Throat Preparations
  - Endocrine Disorders and Contraceptives
  - Gastrointestinal
  - Immunological Preparations
  - Miscellaneous
  - Musculoskeletal and Joint Disease
  - Obstetric and Gynecological Medications
  - Ophthalmic Agents
  - Respiratory
  - Urological and Related drugs
  - Vitamins

	Stock Code	Item Name
□	AI.301.1.44.1	Abacavir - Liquid - 20mg/ml
☑	AI.301.1.86.2	Abacavir - Tablet - 300 mg
☑	OP.302.11.21.1	Aciclovir - Eye Ointment - 3%
□	AI.503.20.86.1	Albendazole - Tablet - 400mg
□	AI.101.45.91.4	Amoxicillin + Clavulanic acid - Tablet (Film Coated) - 500mg + 125mg
□		Calamine - Aerosol -
□	AI.102.142.29.1	Ceftriaxone - Injection - 1 g
□	DE.100.180.10.1	Clotrimazole - Cream - 1%
☑	AI.101.183.85.3	Cloxacillin Sodium - Syrup - 125 mg/5 ml
□	AI.301.232.86.2	Didanosine - Tablet - 250mg
□	AI.301.232.86.3	Didanosine - Tablet - 25 mg
□	AI.301.232.86.1	Didanosine - Tablet - 400mg
☑	AI.301.265.3.1	Efavirenze - Capsule - 200mg
☑	AI.301.265.44.2	Efavirenze - Liquid - 30mg/ml
□	AI.301.265.86.1	Efavirenze - Tablet - 600mg
☑	AI.301.265.3.1	Efavirenze - Capsule - 50mg
☑	AI.200.322.29.2	Fluconazole - Injection - IV infusion 2mg
□	DE.200.383.10.1	Hydrocortisone Acetate - Cream - 1%
□	AI.301.441.85.1	Lamivudine - Syrup - 50mg/5ml
□	AI.301.441.86.2	Lamivudine - Tablet - 150mg
☑	AI.301.442.86.1	Lamivudine + Nevirapine + Stavudine - Tablet - 30mg + 50mg + 6mg
☑	AI.301.442.86.1	Lamivudine + Nevirapine + Stavudine - Tablet - 150mg +200mg +30mg
□	AI.301.442.86.1	Lamivudine + Nevirapine + Stavudine - Tablet - 60mg + 100mg + 12mg
□	AI.301.443.86.1	Lamivudine + Stavudine - Tablet - 30mg +6mg
□	AI.301.443.86.2	Lamivudine + Stavudine - Tablet - 60mg +12mg
□	AI.301.443.86.3	Lamivudine + Stavudine - Tablet - 150mg +30mg

Item Name
Laminvudine + Nevirapine + Stavudine - Tabl...
Laminvudine + Nevirapine + Stavudine - Tabl...
Fluconazole - Injection - IV infusion 2mg
Efavirenze - Liquid - 30mg/ml
Efavirenze - Capsule - 50mg
Efavirenze - Capsule - 200mg
Cloxacillin Sodium - Syrup - 125 mg/5 ml
Aciclovir - Eye Ointment - 3%
Abacavir - Tablet - 300 mg

Figure 5-2: The Main Screen of the Warehouse (Hub) Edition



Antilipemic Agents

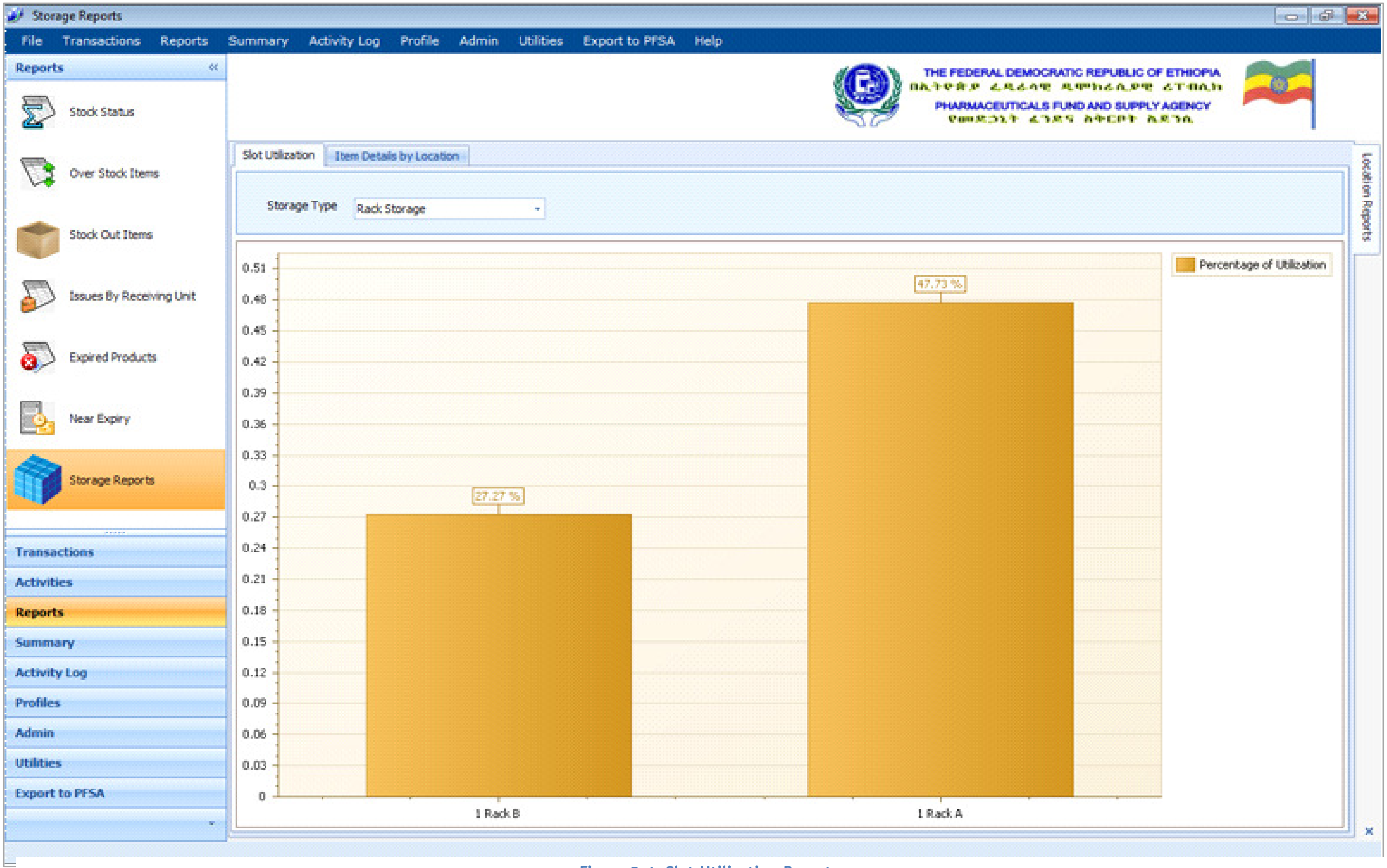
No.	Stock Code	Item Name	ABC	VEN	Used @ Faci...
177	CV.300.66.86.1	Atrovastatin - Tablet - 10mg			□
410	CV.300.162.64.1	Cholestyramine - Powder - 4g			□
411	CV.300.162.64.2	Cholestyramine - Powder - 9g			□
775	CV.300.310.86.1	Ezetimbe - Tablet - 10mg			□
836	CV.300.338.3.1	Fluvastatin - Capsule - 20mg			□
837	CV.300.338.3.2	Fluvastatin - Capsule - 40mg			□
838	CV.300.338.86.3	Fluvastatin - Tablet - 80mg			□
866	CV.300.352.3.1	Gemfibrozil - Capsule - 300mg			□
[illegible]	00.477.86.1	Lovastatin - Tablet - 20mg			□
1763	CV.300.716.86.2	Simvastatin - Tablet - 10mg			□
1764	CV.300.716.86.3	Simvastatin - Tablet - 20mg			□
1765	CV.300.716.86.4	Simvastatin - Tablet - 40mg			□
1766	CV.300.716.86.1	Simvastatin - Tablet - 5mg			□

Figure 5-3: The Main Screen of the Facility Edition

- The Warehouse (Hub) Edition

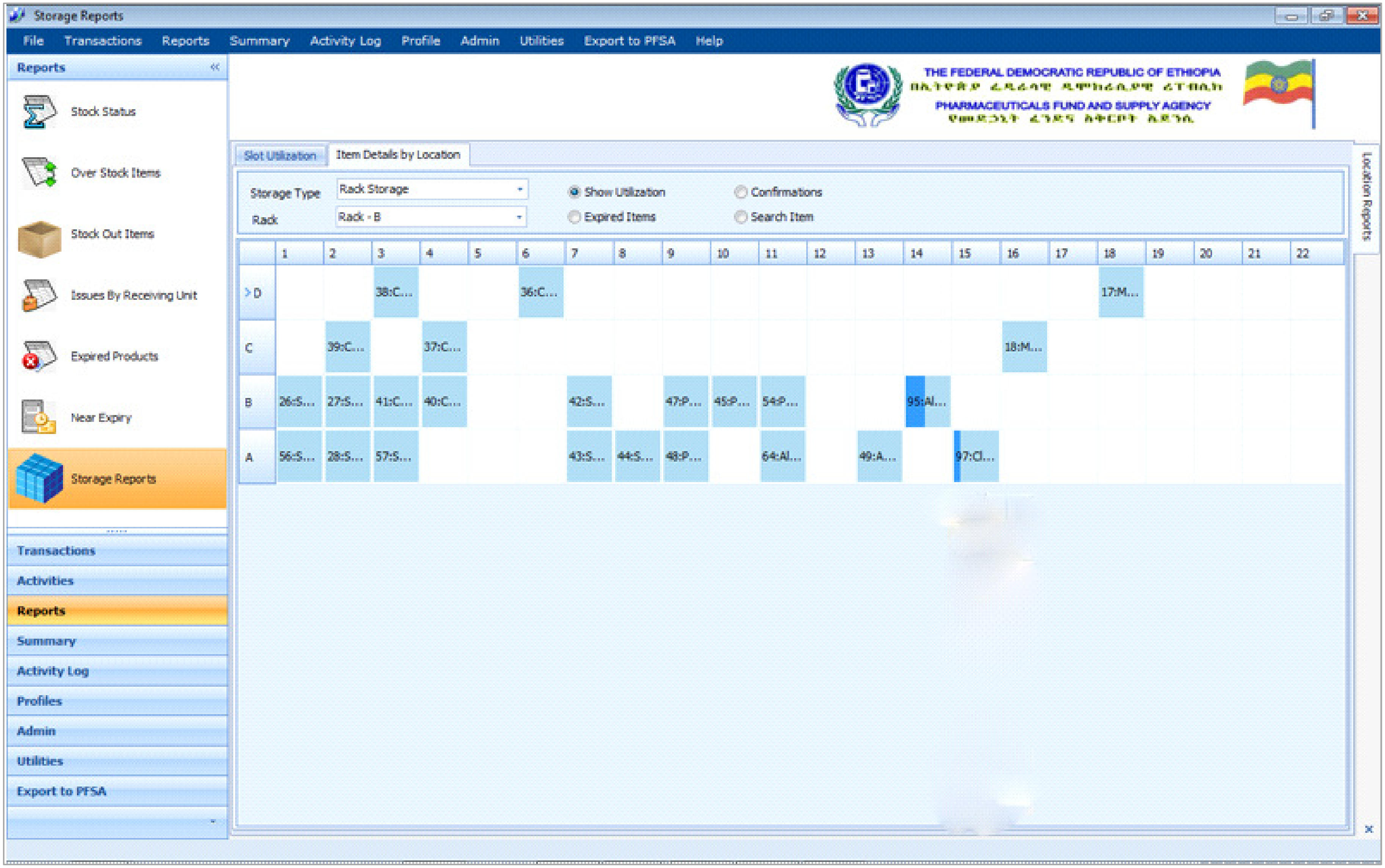
The user interface for the Warehouse (Hub) edition contains features that common with that of the Facility edition. In addition to these common features, the Warehouse (Hub) edition provides options to access and manage information regarding the Hub, Hub Setting, Pick List, Internal Movements, and Pick Face Replenishment.

The reports that will be generated in this edition vary with that of the Facility Edition. Such reports that are pertinent to the Warehouse (Hub) Edition include Issues By the Receiving Unit and Storage Reports. The Storage Report helps determine the slot utilization and item details by location. Moreover, the system presents the Storage Report graphically so that slot utilization and detail item location information can be easily visualized. Figure 5-4 and Figure 5-5 show the Slot Utilization and Item Details By location Storage Reports respectively.



Storage Type: Rack Storage

Figure 5-4: Slot Utilization Report



Storage Type: Rack Storage

Rack: Rack - B

● Show Utilization ○ Confirmations

○ Expired Items ○ Search Item

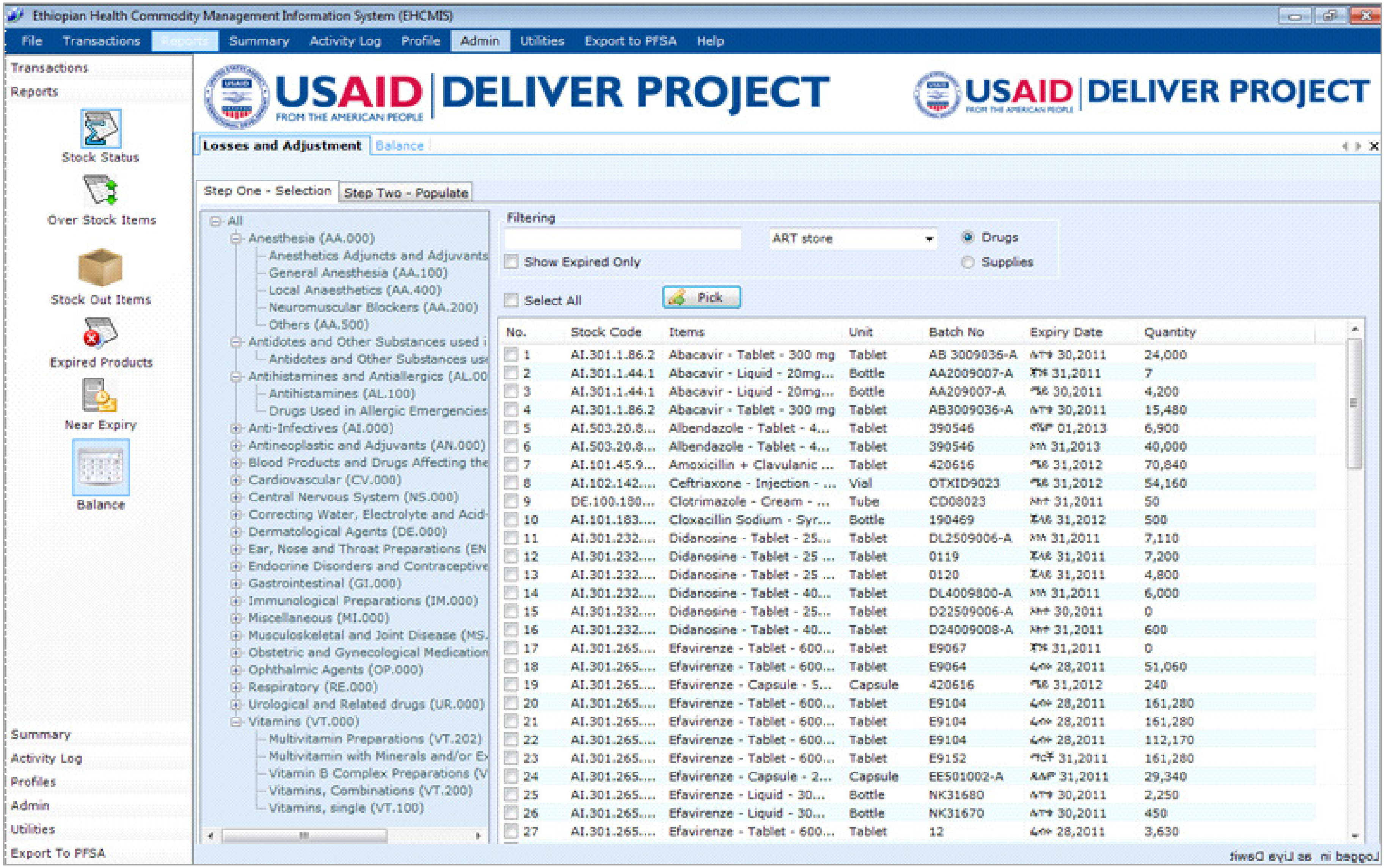
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
D			38:C...			36:C...												17:M...				
C		39:C...		37:C...												18:M...						
B	26:S...	27:S...	41:C...	40:C...			42:S...		47:P...	45:P...	54:P...			95:Al...								
A	56:S...	28:S...	57:S...				43:S...	44:S...	48:P...		64:Al...		49:A...		97:Cl...							

Figure 5-5: The Item Details By Location Storage Report

- ✧ The Facility Edition

This edition of the HCMIS system contains features that are peculiar to the facilities. Handling transactions such as receives, issues, loss/adjustment, and inventory control are some of the features to properly manage information regarding the facilities. In addition, the facility edition provides reports such as the stock balance, and loss and adjustment.

Figures 5-6 to 5-9 show some of the screen shots of the Facility Edition.


Ethiopian Health Commodity Management Information System (EHCMIS)
File
Transactions
Reports
Summary
Activity Log
Profile
Admin
Utilities
Export to PFSA
Help
Transactions
Reports
Stock Status
Over Stock Items
Stock Out Items
Expired Products
Near Expiry
Balance
Summary
Activity Log
Profiles
Admin
Utilities
Export To PFSA
USAID | DELIVER PROJECT
FROM THE AMERICAN PEOPLE
Losses and Adjustment
Balance
Step One - Selection
Step Two - Populate
All
Anesthesia (AA.000)
Anesthetics Adjuncts and Adjuvants
General Anesthesia (AA.100)
Local Anaesthetics (AA.400)
Neuromuscular Blockers (AA.200)
Others (AA.500)
Antidotes and Other Substances used i
Antidotes and Other Substances use
Antihistamines and Antiallergics (AL.00
Antihistamines (AL.100)
Drugs Used in Allergic Emergencies
Anti-Infectives (AI.000)
Antineoplastic and Adjuvants (AN.000)
Blood Products and Drugs Affecting the
Cardiovascular (CV.000)
Central Nervous System (NS.000)
Correcting Water, Electrolyte and Acid-
Dermatological Agents (DE.000)
Ear, Nose and Throat Preparations (EN
Endocrine Disorders and Contraceptive
Gastrointestinal (GI.000)
Immunological Preparations (IM.000)
Miscellaneous (MI.000)
Musculoskeletal and Joint Disease (MS.
Obstetric and Gynecological Medication
Ophthalmic Agents (OP.000)
Respiratory (RE.000)
Urological and Related drugs (UR.000)
Vitamins (VT.000)
Multivitamin Preparations (VT.202)
Multivitamin with Minerals and/or E
Vitamin B Complex Preparations (V
Vitamins, Combinations (VT.200)
Vitamins, single (VT.100)
Filtering
Pick
Logged in as Liya Dawit

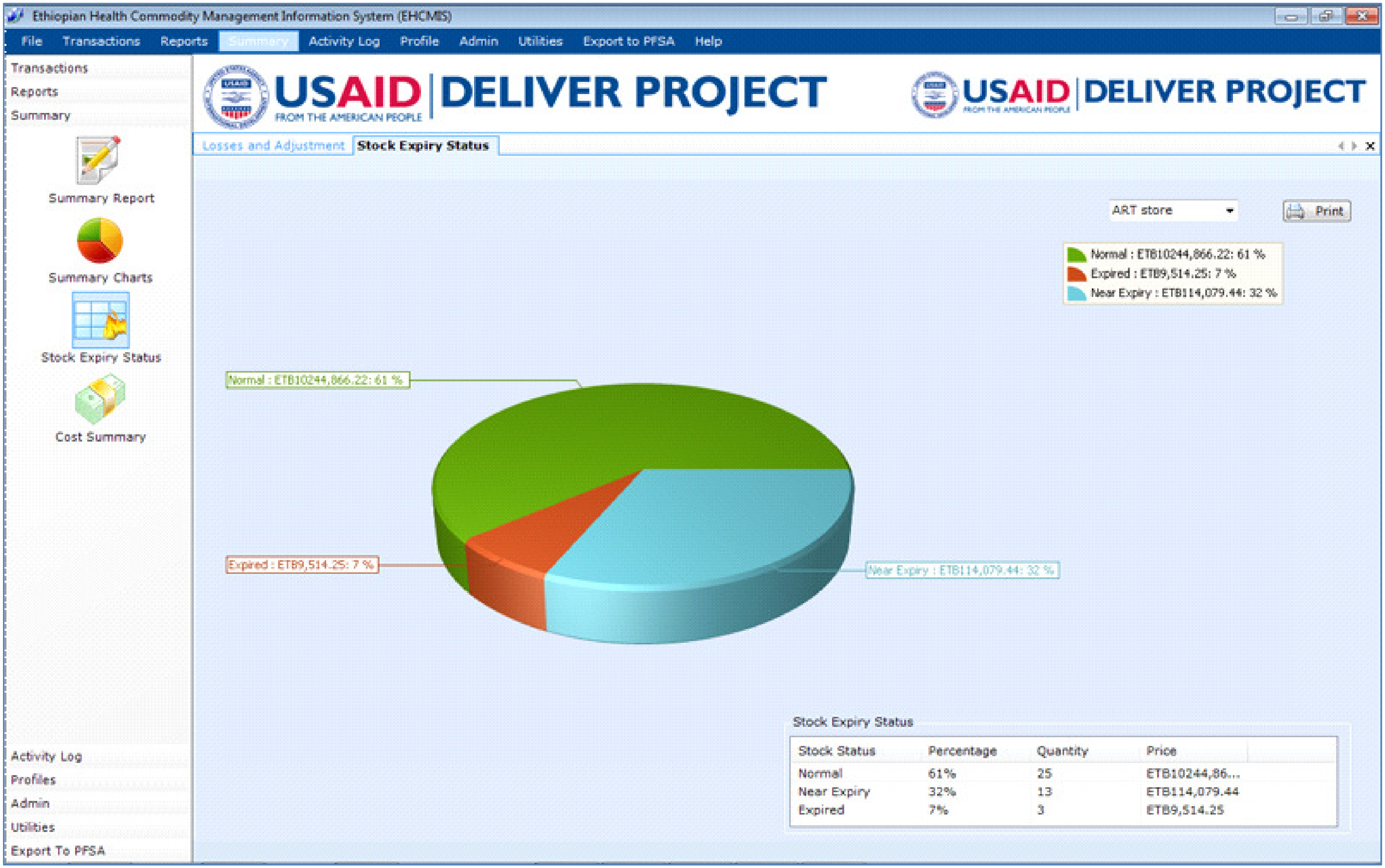
Filtering: ART store; ● Drugs ○ Supplies

□ Show Expired Only

□ Select All

No.	Stock Code	Items	Unit	Batch No	Expiry Date	Quantity
□ 1	AI.301.1.86.2	Abacavir - Tablet - 300 mg	Tablet	AB 3009036-A	ሴፕቴ 30,2011	24,000
□ 2	AI.301.1.44.1	Abacavir - Liquid - 20mg...	Bottle	AA2009007-A	ጃንዩ 31,2011	7
□ 3	AI.301.1.44.1	Abacavir - Liquid - 20mg...	Bottle	AA209007-A	ጁን 30,2011	4,200
□ 4	AI.301.1.86.2	Abacavir - Tablet - 300 mg	Tablet	AB3009036-A	ሴፕቴ 30,2011	15,480
□ 5	AI.503.20.8...	Albendazole - Tablet - 4...	Tablet	390546	ኖቬም 01,2013	6,900
□ 6	AI.503.20.8...	Albendazole - Tablet - 4...	Tablet	390546	ሜይ 31,2013	40,000
□ 7	AI.101.45.9...	Amoxicillin + Clavulanic ...	Tablet	420616	ጁላይ 31,2012	70,840
□ 8	AI.102.142....	Ceftriaxone - Injection - ...	Vial	OTXID9023	ጁላይ 31,2012	54,160
□ 9	DE.100.180...	Clotrimazole - Cream - ...	Tube	CD08023	ማርች 31,2011	50
□ 10	AI.101.183....	Cloxacillin Sodium - Syr...	Bottle	190469	ዲሴም 31,2012	500
□ 11	AI.301.232....	Didanosine - Tablet - 25...	Tablet	DL2509006-A	ሜይ 31,2011	7,110
□ 12	AI.301.232....	Didanosine - Tablet - 25 ...	Tablet	0119	ዲሴም 31,2011	7,200
□ 13	AI.301.232....	Didanosine - Tablet - 25 ...	Tablet	0120	ዲሴም 31,2011	4,800
□ 14	AI.301.232....	Didanosine - Tablet - 40...	Tablet	DL4009800-A	ሜይ 31,2011	6,000
□ 15	AI.301.232....	Didanosine - Tablet - 25...	Tablet	D22509006-A	ማርች 30,2011	0
□ 16	AI.301.232....	Didanosine - Tablet - 40...	Tablet	D24009008-A	ማርች 31,2011	600
□ 17	AI.301.265....	Efavirenze - Tablet - 600...	Tablet	E9067	ጃንዩ 31,2011	0
□ 18	AI.301.265....	Efavirenze - Tablet - 600...	Tablet	E9064	ፌብሩ 28,2011	51,060
□ 19	AI.301.265....	Efavirenze - Capsule - 5...	Capsule	420616	ጁላይ 31,2012	240
□ 20	AI.301.265....	Efavirenze - Tablet - 600...	Tablet	E9104	ፌብሩ 28,2011	161,280
□ 21	AI.301.265....	Efavirenze - Tablet - 600...	Tablet	E9104	ፌብሩ 28,2011	161,280
□ 22	AI.301.265....	Efavirenze - Tablet - 600...	Tablet	E9104	ፌብሩ 28,2011	112,170
□ 23	AI.301.265....	Efavirenze - Tablet - 600...	Tablet	E9152	ኦክተ 31,2011	161,280
□ 24	AI.301.265....	Efavirenze - Capsule - 2...	Capsule	EE501002-A	ኦገስ 31,2011	29,340
□ 25	AI.301.265....	Efavirenze - Liquid - 30...	Bottle	NK31680	ሴፕቴ 30,2011	2,250
□ 26	AI.301.265....	Efavirenze - Liquid - 30...	Bottle	NK31670	ሴፕቴ 30,2011	450
□ 27	AI.301.265....	Efavirenze - Tablet - 600...	Tablet	12	ፌብሩ 28,2011	3,630

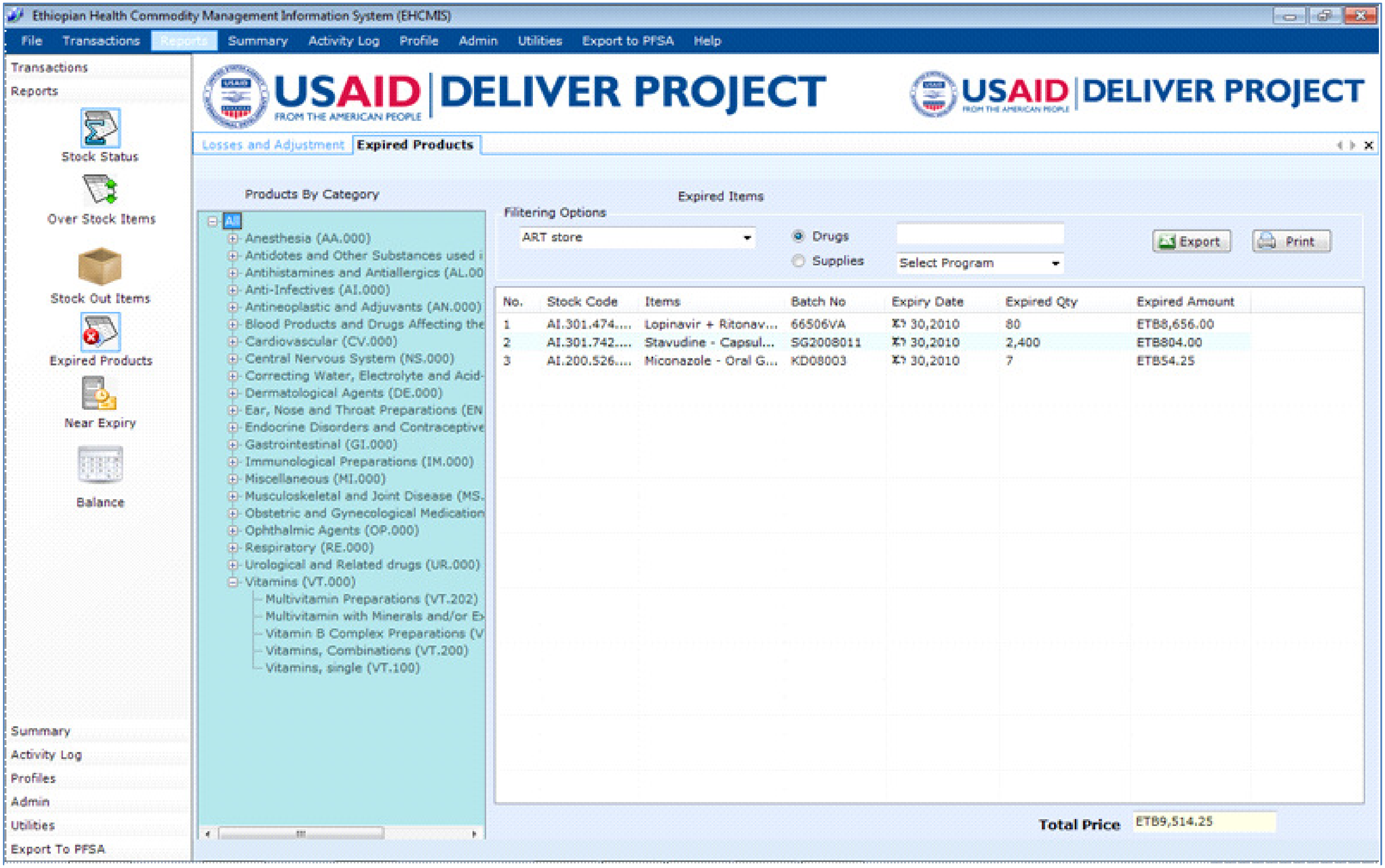
Figure 5-6: Stock Balance Report of the Facility Edition


Ethiopian Health Commodity Management Information System (EHCMIS)
File
Transactions
Reports
Summary
Activity Log
Profile
Admin
Utilities
Export to PFSA
Help
Transactions
Reports
Summary
Summary Report
Summary Charts
Stock Expiry Status
Cost Summary
Activity Log
Profiles
Admin
Utilities
Export To PFSA
USAID FROM THE AMERICAN PEOPLE
DELIVER PROJECT
USAID FROM THE AMERICAN PEOPLE
DELIVER PROJECT
Losses and Adjustment
Stock Expiry Status
ART store
Print
🟩 Normal : ETB10244,866.22: 61 %
🟥 Expired : ETB9,514.25: 7 %
🟦 Near Expiry : ETB114,079.44: 32 %
Normal : ETB10244,866.22: 61 %
Expired : ETB9,514.25: 7 %
Near Expiry : ETB114,079.44: 32 %

Stock Expiry Status

Stock Status	Percentage	Quantity	Price
Normal	61%	25	ETB10244,86...
Near Expiry	32%	13	ETB114,079.44
Expired	7%	3	ETB9,514.25

Figure 5-7: Stock Expiry Status Summary Report


Ethiopian Health Commodity Management Information System (EHCMIS)
File
Transactions
Reports
Summary
Activity Log
Profile
Admin
Utilities
Export to PFSA
Help
Transactions
Reports
Stock Status
Over Stock Items
Stock Out Items
Expired Products
Near Expiry
Balance
Summary
Activity Log
Profiles
Admin
Utilities
Export To PFSA
USAID | DELIVER PROJECT
FROM THE AMERICAN PEOPLE
USAID | DELIVER PROJECT
FROM THE AMERICAN PEOPLE
Losses and Adjustment
Expired Products
Products By Category
All
Anesthesia (AA.000)
Antidotes and Other Substances used i
Antihistamines and Antiallergics (AL.00
Anti-Infectives (AI.000)
Antineoplastic and Adjuvants (AN.000)
Blood Products and Drugs Affecting the
Cardiovascular (CV.000)
Central Nervous System (NS.000)
Correcting Water, Electrolyte and Acid-
Dermatological Agents (DE.000)
Ear, Nose and Throat Preparations (EN
Endocrine Disorders and Contraceptive
Gastrointestinal (GI.000)
Immunological Preparations (IM.000)
Miscellaneous (MI.000)
Musculoskeletal and Joint Disease (MS.
Obstetric and Gynecological Medication
Ophthalmic Agents (OP.000)
Respiratory (RE.000)
Urological and Related drugs (UR.000)
Vitamins (VT.000)
Multivitamin Preparations (VT.202)
Multivitamin with Minerals and/or E
Vitamin B Complex Preparations (V
Vitamins, Combinations (VT.200)
Vitamins, single (VT.100)
Expired Items
Filtering Options
Export
Print

ART store

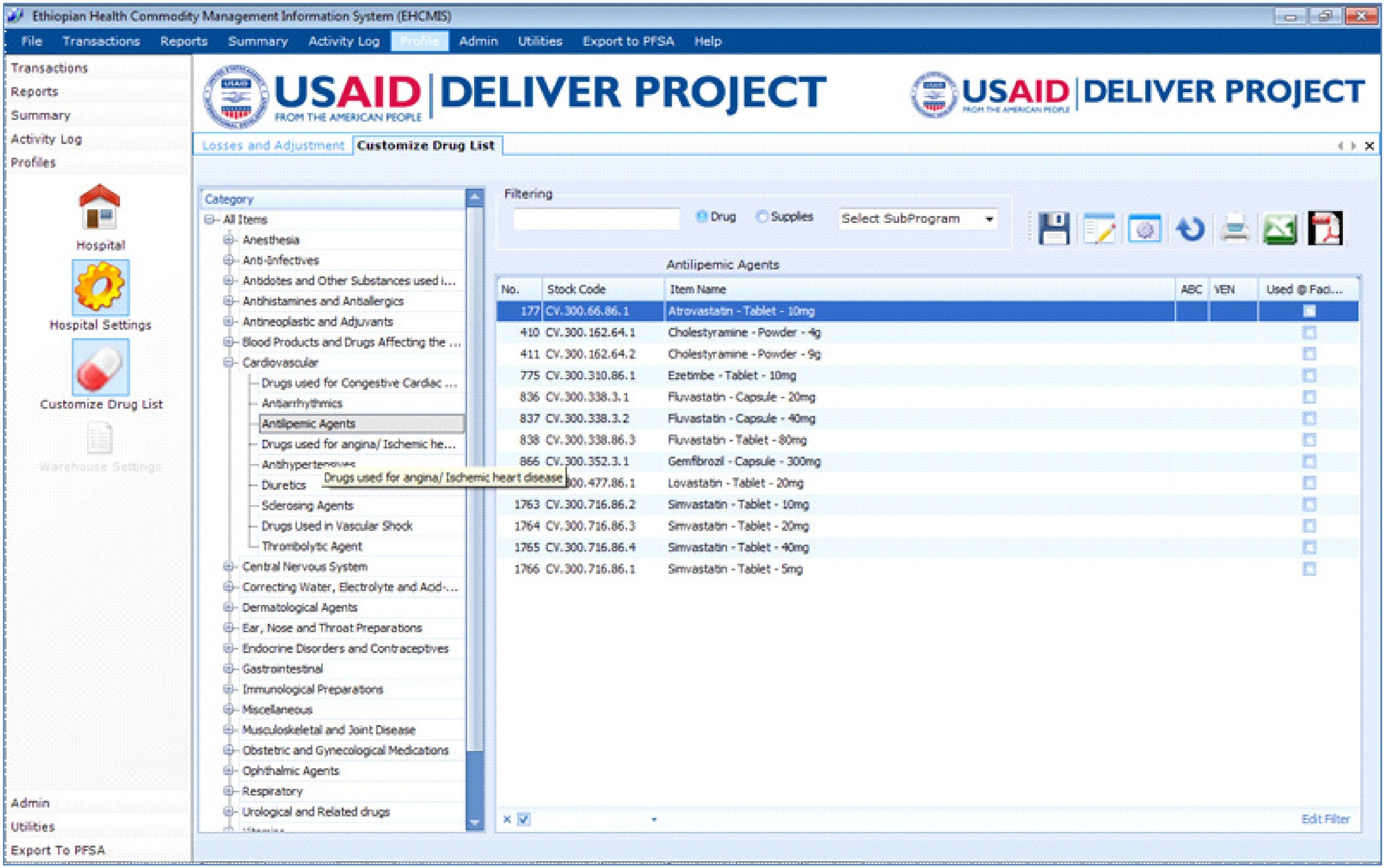
● Drugs
○ Supplies

Select Program

No.	Stock Code	Items	Batch No	Expiry Date	Expired Qty	Expired Amount
1	AI.301.474....	Lopinavir + Ritonav...	66506VA	ጥር 30,2010	80	ETB8,656.00
2	AI.301.742....	Stavudine - Capsul...	SG2008011	ጥር 30,2010	2,400	ETB804.00
3	AI.200.526....	Miconazole - Oral G...	KD08003	ጥር 30,2010	7	ETB54.25

Total Price ETB9,514.25

Figure 5-8: Expired Products Report



Antilipemic Agents

No.	Stock Code	Item Name	ABC	VEN	Used @ Faci...
177	CV.300.66.86.1	Atrovastatin - Tablet - 10mg			□
410	CV.300.162.64.1	Cholestyramine - Powder - 4g			□
411	CV.300.162.64.2	Cholestyramine - Powder - 9g			□
775	CV.300.310.86.1	Ezetimbe - Tablet - 10mg			□
836	CV.300.338.3.1	Fluvastatin - Capsule - 20mg			□
837	CV.300.338.3.2	Fluvastatin - Capsule - 40mg			□
838	CV.300.338.86.3	Fluvastatin - Tablet - 80mg			□
866	CV.300.352.3.1	Gemfibrozil - Capsule - 300mg			□
[illegible]	[illegible]00.477.86.1	Lovastatin - Tablet - 20mg			□
1763	CV.300.716.86.2	Simvastatin - Tablet - 10mg			□
1764	CV.300.716.86.3	Simvastatin - Tablet - 20mg			□
1765	CV.300.716.86.4	Simvastatin - Tablet - 40mg			□
1766	CV.300.716.86.1	Simvastatin - Tablet - 5mg			□

Figure 5-9: Facility Edition Screen to Customize Drug List Setting

□ USER LEVELS (ACCESS CONTROL)

The system users the Facility as well as Warehouse (Hub) editions of the HCMIS system include the Administrator, Editor and Reader. These users have some degree of control on how to execute an operation on some objects of the system. In the HCMIS system, this is indicated using an access control matrix.

The access control matrix in Table 5-1 shows that the administrator has access and permission to create an account and manage it (by editing the profile, changing password, log in, and log out). On the other hand, the Editor and Reader users are granted access only to log in and out of the system, and change their password.

Moreover, the administrator has the permission to create, edit, and delete information regarding activity logs (i.e. receive transaction, issue transaction, loss/adjustment and inventory logs), activities (such as internal movements, loss/adjustment, pick face replenishment, and inventory), profiles (regarding the Hub, Hospital, Hospital Settings, Customize Drug List, and Warehouse Settings) backup and system password, reports, and system settings. The Editor, on the other hand, is also granted permission to create and edit activity logs and activity information. Moreover, the Editor, similar to that of the administrator, can also view reports and export data to PFSA format.

The other user in the HCMIS System, the Reader, has the access controls that can help manage its account settings (by changing the password), log in and out of the system. Besides, the Reader is allowed to view reports that are useful to make sound decisions regarding the facilities and/or warehouses (hubs).

Table 5-1: Access Control Matrix

Objects / Actors	Administrator	Editor	Reader

User Accounts	create()
	changePW()
	delete()
	logIn()
	logOut()
		changePW()
		logIn()
		logOut()
			changePW()
			logIn()
			logOut()

Activity Logs	create()
	edit()
	delete()
		create()
		edit()	

Activities	create()
	edit()
	delete()
		create()
		edit()
		delete()	

Utilities	create()
	edit()
	delete()
		create()
		edit()	

Reports	view()	view()	view()

System Settings	create()
	edit()
	delete()
		create()
		edit()	

Exporting Data to PFSA Format	export()	export()	

5.4. Installation Guidelines

As indicated in the previous sections of the document, the HCMIS System is offers an easy to use option for non-technical as well as IT personnel at the facilities and/or warehouses. In this section, therefore, the HCMIS System installation steps are listed out. Note that the installation steps are applicable for both the Warehouse (Hub) as well as Facility editions.

□ PREREQUISITE

The HCMIS software requires a fully functional Windows environment with .NET Framework and SQL Server Express edition installed.

□ TO INSTALL THE HCMIS SOFTWARE

1. Insert either the CD-ROM or the removable media (such as Flash Disk) containing the HCMIS software setup.
2. From the root folder double click the setup.exe.
3. Next, the setup will check for the installation prerequisites. If .NET framework is not installed, it will automatically start to install the .NET 3.5 framework setup.
4. Just follow the simple installation wizard until the installation is finished
5. Next it will detect the presence of the SQL Server Express edition installation. If the installation is not present, it will prompt the user to install it. Follow the instruction wizard until all dependencies are properly installed. Finally, it will install the software.
6. Once the installation is successfully completed, double click the HCMIS Icon from the desktop or from the program menu list. Then, enter the username and password on the login window and start using the system.

Appendix A

Table: Items		
Column Name	Data Type	Allow Nulls
ID (Primary Key)	int	No
StockCode	varchar(50)	Yes
Strength	varchar(50)	Yes
DosageFormID	int	Yes
UnitID	int	No
VEN	int	Yes
ABC	int	Yes
IsFree	bit	Yes
IsDiscontinued	bit	Yes
Cost	decimal(18, 0)	Yes
EDL	bit	Yes
Refrigeratored	bit	Yes
Pediatric	bit	Yes
IINID	int	Yes
IsInHospitalList	bit	Yes
NeedExpiryBatch	bit	Yes
Code	varchar(50)	Yes
StockCodeDACA	varchar(50)	Yes
NearExpiryTrigger	float	Yes
StorageTypeID	int	Yes
IsStackStored	bit	No

Table: ItemSupplier		
Column Name	Data Type	Allow Nulls
ID (Primary Key)	int	No
SupplierID	int	Yes
ItemID	int	Yes

Table: IssueDoc

Column Name	Data Type	Allow Nulls
ID (Primary Key)	int	No
ItemID	int	Yes
StoreId	int	Yes
ReceivingUnitID	int	Yes
LocalBatchNo	varchar(50)	Yes
Quantity	bigint	Yes
Date	datetime	Yes
IsTransfer	bit	Yes
IssuedBy	varchar(50)	Yes
Remark	text	Yes
RefNo	varchar(50)	Yes
BatchNo	varchar(50)	Yes
IsApproved	bit	Yes
Cost	float	Yes
NoOfPack	int	Yes
QtyPerPack	int	Yes
DUSOH	bigint	Yes
DURequestedBy	bigint	Yes
RecomendedQty	bigint	Yes
ReceiveDocID	int	Yes
EurDate	datetime	Yes
OrderID	int	Yes

Table: VEN

Column Name	Data Type	Allow Nulls
ID (Primary Key)	int	No
Value	varchar(50)	Yes
Description	text	Yes

Table: ABC

Column Name	Data Type	Allow Nulls
ID (Primary Key)	int	No
Value	Varchar(50)	Yes
Description	text	Yes

Table: Disposal

Column Name	Data Type	Allow Nulls
ID (Primary Key)	int	No
ItemID	int	Yes
StoreId	int	Yes
ReasonID	int	Yes
Quantity	bigint	Yes
Date	datetime	Yes
ApprovedBy	varchar(50)	Yes
Losses	bit	Yes
BatchNo	varchar(50)	Yes
Remark	text	Yes
Cost	float	Yes
RefNo	varchar(50)	Yes
EurDate	datetime	Yes
RecID	int	Yes

Table: DisposalReasons

Column Name	Data Type	Allow Nulls
ID (Primary Key)	int	No
Reason	text	Yes
Description	text	Yes

Table: ReceiveDoc

Column Name	Data Type	Allow Nulls
ID (Primary Key)	int	No
BatchNo	varchar(50)	Yes
ItemId	int	Yes
SupplierID	int	Yes
Quantity	bigint	Yes
Date	datetime	Yes
ExpDate	datetime	Yes
Out	bit	Yes
ReceivedStatus	int	Yes
ReceivedBy	varchar(50)	Yes
Remark	text	Yes
StoreID	int	Yes
LocalBatchNo	varchar(50)	Yes
RefNo	varchar(50)	Yes
Cost	float	Yes
IsApproved	bit	Yes
ManufactureId	int	Yes
QuantityLeft	bigint	Yes
NoOfPack	int	Yes
EurDate	datetime	Yes
BoxLevel	int	Yes
SellingPrice	float	Yes
PricePerPack	float	Yes

Table: Supplier

Column Name	Data Type	Allow Nulls
ID (Primary Key)	int	No
CompanyName	varchar(50)	Yes
Address	varchar(50)	Yes
Telephone	varchar(50)	Yes
ContactPerson	varchar(50)	Yes
Mobile	varchar(50)	Yes
CompanyInfo	varchar(50)	Yes
IsActive	bit	Yes
Email	varchar(50)	Yes

Table: ReceivePallet

Column Name	Data Type	Allow Nulls
ID (Primary Key)	int	No
ReceiveID	int	Yes
PalletID	int	Yes
ReceivedQuantity	bigint	Yes
Balance	bigint	Yes
ReservedStock	int	No
ReservedOrderID	int	Yes
BoxSize	int	Yes

Table: DosageForm

Column Name	Data Type	Allow Nulls
ID (Primary Key)	int	No
Form	varchar(50)	Yes
Description	text	Yes
TypeID	int	Yes

Table: Balance

Column Name	Data Type	Allow Nulls
ID (Primary Key)	int	No
ItemID	int	Yes
StoreId	int	Yes
LastDate	datetime	Yes
SOH	bigint	Yes
PhysicalCount	bigint	Yes
Month	int	Yes
Year	int	Yes
AMC	bigint	Yes
SOHCost	float	Yes

Table: Stores

Column Name	Data Type	Allow Nulls
ID (Primary Key)	int	No
HospitalID	int	Yes
StoreName	varchar(50)	Yes

Table: GeneralInfo		
Column Name	Data Type	Allow Nulls
ID (Primary Key)	int	No
HospitalName	varchar(50)	Yes
Woreda	int	Yes
Zone	int	Yes
Region	int	Yes
Telephone	varchar(50)	Yes
HospitalContact	varchar(50)	Yes
LeadTime	int	Yes
Min	int	Yes
Max	int	Yes
SafetyStock	int	Yes
AMCRange	int	Yes
ReviewPeriod	int	Yes
EOP	float	Yes
Description	text	Yes
IsEven	bit	Yes
Logo	varchar(50)	Yes
DUMin	float	Yes
DUMax	float	Yes
DUAMCRange	int	Yes
LastBackUp	datetime	Yes

Table: ProgramProduct		
Column Name	Data Type	Allow Nulls
ID (Primary Key)	int	No
ProgramID	int	Yes
ItemID	int	Yes

Table: Programs

Column Name	Data Type	Allow Nulls
ID (Primary Key)	int	No
Name	varchar(50)	Yes
Description	text	Yes
ParentID	int	Yes
ProgramCode	varchar(50)	Yes

Table: Unit

Column Name	Data Type	Allow Nulls
ID (Primary Key)	int	No
Unit	varchar(50)	Yes
Description	text	Yes

Table: ReceivingUnits

Column Name	Data Type	Allow Nulls
ID (Primary Key)	int	No
Name	varchar(50)	Yes
Description	text	Yes
Phone	varchar(50)	Yes
Woreda	int	Yes
Route	int	Yes
RouteSequence	int	Yes

Table: Woreda

Column Name	Data Type	Allow Nulls
ID (Primary Key)	int	No
WoredaName	varchar(50)	Yes
ZoneID	int	Yes
WoredaCode	varchar(50)	Yes

Table: Zone

Column Name	Data Type	Allow Nulls
ID (Primary Key)	int	No
ZoneName	varchar(50)	Yes
RegionId	int	Yes
ZoneCode	varchar(50)	Yes

Table: Region

Column Name	Data Type	Allow Nulls
ID (Primary Key)	int	No
RegionName	varchar(50)	Yes
RegionCode	varchar(50)	Yes

Table: Category

Column Name	Data Type	Allow Nulls
ID (Primary Key)	int	No
CategoryName	varchar(50)	Yes
CategoryCode	varchar(50)	Yes
Description	text	Yes

Table: ProductCategory

Column Name	Data Type	Allow Nulls
ID (Primary Key)	int	No
SubCategoryID	Int	Yes
ItemId	int	Yes

Table: SubCategory

Column Name	Data Type	Allow Nulls
ID (Primary Key)	int	No
CategoryId	int	Yes
SubCategoryName	varchar(50)	Yes
SucCategoryCode	varchar(50)	Yes
Description	text	Yes
ParentID	int	Yes

Table: User

Column Name	Data Type	Allow Nulls
ID (Primary Key)	int	No
FullName	varchar(50)	Yes
Address	varchar(50)	Yes
Mobile	varchar(50)	Yes
UserName	varchar(50)	Yes
Password	varchar(50)	Yes
UserType	int	Yes
Active	bit	Yes

Table: UserType

Column Name	Data Type	Allow Nulls
ID (Primary Key)	int	No
Type	varchar(50)	Yes

Table: Shelf

Column Name	Data Type	Allow Nulls
ID (Primary Key)	int	No
StoreID	int	Yes
ShelfCode	varchar(50)	Yes
ShelfType	varchar(50)	Yes
Rows	int	Yes
Columns	int	Yes
CoordinateX	float	Yes
CoordinateY	float	Yes
Rotation	float	Yes
Length	float	Yes
Width	float	Yes
Height	float	Yes
ShelfStorageType	int	Yes