Self Correcting Inventory System

By

Sandip Kumar Singh

Submitted to Prof. Hazem Said
the Faculty of the Information Engineering Technology Program
in Partial Fulfillment of the Requirements for
the Degree of Bachelor of Science
in Information Engineering Technology

University of Cincinnati
College of Applied Science

June 2006

Self Correcting Inventory System

By

Sandip Kumar Singh

Submitted to Prof. Hazem Said
the Faculty of the Information Engineering Technology Program
in Partial Fulfillment of the Requirements
for
the Degree of Bachelor of Science
in Information Engineering Technology

Sandip Kumar Singh	Date
Prof. Russell McMahon, Faculty Advisor	Date
Prof. Patrick C. Kumpf, Ed.D. Interim Department Head	Date

Table of Contents

Section	Page
1. Project Description	8
2. Statement of Problem	9
3. Description of Solution	10-11
4. User Profile	12
5. Design Protocol	13-15
6. Functional and Technical Architecture	16-17
7. Proof of design (User Interface)	18-23
8. Testing	24-26
9. Database Diagram	27-28
10. Reports	29-31
11. Timeline	33-35
12. Cost	36
13. Deliverables	37-38
14. Conclusions and Recommendations	39
15. Flex Process database documentation (Appendix A)	40-49
16. Supporting documentation	50-57
17. Tech Expo slides	58-64
18. References	65

List of Figures

Figure Name	Page
Figure 1: Use Case Model for a Material Handler (Regular User)	14
Figure 2: Use Case Model for the Warehouse Managers (Power User)	15
Figure 3: Functional Architecture	16
Figure 4: Technical Architecture	17
Figure 5: Login Screen	19
Figure 6: Password Screen	19
Figure 7: Scan Zone Screen	19
Figure 8: Scan Location Screen	20
Figure 9: Invalid Location Screen	20
Figure 10: Scan Pallet Screen	20
Figure 11: Verify Quantity Screen	21
Figure 12: Enter New Quantity	21
Figure 13: Uncounted Locations Screen	21
Figure 14: Login	22
Figure 15: User Maintenance	23
Figure 16: Database Diagram	28
Figure 17: Inventory Count Report	29
Figure 18: Inventory Count Scorecard Report	30
Figure 19: Inventory Count List Report	31
Figure 20: Code Comments	32

Figure 21/22: Timeline **34-35**

Figure 23: Cost **36**

Acknowledgements

My thanks and appreciation to Prof. Russell McMahon for helping me as my advisor through out the time it took me to complete my Senior Design project. I would also like to thank the entire faculty in my achievement of getting the Bachelor's degree from University of Cincinnati.

Abstract

Self Correcting Inventory System
By
Sandip Kumar Singh

Self correcting inventory system: a system to resolve all the discrepancies in your inventory. This product is being developed for The Wornick Company to address the inventory issues they are experiencing due to human errors. When material handlers misplace product in the warehouse it cannot be found and an inventory adjustment is made. This causes discrepancies between the perpetual and physical inventory. The misplaced product is reported as lost in the system. Due to this, material handlers are not able to locate product in the warehouse. Also, there is no way to perform this function in real time. Material handlers are forced to write the transactions and manually enter them in the system. Self correcting inventory system is a wireless application and will allow the users to count the inventory in real time. Multiple levels of validation are performed to minimize the errors. Easy to navigate screens are implemented for users. Reports will be generated for Warehouse Supervisors for review. The Wornick Company uses the Flex Process ERP System to support their business. Flex Process has modules to support production, customer order management, Finance, human resources, and the warehouse management system. This product interfaces with **Flex Process ERP** (Enterprise Resource Planning) system and utilizes ERP API's (Application Program Interface) to populate the database.

Project Description

The Wornick Company is a recognized innovator in the food industry, providing meal solutions in manufacturing, processing, packaging, assembly and new food product development. Domestic and international food marketers, the U.S. government, the Canadian government and other governments around the world look to help them test, develop and introduce new products, refine existing products and meet their ever-changing needs. The Wornick Company has two divisions. The Prepared Food Division is located in Cincinnati, OH and is responsible for making, packing and shipping the food to the Right Away Division in McAllen, TX. The Right Away Division is a distribution center for the Wornick Company. The Wornick Company has about 1,000 employees between Cincinnati and Texas.

The Wornick Company uses the Flex Process ERP (Enterprise Resource Planning) System to support their business. Flex Process has modules to support production, customer order management, human resources, and the warehouse management system. Flex Process is also integrated with another system named Avantis. Avantis is utilized by the purchasing department. Flex Process doesn't provide any functionality for Radio Frequency (RF) out of the box. Applications can be written using the application program interface provided by Flex Process. The decision was made to design a RF system by the internal staff and add it on to the Flex Process system. The RF system is built on Flex Process and it also enforces all the business rules of Flex Process. One cannot do any transaction via RF that cannot be done in Flex Process. All the errors that occur on the RF are actually Flex Process errors. Appropriate forms are created in the Flex Process depending on the kind of activity.

Statement of Problem

When material handlers misplace product in the warehouse it cannot be found and an inventory adjustment is made. This causes discrepancies between the perpetual and physical inventory. The misplaced product is reported as lost in the system. Due to this, material handlers are not able to locate product in the warehouse. Also, there is no way to perform this function in real time. Material handlers are forced to write the transactions and manually enter them in the system. The Wornick Company would like to count products in all the locations within the warehouse and match the physical inventory with the system. All the locations will be counted every month and the application will generate a report of all the locations that are due for counting.

Description of Solution

I have written a wireless application to count and correct the discrepancies at all five warehouses. This application will build RF interface between the ERP system and symbol scanners used by warehouse coordinators. Flex Process ERP system API's (Application Program Interface) will be used to enhance Flex Process's functionality by utilizing mobile computing devices. These devices perform transactions in real-time using barcode technology to automate the process and reduce the human error factor. Flex Process ERP and company business rules will be enforced. The application will run on symbol handheld devices and will communicate with the server using Wavelink software. The RF environment will allow the end user to scan type and move freely within the working area. Non-RF devices (like wall mounted or Tablet PC's) can also interact with the Wavelink software. This solution will make real time inventory correction possible and minimize the loss of time and material. The purpose of this system is to match the system inventory with the physical inventory. A site will be broken into several small pieces called zones. Locations come under zones and different areas of the warehouse are considered different zones. The user will scan the zone in the RF. The system will validate that the user scanned a valid zone. The next step is to scan the location; the location must belong to the zone or the system will generate an error message for the user. Once the valid location is scanned all the pallets are moved onto the user's truck. *It is not necessary that the system data and physical pallet match.* The only pallets that will be moved onto the user's truck are those that the system says are in that location. The user will start scanning the physical pallets and during this process pallets will be moved back to that location one by one. There are two different scenarios that can occur during this process. Note that the location has been scanned and all the pallets from that location have been moved to the user’s truck.

1. There can be more pallets at the location according to the system than the actual inventory. In this case the extra pallets will be moved to the location called “cclost” and are considered lost. At the end of the day, the Warehouse Supervisor will review it and instruct the forklift drivers to physically find the pallets and move them to the correct location.
2. There can also be more pallets physically in the location than what the system says. When the user scans a pallet that is not on the user’s truck, the RF system finds that pallet in the entire inventory and moves it back to the scanned location. Along with this transaction the system attaches a comment that this pallet should have been in this location but it wasn’t. This transaction will show up on the zone count report presented to the Warehouse Supervisor.

User Profile

There are two user profiles based on the specifications that were provided by The Wornick Company.

Warehouse Material Handlers

Warehouse material handlers are responsible for using the "Self Correcting Inventory System" to count the product pallets in the warehouse. Material handlers have no direct access to the Flex Process ERP System. All the transaction must be done via this application.

Warehouse Managers

Warehouse managers will have full access to this application and also to the Flex Process ERP System. Flex Process access is provided to them because they are responsible for other areas of the system and must have full control of the inventory control, receiving, and shipping module of the ERP system. Warehouse managers will also have access to the maintenance application that will be used to add and delete users from the system.

Design Protocols

Use Case Diagram

The use case is divided into two main sections based on the user profiles discussed above. Self Correcting Inventory System will be access from several (20 – 30) handheld devices. This system will have different access for the Material Handler and the Warehouse Managers. The use case is explained in the following diagram for the material handlers.

Figure 1: Use Case Model for a Material Handler (Regular User)

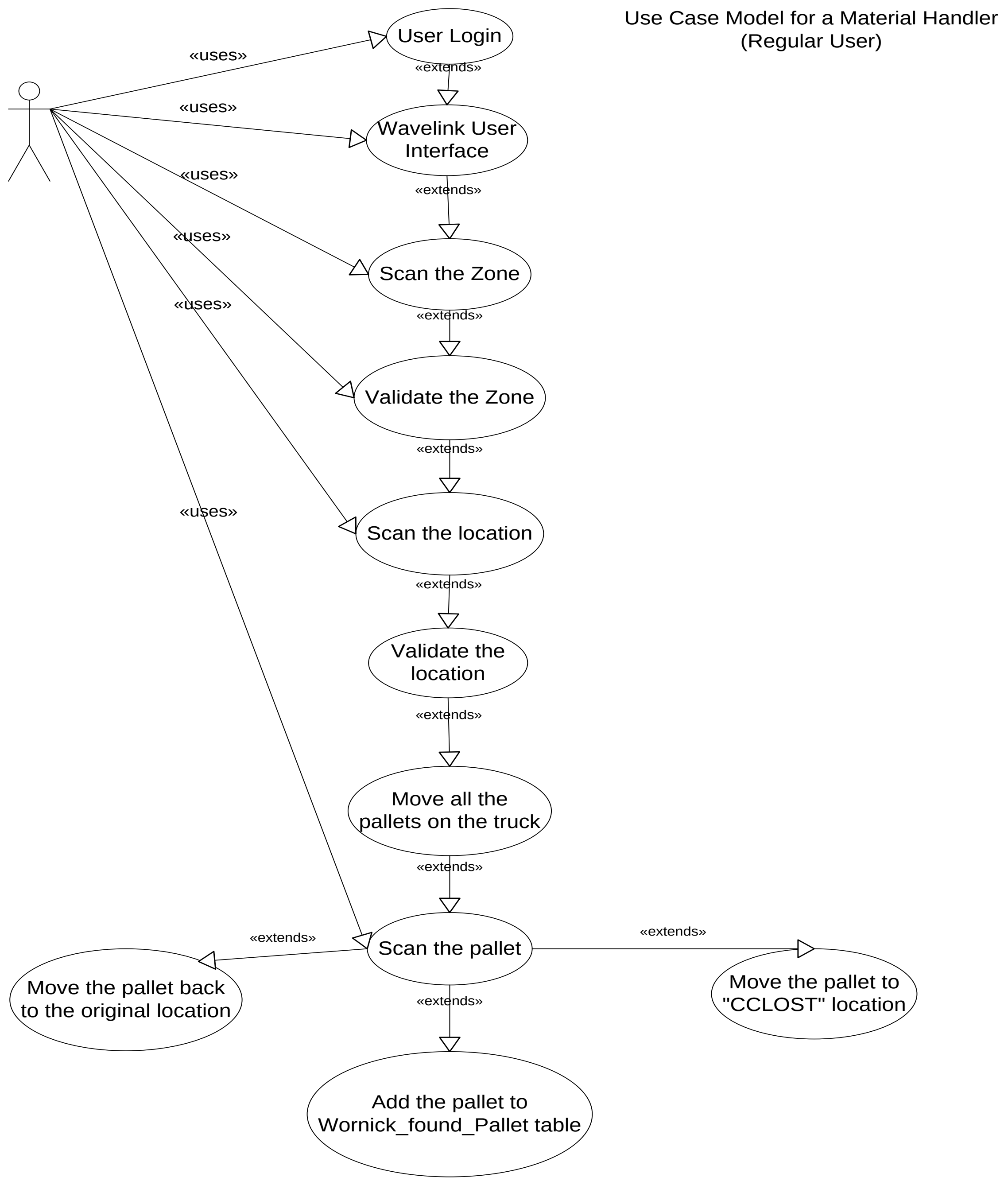


Figure 2: Use Case Model for the Warehouse Managers (Power User)

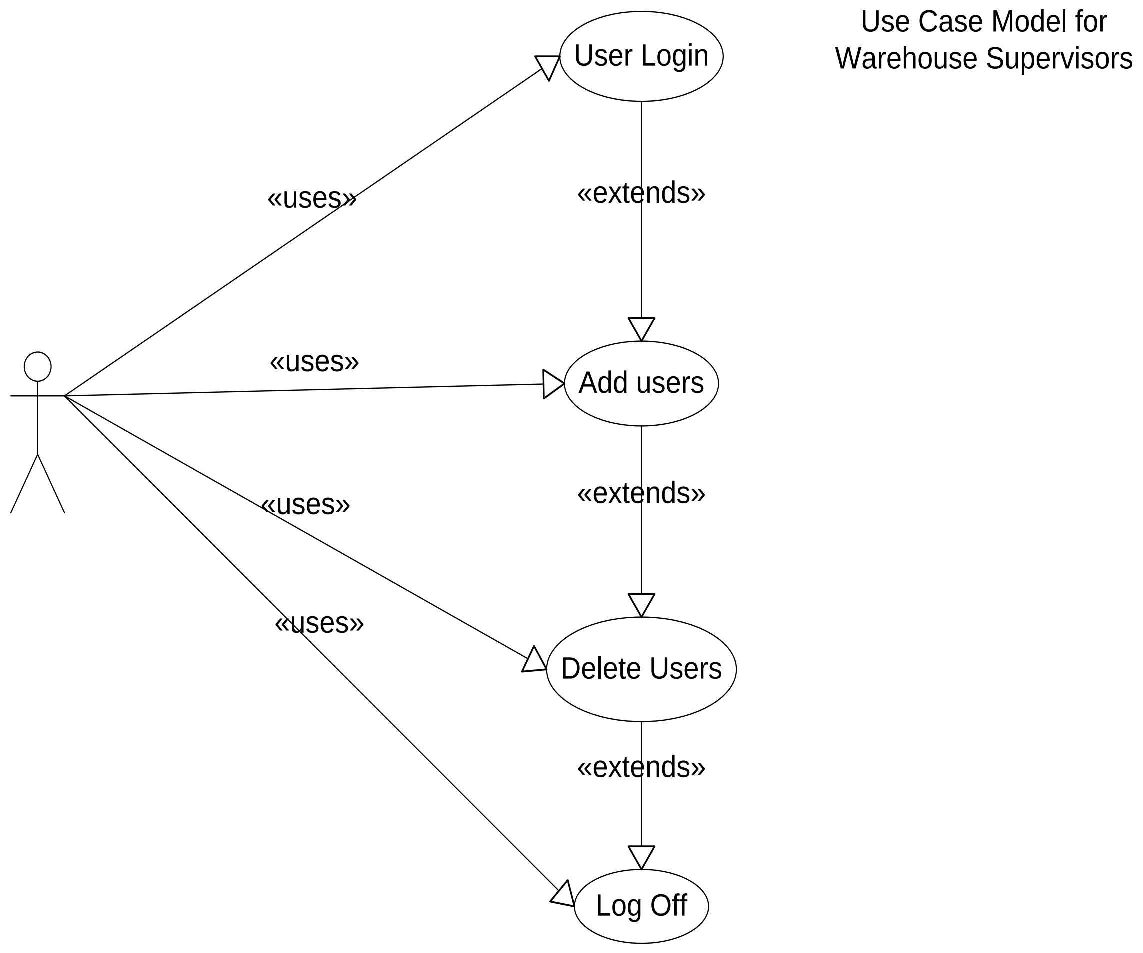


Figure 3: Functional Architecture

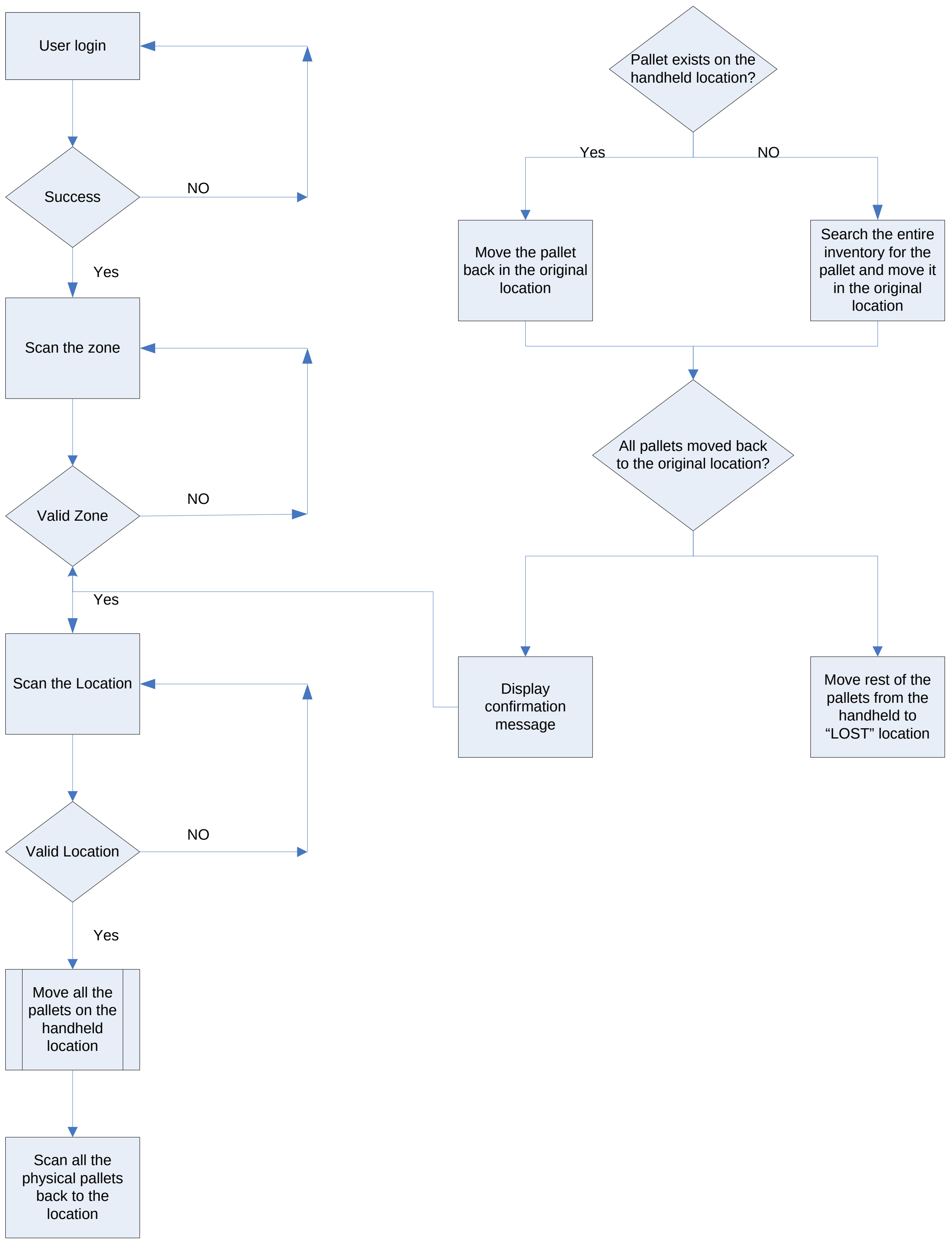
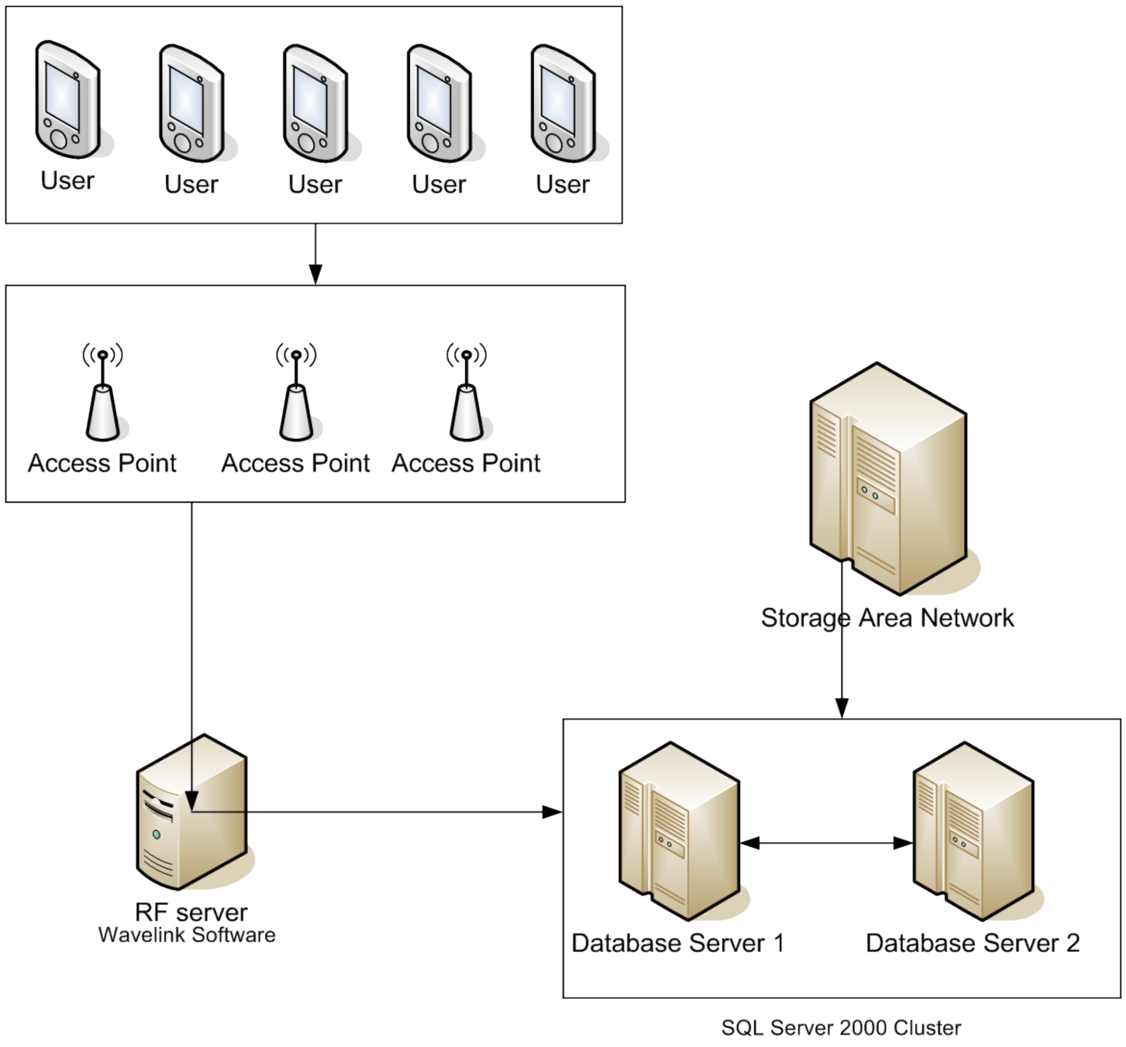

User login
Success
NO
Yes
Scan the zone
Valid Zone
NO
Yes
Scan the Location
Valid Location
NO
Yes
Move all the pallets on the handheld location
Scan all the physical pallets back to the location
Pallet exists on the handheld location?
Yes
NO
Move the pallet back in the original location
Search the entire inventory for the pallet and move it in the original location
All pallets moved back to the original location?
Display confirmation message
Move rest of the pallets from the handheld to "LOST" location

Figure 4: Technical Architecture



Proof of design (User Interface)

Material handlers will log into the RF application using the handhelds. This application is character based and has no mouse or touch screen functionality associated with it. After the successful login, the user will be able to start counting the material in the warehouse by using the Self Correcting Inventory System.

Graphics

This system is completely character based and a graphical interface is not provided to the extent because of the system limitations. Function keys will be used to navigate between the screens and they will be made visible and highlighted on the screen for the user.

User Manual

The user will be provided with the user manual (paper printed) to help them use this system because it is impossible to incorporate the help files with this system.

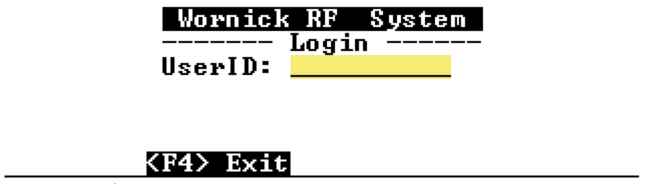
Following are the screen designs for the Self Correcting Inventory System. All the screens are designed using the Wavelink 3.7 software.

Login:

In order to enter the system you must have a valid username and password. This application requires a username and password for the following two reasons:

- Security
- Tracking Purposes

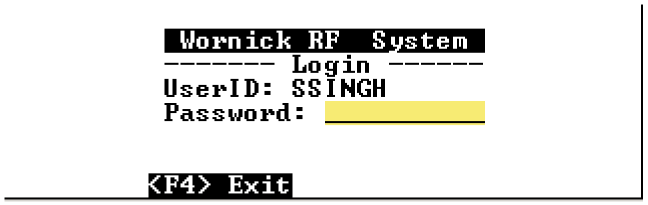
1) When you first come to your device you will see the login screen as shown in Figure 5.



```
Wornick RF  System
------- Login ------
UserID:
<F4> Exit
```

Figure 5: Login Screen

2) Type in your username and then press the enter key. Now you will be prompted for your password as shown in Figure 6.

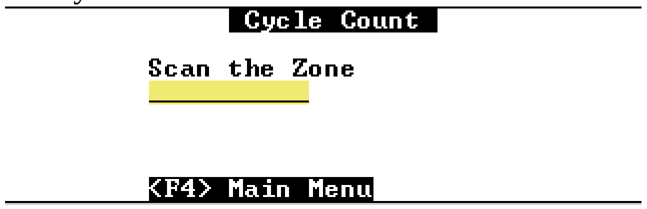


```
Wornick RF  System
------- Login ------
UserID: SSINGH
Password:
<F4> Exit
```

Figure 6: Password Screen

Self Correcting Inventory System:

1.) Next you will see the screen shown in Figure 7. Scan the zone barcode for the zone you are about to count.



```
Cycle Count

Scan the Zone

<F4> Main Menu
```

Figure 7: Scan Zone Screen

2.) After you successfully scan the zone, scan the first location that belongs to the current zone.

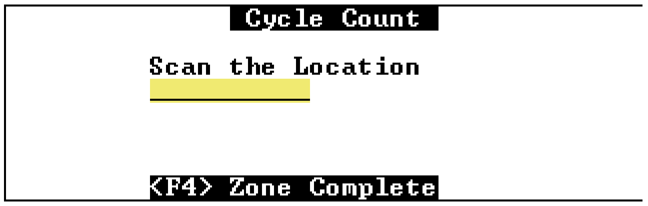

Cycle Count
Scan the Location
<F4> Zone Complete

Figure 8: Scan Location Screen

NOTE: If you scan an invalid location the screen shown in figure 8 will appear. This occurs if an invalid location is scanned for the current zone

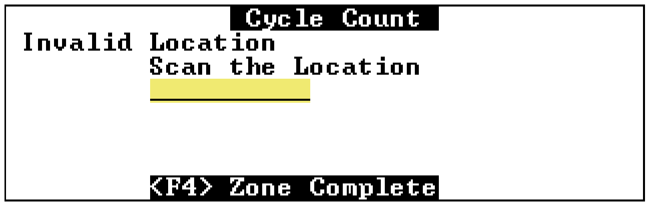

Cycle Count
Invalid Location
Scan the Location
<F4> Zone Complete

Figure 9: Invalid Location Screen

3.) Once a valid location has been scanned, the system is ready to begin scanning individual inventory items for the current location. Notice that the "Scan Pallet" screen displays the current location being scanned on the screen in Figure 10.

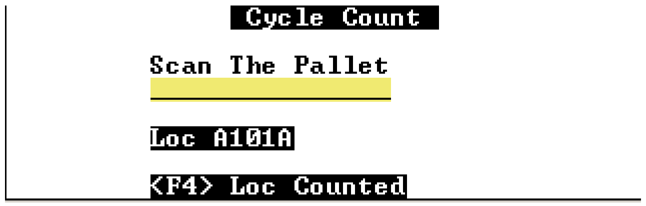
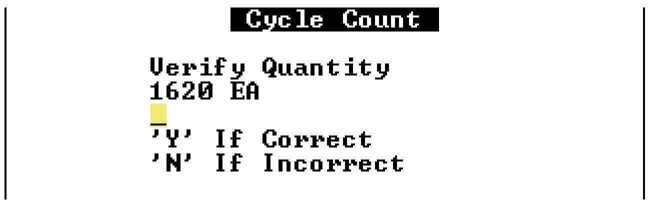

Cycle Count
Scan The Pallet
Loc A101A
<F4> Loc Counted

Figure 10: Scan Pallet Screen

At this time, all items found at the current location in the system are moved from the current location to the location of the scanning truck. Inventory items remain there temporarily until they are physically scanned and counted. If any items are not physically scanned back into the current location by the end of the location count, they are moved to CCLost.

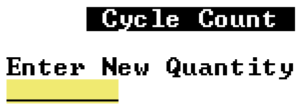
4.) As you scan the physical pallets from that location you will be asked to verify the quantity. If the pallets is sealed (full quantity) you can just press "Y" on the scanner and the system quantity will remain. In the example shown in figure 11, the quantity would be 1620 EA and the location would be moved back to "A101A



```
Cycle Count

Verify Quantity
1620 EA
_
'Y' If Correct
'N' If Incorrect
```

Figure 11: Verify Quantity Screen

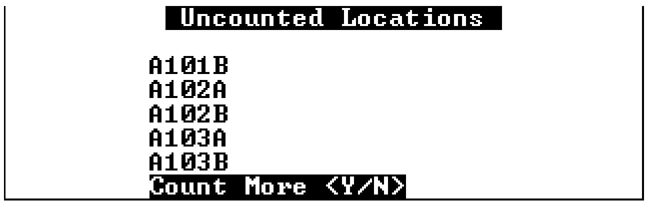
If the system quantity displayed is not correct press “N” on the scanner keypad. The screen shown in figure 12 will be displayed. At this point, you can enter the correct quantity for the pallet.



```
Cycle Count

Enter New Quantity
```

Figure 12: Enter New Quantity Screen

5.) Repeat steps 1 through 5 for each location in the zone that is being counted. All locations in a zone should be scanned. If any locations in a particular zone are not scanned, the following screen in figure 13 will be displayed.



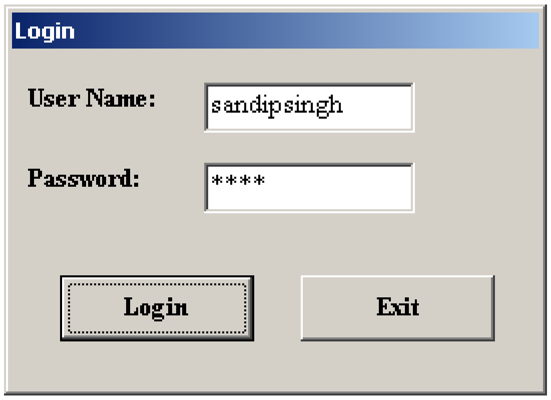
```
Uncounted Locations

A101B
A102A
A102B
A103A
A103B
Count More <Y/N>
```

Figure 13: Uncounted Locations Screen

Pressing “Y” will enable you to scan another location for the current zone. If you do not wish to count any more locations in the current zone, press “N” and exit zone count. Any remaining items found in locations that have not been scanned will be moved to CCLost in the system until they are physically scanned at another point in time.

RF User Maintenance:

This application will be created for the warehouse managers. Warehouse managers are considered administrator of the entire Warehouse Management System. The following login screen requires an account that has access in Flex Process ERP system.

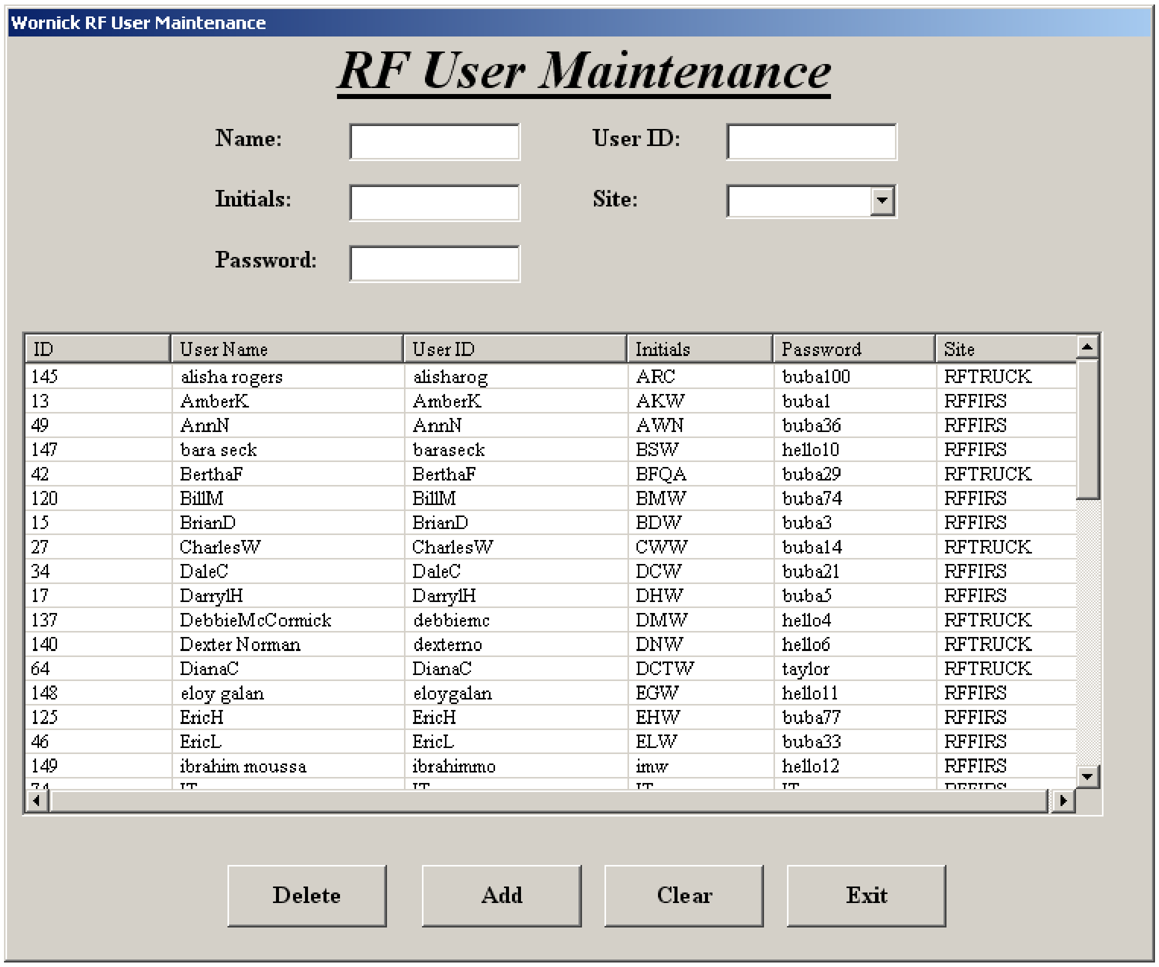


User Name: sandipsingh

Password: ****

Figure 14: Login

After a successful login warehouse managers will be able to change, delete, and add new users in the system to use the RF application.



ID	User Name	User ID	Initials	Password	Site
145	alisha rogers	alisharog	ARC	buba100	RFTRUCK
13	AmberK	AmberK	AKW	buba1	RFFIRS
49	AnnN	AnnN	AWN	buba36	RFFIRS
147	bara seck	baraseck	BSW	hello10	RFFIRS
42	BerthaF	BerthaF	BFQA	buba29	RFTRUCK
120	BillM	BillM	BMW	buba74	RFFIRS
15	BrianD	BrianD	BDW	buba3	RFFIRS
27	CharlesW	CharlesW	CWW	buba14	RFTRUCK
34	DaleC	DaleC	DCW	buba21	RFFIRS
17	DarrylH	DarrylH	DHW	buba5	RFFIRS
137	DebbieMcCormick	debbiemc	DMW	hello4	RFTRUCK
140	Dexter Norman	dexterno	DNW	hello6	RFTRUCK
64	DianaC	DianaC	DCTW	taylor	RFTRUCK
148	eloy galan	eloygalan	EGW	hello11	RFFIRS
125	EricH	EricH	EHW	buba77	RFFIRS
46	EricL	EricL	ELW	buba33	RFFIRS
149	ibrahim moussa	ibrahimmo	imw	hello12	RFFIRS

Figure 15: User Maintenance

Testing

The Wornick Company has a standard template for functional testing and that same template has been used to get the user to sign off after the unit testing. Following is the example of the testing form that will be created to test this application. The following form will be created for every use case that is described above in the Use Case model. All of these forms must be signed by the user and the development leader for this application to be implemented in the live environment of The Wornick Company.

Wornick Unit Testing Form

Component	**Developer**	**Date of Test**
Self Correcting Inventory Sys.	Sandip K. Singh	02/14/2006

Defect ID's (If any)	**Applicable Version(s)**
	1.0

Workstation Tested On	**Database (Test/Production)**
WSTCLIENT01/Svexch2k	Test

Description of the Test
This application will allow users to count the pallets that reside in a particular zone and location. This application will also help The Wornick Company to correct the discrepancies between the physical inventory and the perpetual inventory. Reports will be generated based on the data that this application collects to help the Warehouse Managers to make necessary adjustments to correct the physical inventory to avoid the total physical count at all the warehouses.

Test Step	**Expected Result**	**Actual Result**
1. Log into the RF system	Default value screen shows up. Make sure that the default site is "WOR". If not, go to "Default Values" and change it.	
2. Select "Zone Count" and hit Enter key	Application prompts user to enter the Zone	
3. Enter "A01" for zone	If valid zone was entered application prompts user to	

	enter the “Location” that needs to be counted	
4. Enter the Location and hit Enter key	All the pallets that were at that location will be moved to users handheld or the truck and will prompt the user for the pallet ID	
5. Start scanning the pallets at that location	Application will display either the confirmation message or an error message based on the transaction	
6. Once the user is done counting that location **a.** If any pallets were not counted but they exist at the specified location will be moved to CCLOST location. **b.** If the pallet was physically at the specified location but not in the system. The information will be recorded and displayed on RF Zone Count report **c.** If the pallet was physically at the specified location but in Flex Process it was at another location. Application will move it to the specified location and this transaction will show up on the RF Zone Count report.		
7. User will hit F4 key to go back to Scan Location screen	Application will prompt the user to count another location	
8. If user wants to continue		

counting follow steps from 4 – 6		
9. If user doesn’t want to continue counting. Hit F4 to complete the zone.	Any uncounted locations under the specified zone will be displayed with the option to allow the user to count more or to end the application.	
10. User hits “Y” to count more	Application will prompt the user to scan another location	
11. User hits “N”	Application ends	

______________________________ ______________________

Test Executed By **Date**

______________________________ ______________________

Development Lead **Date**

Database Diagram

The database design for this application is very complex. The database architecture is provided by the Flex Process ERP system. In some cases the database is extremely normalized and in others extremely de-normalized. I cannot make any changes to the database design and will have to study and understand the entire database structure necessary for this application.

The database design for this application consists of twelve tables. These tables refer to several different objects in Flex Process ERP system including lot, site, inventory inquiry, user defined activity form, area, location. Most of these tables are very big and contain several million rows. If the application experiences any performance degradation, I have analyzed it and create the necessary indexes.

Following is the database design for the Self Correcting Inventory System and also for the administration application.

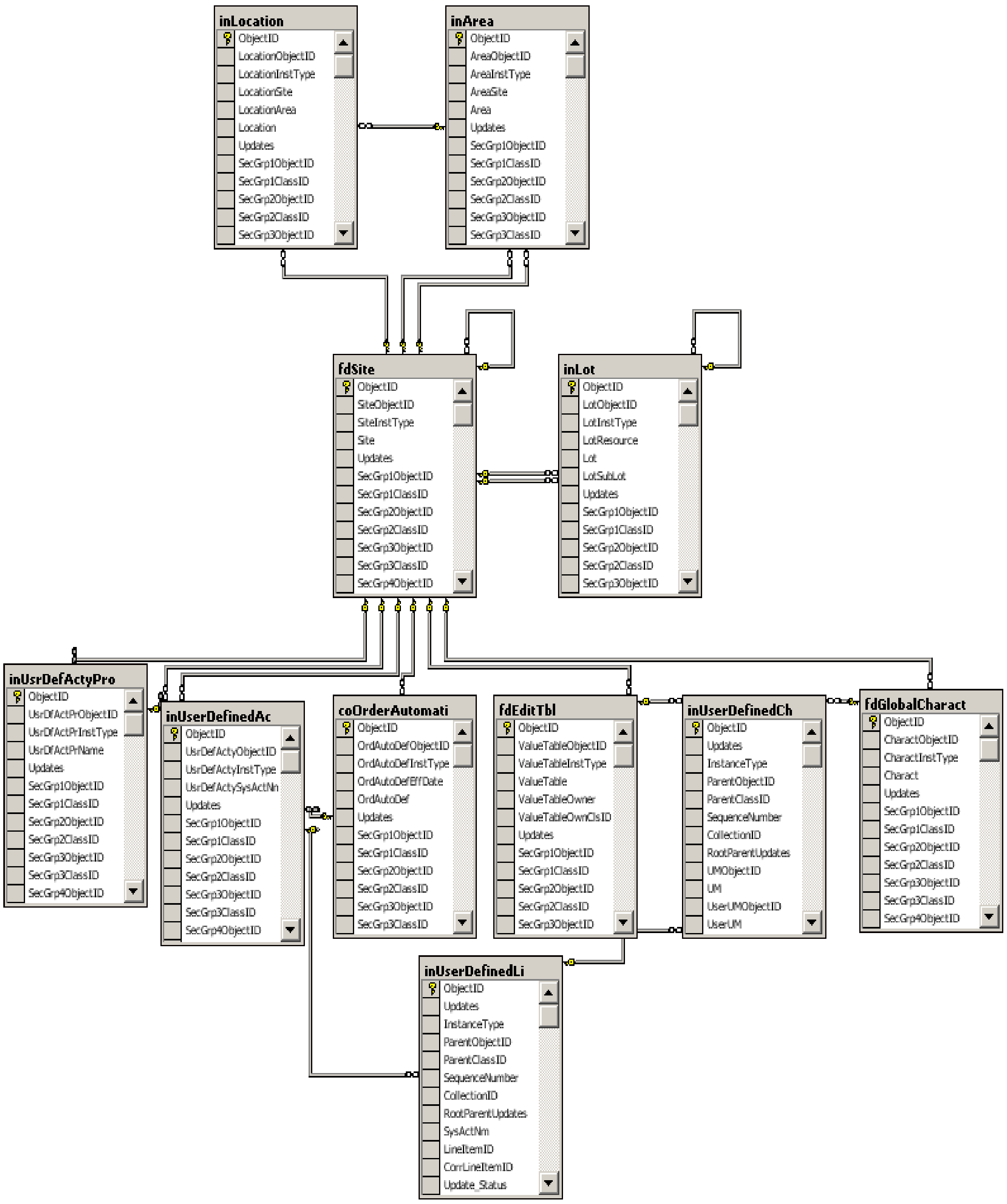


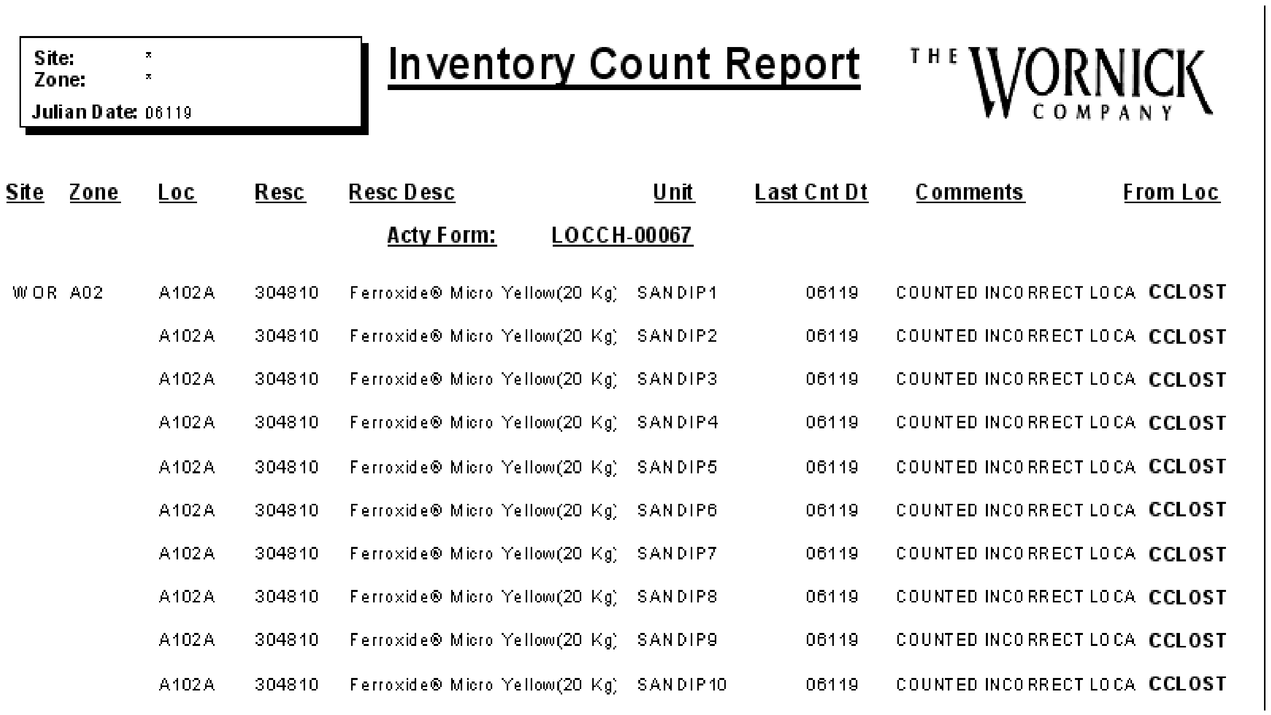
Figure: 16 Database diagram

Reports

This system provides three reports for the user to review. Reports have been designed in Crystal Reports XI.

- Inventory Count Report

Inventory Count Report provides transactional information based on given date, site, location or Zone. It includes every transaction that was performed based on above parameters. The information is grouped by the activity form that was used to perform the following transactions.



Site: *
Zone: *
Julian Date: 06119

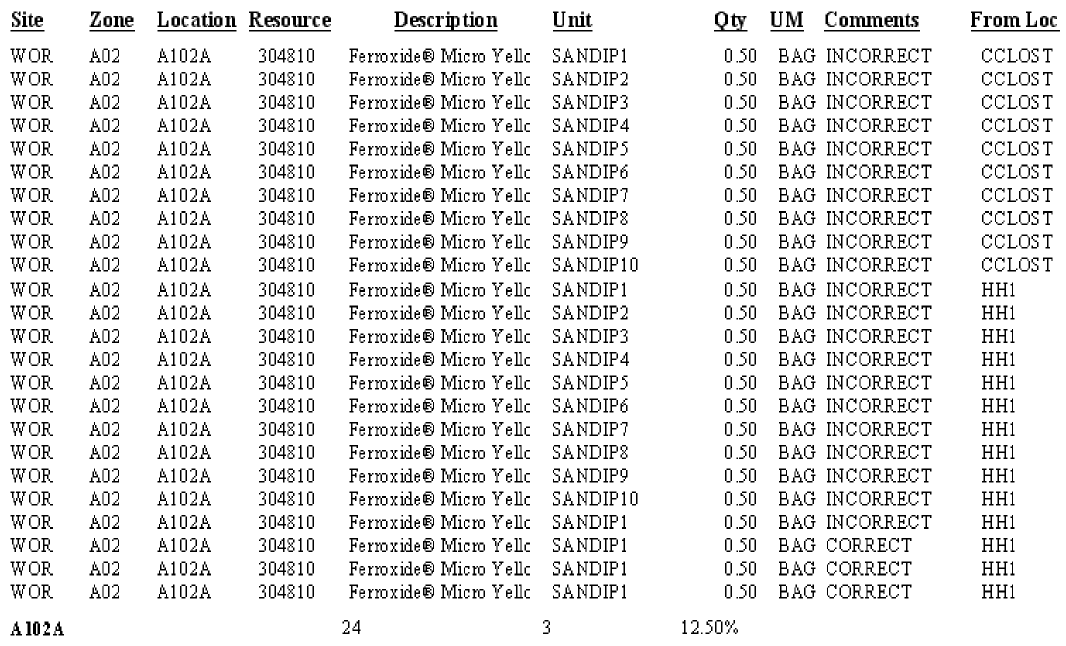
Inventory Count Report

Site	Zone	Loc	Resc	Resc Desc	Unit	Last Cnt Dt	Comments	From Loc
				Acty Form: LOCCH-00067				
WOR	A02	A102A	304810	Ferroxide® Micro Yellow(20 Kg)	SANDIP1	06119	COUNTED INCORRECT LOCA	CCLOST
		A102A	304810	Ferroxide® Micro Yellow(20 Kg)	SANDIP2	06119	COUNTED INCORRECT LOCA	CCLOST
		A102A	304810	Ferroxide® Micro Yellow(20 Kg)	SANDIP3	06119	COUNTED INCORRECT LOCA	CCLOST
		A102A	304810	Ferroxide® Micro Yellow(20 Kg)	SANDIP4	06119	COUNTED INCORRECT LOCA	CCLOST
		A102A	304810	Ferroxide® Micro Yellow(20 Kg)	SANDIP5	06119	COUNTED INCORRECT LOCA	CCLOST
		A102A	304810	Ferroxide® Micro Yellow(20 Kg)	SANDIP6	06119	COUNTED INCORRECT LOCA	CCLOST
		A102A	304810	Ferroxide® Micro Yellow(20 Kg)	SANDIP7	06119	COUNTED INCORRECT LOCA	CCLOST
		A102A	304810	Ferroxide® Micro Yellow(20 Kg)	SANDIP8	06119	COUNTED INCORRECT LOCA	CCLOST
		A102A	304810	Ferroxide® Micro Yellow(20 Kg)	SANDIP9	06119	COUNTED INCORRECT LOCA	CCLOST
		A102A	304810	Ferroxide® Micro Yellow(20 Kg)	SANDIP10	06119	COUNTED INCORRECT LOCA	CCLOST

Figure 17: Inventory Count Report

- Inventory Count Scorecard Report

This report provides similar information as the inventory count report but also gives the ability to the Warehouse Supervisor to review the work and accuracy of the material handler and to find out the accuracy of the worker. In the example below, under the comments column we can find out whether the material handler counted the correct location or incorrect location. For location A102A the accuracy was only 12.5%. Based on this data supervisors are able to determine who is doing a poor job and identify the cause of it.

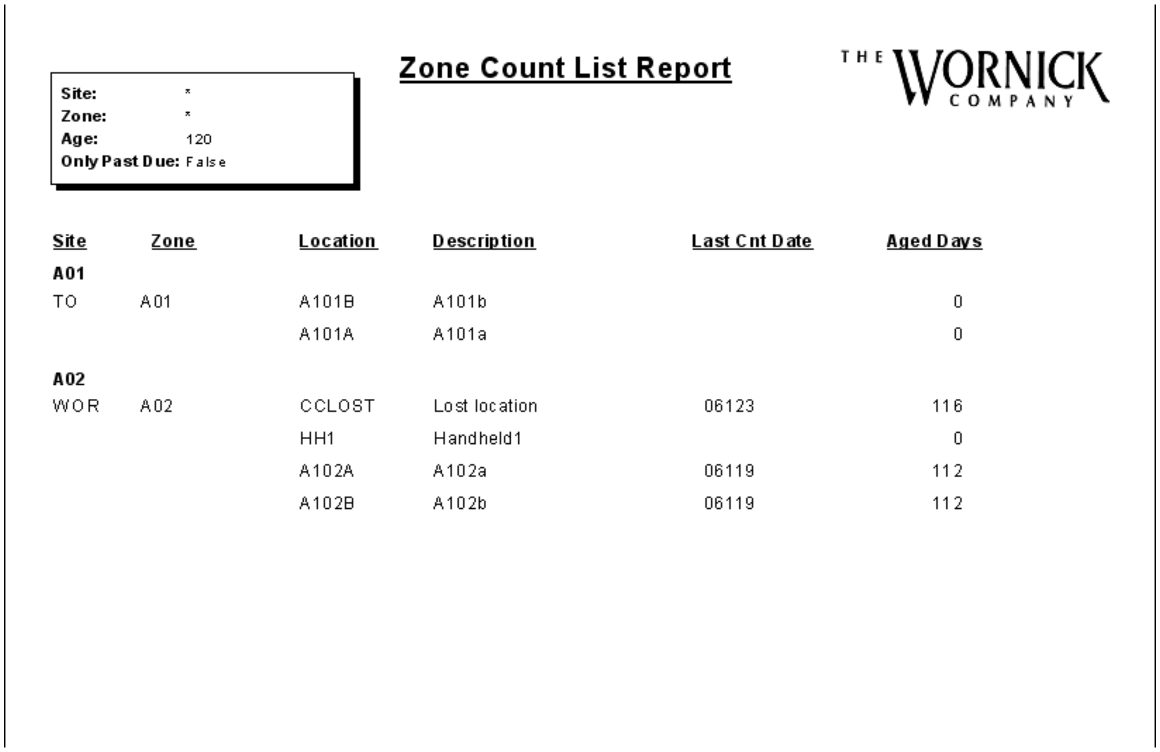


Site	Zone	Location	Resource	Description	Unit	Qty	UM	Comments	From Loc
WOR	A02	A102A	304810	Ferroxide® Micro Yellc	SANDIP1	0.50	BAG	INCORRECT	CCLOST
WOR	A02	A102A	304810	Ferroxide® Micro Yellc	SANDIP2	0.50	BAG	INCORRECT	CCLOST
WOR	A02	A102A	304810	Ferroxide® Micro Yellc	SANDIP3	0.50	BAG	INCORRECT	CCLOST
WOR	A02	A102A	304810	Ferroxide® Micro Yellc	SANDIP4	0.50	BAG	INCORRECT	CCLOST
WOR	A02	A102A	304810	Ferroxide® Micro Yellc	SANDIP5	0.50	BAG	INCORRECT	CCLOST
WOR	A02	A102A	304810	Ferroxide® Micro Yellc	SANDIP6	0.50	BAG	INCORRECT	CCLOST
WOR	A02	A102A	304810	Ferroxide® Micro Yellc	SANDIP7	0.50	BAG	INCORRECT	CCLOST
WOR	A02	A102A	304810	Ferroxide® Micro Yellc	SANDIP8	0.50	BAG	INCORRECT	CCLOST
WOR	A02	A102A	304810	Ferroxide® Micro Yellc	SANDIP9	0.50	BAG	INCORRECT	CCLOST
WOR	A02	A102A	304810	Ferroxide® Micro Yellc	SANDIP10	0.50	BAG	INCORRECT	CCLOST
WOR	A02	A102A	304810	Ferroxide® Micro Yellc	SANDIP1	0.50	BAG	INCORRECT	HH1
WOR	A02	A102A	304810	Ferroxide® Micro Yellc	SANDIP2	0.50	BAG	INCORRECT	HH1
WOR	A02	A102A	304810	Ferroxide® Micro Yellc	SANDIP3	0.50	BAG	INCORRECT	HH1
WOR	A02	A102A	304810	Ferroxide® Micro Yellc	SANDIP4	0.50	BAG	INCORRECT	HH1
WOR	A02	A102A	304810	Ferroxide® Micro Yellc	SANDIP5	0.50	BAG	INCORRECT	HH1
WOR	A02	A102A	304810	Ferroxide® Micro Yellc	SANDIP6	0.50	BAG	INCORRECT	HH1
WOR	A02	A102A	304810	Ferroxide® Micro Yellc	SANDIP7	0.50	BAG	INCORRECT	HH1
WOR	A02	A102A	304810	Ferroxide® Micro Yellc	SANDIP8	0.50	BAG	INCORRECT	HH1
WOR	A02	A102A	304810	Ferroxide® Micro Yellc	SANDIP9	0.50	BAG	INCORRECT	HH1
WOR	A02	A102A	304810	Ferroxide® Micro Yellc	SANDIP10	0.50	BAG	INCORRECT	HH1
WOR	A02	A102A	304810	Ferroxide® Micro Yellc	SANDIP1	0.50	BAG	INCORRECT	HH1
WOR	A02	A102A	304810	Ferroxide® Micro Yellc	SANDIP1	0.50	BAG	CORRECT	HH1
WOR	A02	A102A	304810	Ferroxide® Micro Yellc	SANDIP1	0.50	BAG	CORRECT	HH1
WOR	A02	A102A	304810	Ferroxide® Micro Yellc	SANDIP1	0.50	BAG	CORRECT	HH1
A102A				24	3	12.50%			

Figure 18: Inventory Count Scorecard Report

- Inventory Count List Report

This report generates a list of locations that need to be counted on any given day. The user can provide the number of days it has been since the location was counted and generate this report. The report tells them exactly which site, zone and location they need to count today. The report also displays the aged days since the location was counted.



Zone Count List Report

Site:	*
Zone:	*
Age:	120
Only Past Due:	False

Site	Zone	Location	Description	Last Cnt Date	Aged Days
A01					
TO	A01	A101B	A101b		0
		A101A	A101a		0
A02					
WOR	A02	CCLOST	Lost location	06123	116
		HH1	Handheld1		0
		A102A	A102a	06119	112
		A102B	A102b	06119	112

Figure 19: Inventory Count List Report

Code Comments

Module level and procedure level comments have been implemented in the application. Complex code inside the procedure has also been properly documented.

For example:

```
      ' Purpose:
        '       Creates a user defined activity form in Flex ERP system
        '
        'Input Variables:
        '       ByVal UserActivityProfile As String
        '       ByRef ErrorChain As String
        '       ByVal ReportingDate As Date
        '       ByVal Site As String
        '       ByVal UserReference As String
        '       ByVal Shift As Integer
        '       ByVal OperatorID As String, _
        '       ByRef ActivityNumber As String
        '       ByRef oMyUserActyProfileRef As
                        CUserDefinedActyProfileReference
        '       ByRef oMyUserActySumRef As
                        CUserDefinedActivityReference
        '       ByRef oMyUserActySumMgr As CUserDefinedActivitySummarys
        '       ByRef oMyUserActySum As
                        IPROTEANUserDefinedActivitySummary

        'Output Variables
        '       NONE
        '
        'Return value:
        ' True if successful.
        ' False in case an error occurs.
        '
        'Remarks:
        '       This procedure must succeed for the application to
continue
```

Figure 20: Code Comments

Timeline

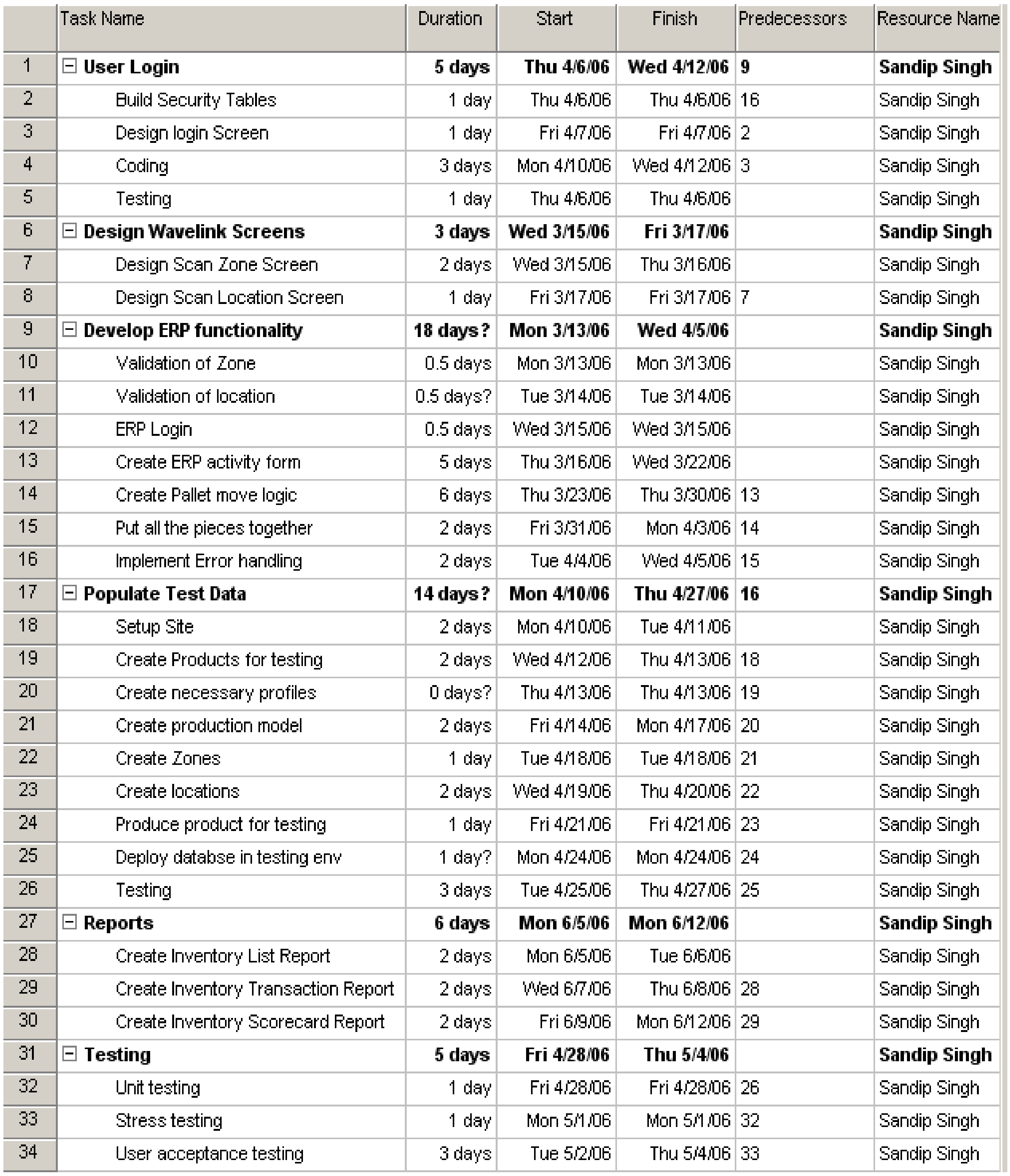
- **Learn Current System:** This part of the project requires commitment from The Wornick Company stakeholders and Mr. Sandip Kumar Singh. During this task, regular meetings will be scheduled to explain in detail the way this application needs to be designed.

- **Preliminary Documentation:** The project proposal will be reviewed and signed by The Wornick Company. After signing the proposal, the project specifications will be developed and sign off will be required from The Wornick Company.

- **Database Design:** The Wornick Company will provide a logical database design of their system to help me understand the system and develop the solution.

- **Application Coding:** This will entail completion of all of the development that is required for this project. In addition, I have developed an application that interfaces with Flex Process ERP system using the API provided by SSA Global.

- **Testing:** Quality assurance will be performed by the developer before the product is released to The Wornick Company. Due to possible conflicts between testing environments, it will be necessary for thorough testing to be performed on site after the product is released, and before the implementation. The Wornick Company employees will be engaged in the user acceptance testing and will be required to sign off on it.



	Task Name	Duration	Start	Finish	Predecessors	Resource Name
1	⊟ **User Login**	**5 days**	**Thu 4/6/06**	**Wed 4/12/06**	**9**	**Sandip Singh**
2	Build Security Tables	1 day	Thu 4/6/06	Thu 4/6/06	16	Sandip Singh
3	Design login Screen	1 day	Fri 4/7/06	Fri 4/7/06	2	Sandip Singh
4	Coding	3 days	Mon 4/10/06	Wed 4/12/06	3	Sandip Singh
5	Testing	1 day	Thu 4/6/06	Thu 4/6/06		Sandip Singh
6	⊟ **Design Wavelink Screens**	**3 days**	**Wed 3/15/06**	**Fri 3/17/06**		**Sandip Singh**
7	Design Scan Zone Screen	2 days	Wed 3/15/06	Thu 3/16/06		Sandip Singh
8	Design Scan Location Screen	1 day	Fri 3/17/06	Fri 3/17/06	7	Sandip Singh
9	⊟ **Develop ERP functionality**	**18 days?**	**Mon 3/13/06**	**Wed 4/5/06**		**Sandip Singh**
10	Validation of Zone	0.5 days	Mon 3/13/06	Mon 3/13/06		Sandip Singh
11	Validation of location	0.5 days?	Tue 3/14/06	Tue 3/14/06		Sandip Singh
12	ERP Login	0.5 days	Wed 3/15/06	Wed 3/15/06		Sandip Singh
13	Create ERP activity form	5 days	Thu 3/16/06	Wed 3/22/06		Sandip Singh
14	Create Pallet move logic	6 days	Thu 3/23/06	Thu 3/30/06	13	Sandip Singh
15	Put all the pieces together	2 days	Fri 3/31/06	Mon 4/3/06	14	Sandip Singh
16	Implement Error handling	2 days	Tue 4/4/06	Wed 4/5/06	15	Sandip Singh
17	⊟ **Populate Test Data**	**14 days?**	**Mon 4/10/06**	**Thu 4/27/06**	**16**	**Sandip Singh**
18	Setup Site	2 days	Mon 4/10/06	Tue 4/11/06		Sandip Singh
19	Create Products for testing	2 days	Wed 4/12/06	Thu 4/13/06	18	Sandip Singh
20	Create necessary profiles	0 days?	Thu 4/13/06	Thu 4/13/06	19	Sandip Singh
21	Create production model	2 days	Fri 4/14/06	Mon 4/17/06	20	Sandip Singh
22	Create Zones	1 day	Tue 4/18/06	Tue 4/18/06	21	Sandip Singh
23	Create locations	2 days	Wed 4/19/06	Thu 4/20/06	22	Sandip Singh
24	Produce product for testing	1 day	Fri 4/21/06	Fri 4/21/06	23	Sandip Singh
25	Deploy databse in testing env	1 day?	Mon 4/24/06	Mon 4/24/06	24	Sandip Singh
26	Testing	3 days	Tue 4/25/06	Thu 4/27/06	25	Sandip Singh
27	⊟ **Reports**	**6 days**	**Mon 6/5/06**	**Mon 6/12/06**		**Sandip Singh**
28	Create Inventory List Report	2 days	Mon 6/5/06	Tue 6/6/06		Sandip Singh
29	Create Inventory Transaction Report	2 days	Wed 6/7/06	Thu 6/8/06	28	Sandip Singh
30	Create Inventory Scorecard Report	2 days	Fri 6/9/06	Mon 6/12/06	29	Sandip Singh
31	⊟ **Testing**	**5 days**	**Fri 4/28/06**	**Thu 5/4/06**		**Sandip Singh**
32	Unit testing	1 day	Fri 4/28/06	Fri 4/28/06	26	Sandip Singh
33	Stress testing	1 day	Mon 5/1/06	Mon 5/1/06	32	Sandip Singh
34	User acceptance testing	3 days	Tue 5/2/06	Thu 5/4/06	33	Sandip Singh

Figure 21: Timeline

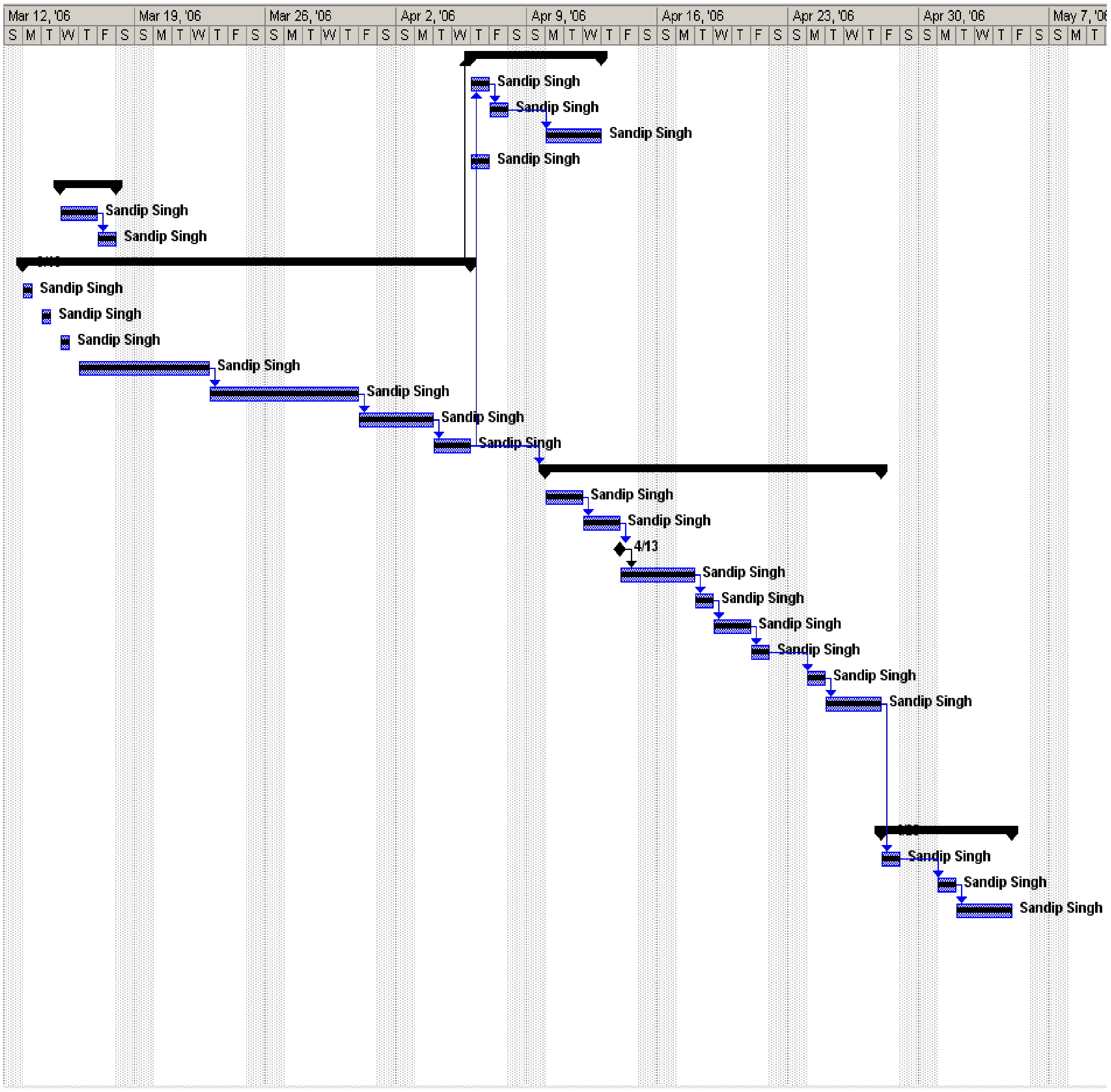
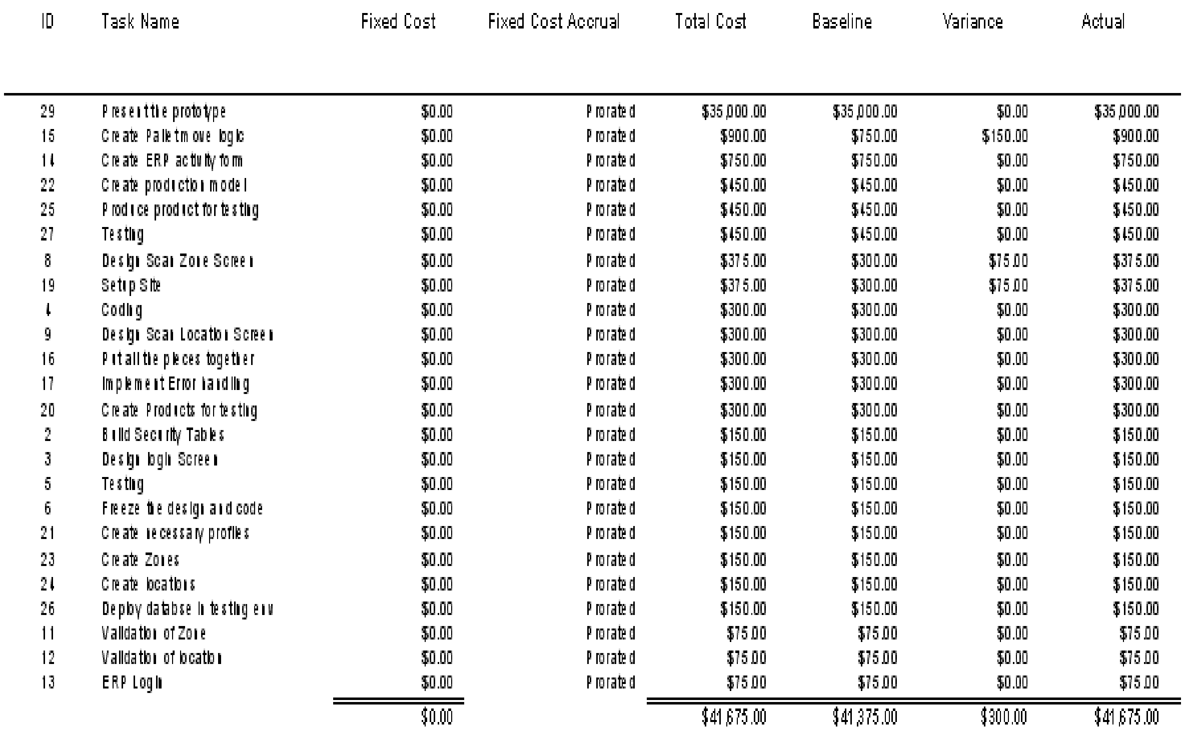


Figure 22: Timeline

Cost

Most of the hardware and software required for this project has already been purchased and installed by The Wornick Company. Following is the list of equipment that the new application will utilize.



ID	Task Name	Fixed Cost	Fixed Cost Accrual	Total Cost	Baseline	Variance	Actual
29	Present the prototype	$0.00	Prorated	$35,000.00	$35,000.00	$0.00	$35,000.00
15	Create Pallet move logic	$0.00	Prorated	$900.00	$750.00	$150.00	$900.00
14	Create ERP activity form	$0.00	Prorated	$750.00	$750.00	$0.00	$750.00
22	Create production model	$0.00	Prorated	$450.00	$450.00	$0.00	$450.00
25	Produce product for testing	$0.00	Prorated	$450.00	$450.00	$0.00	$450.00
27	Testing	$0.00	Prorated	$450.00	$450.00	$0.00	$450.00
8	Design Scan Zone Screen	$0.00	Prorated	$375.00	$300.00	$75.00	$375.00
19	Setup Site	$0.00	Prorated	$375.00	$300.00	$75.00	$375.00
4	Coding	$0.00	Prorated	$300.00	$300.00	$0.00	$300.00
9	Design Scan Location Screen	$0.00	Prorated	$300.00	$300.00	$0.00	$300.00
16	Put all the pieces together	$0.00	Prorated	$300.00	$300.00	$0.00	$300.00
17	Implement Error handling	$0.00	Prorated	$300.00	$300.00	$0.00	$300.00
20	Create Products for testing	$0.00	Prorated	$300.00	$300.00	$0.00	$300.00
2	Build Security Tables	$0.00	Prorated	$150.00	$150.00	$0.00	$150.00
3	Design login Screen	$0.00	Prorated	$150.00	$150.00	$0.00	$150.00
5	Testing	$0.00	Prorated	$150.00	$150.00	$0.00	$150.00
6	Freeze the design and code	$0.00	Prorated	$150.00	$150.00	$0.00	$150.00
21	Create necessary profiles	$0.00	Prorated	$150.00	$150.00	$0.00	$150.00
23	Create Zones	$0.00	Prorated	$150.00	$150.00	$0.00	$150.00
24	Create locations	$0.00	Prorated	$150.00	$150.00	$0.00	$150.00
26	Deploy databse in testing env	$0.00	Prorated	$150.00	$150.00	$0.00	$150.00
11	Validation of Zone	$0.00	Prorated	$75.00	$75.00	$0.00	$75.00
12	Validation of location	$0.00	Prorated	$75.00	$75.00	$0.00	$75.00
13	ERP Login	$0.00	Prorated	$75.00	$75.00	$0.00	$75.00
		$0.00		$41,675.00	$41,375.00	$300.00	$41,675.00

Figure 23: Cost

Deliverables

I will be providing a well-designed and easy-to-use system; data integrity is one of my main concerns because the system will affect many different functionalities of the Enterprise Resource Planning.

1. Login functionality will be provided in the application. Login screens will run on top of the custom designed SQL server tables.
2. Database design will be provided to the client.
3. Easy-to-navigate user interface will be provided. All the UI's will be designed based on the client specifications.
4. Reports will be provided for the better visibility of the data. This feature will also help the Warehouse Managers to make adjustments to the inventory.
5. UI's will have validations for all the critical input from the users to minimize errors.
6. This solution will make real time inventory correction possible and minimize the loss of time and material.
7. This application will match the system inventory with the physical inventory.
8. Reports will be created for the Warehouse Managers for review.
9. This application will eliminate the mandatory physical count for the entire warehouse at the end of the year.
10. User manual will be provided.
11. Code will be commented based on the client's standard. Technical documentation will also be provided.

12. User maintenance will be provided to the Warehouse Manager to maintain the users that are allowed to use the wireless application. Application will be written in VB.Net and SQL Server 2000 will host the database to provide the back-end support.
13. Application will be accessible from the handheld devices and the desktop computer via an emulator.
14. All the Enterprise Resource Planning business rules will be followed.

Conclusions and Recommendations

Self correction inventory system was developed to satisfy the current needs of The Wornick Company. This application will help the company to resolve several issues with their inventory system. It was very exiting project for me. During the development of this application I learned a lot about project management and also extended my technical knowledge.

Appendix A

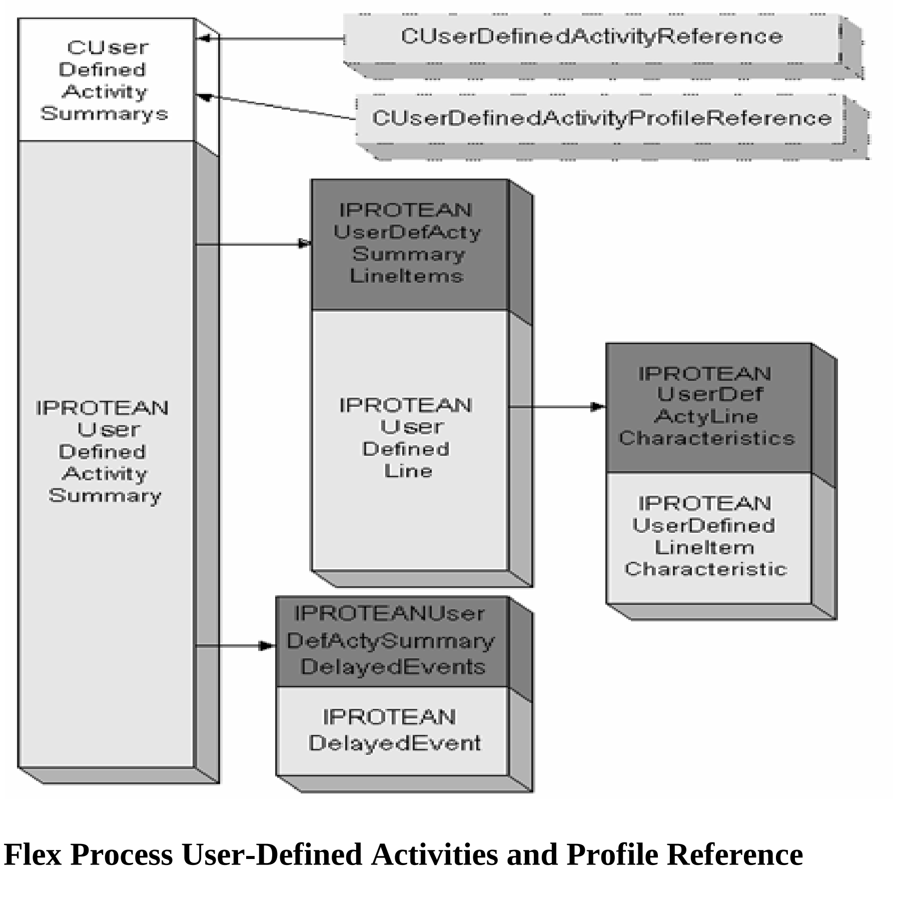
This appendix describes all the Flex Process classes that are used to manipulate the ERP system.

User-Defined Activities and Profiles OLE Classes

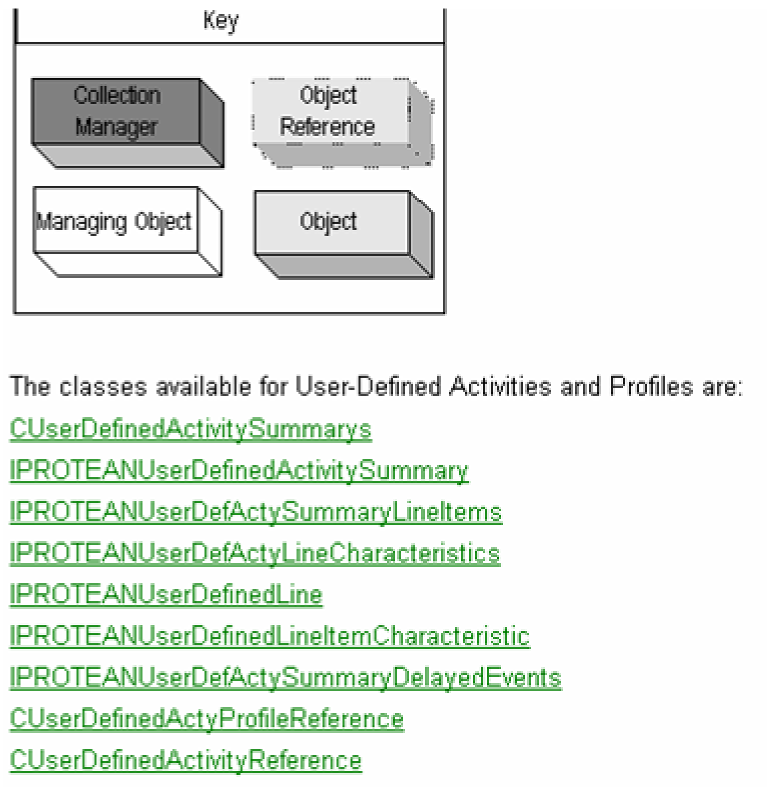
You enter user-defined activities to record changes in inventory placement or status. This may include any of the following:

- inventory movement, such as a change in site, area, and location
- inventory transition into/from a cost center, such as between a location and cost center or between cost centers
- change of classification

User-defined activity requires one or more user-defined activity profiles that determine rules for activity reporting and aspects such as field defaults, layout, inclusion, and restrictions.


CUser Defined Activity Summarys
CUserDefinedActivityReference
CUserDefinedActivityProfileReference
IPROTEAN User Defined Activity Summary
IPROTEAN UserDefActy Summary LineItems
IPROTEAN User Defined Line
IPROTEAN UserDef ActyLine Characteristics
IPROTEAN UserDefined LineItem Characteristic
IPROTEANUser DefActySummary DelayedEvents
IPROTEAN DelayedEvent

Flex Process User-Defined Activities and Profile Reference



The classes available for User-Defined Activities and Profiles are:

CUserDefinedActivitySummarys
IPROTEANUserDefinedActivitySummary
IPROTEANUserDefActySummaryLineItems
IPROTEANUserDefActyLineCharacteristics
IPROTEANUserDefinedLine
IPROTEANUserDefinedLineItemCharacteristic
IPROTEANUserDefActySummaryDelayedEvents
CUserDefinedActyProfileReference
CUserDefinedActivityReference

1. CUserDefinedActivitySummary class has the following methods

Method Name	**Comments**
Close	
CreateObjectReference	
CreateWith	
Execute	
IsAvailableForProcessing	
OpenForChange	
OpenForReview	
Remove	
RemoveByReference	
Save	
Validate	

CUserDefinedActivitySummary class

2. IPROTEANUserDefinedActivitySummary Interface Class

Used to enter user-defined activities to record changes in inventory placement or status, including:

- inventory movement, such as a change in site, area, and location

- inventory transition into/from a cost center, such as between a location and cost center or between cost centers
- change of classification of the product

Getting the UserDefActySummaryLineItems property creates an object that manages lines in the user-defined activity form (IPROTEANUserDefinedLine objects). This object also requires a user-defined activity profile, which determines rules for activity reporting and aspects such as field defaults, layout, inclusion, and restrictions. The user-defined profile is also not created with OLE automation.

Create using the following class

Managing Class: **CUserDefinedActivitySummarys**

Properties:

Property Name	Datatype	Comments
ActivityDate	CDateTime	
AddOns	CAddOns	
AdministrativeSite	CSiteReference	
AuditState	LONG	
AutomationFailureFlag	CBoolean	
CheckedOutDate	CDateTime	
Comments	BSTR	
Creator	BSTR	
CreatorID	BSTR	
CurrentEditor	BSTR	
CurrentEditorID	BSTR	
DateCreated	CDateTime	
Description	BSTR	
Description2	BSTR	
FiscalDate	CDateTime	
FiscalPeriodData	BSTR	
FiscalSubperiodData	BSTR	
FiscalYearData	BSTR	
LastActionIndex	CUnsignedInteger	
LastEditDate	CDateTime	
LastEditor	BSTR	
LastEditorID	BSTR	
Modified	BOOL	
ObjectReference	CUserDefinedActivity Reference	
OpenByDEP	CBoolean	
OperatorID	BSTR	
OrderAutoDesign	COrderAutomRefere nce	

OverrideAllUserEventsToDelayed	CBoolean	
RelatedToTriggeringObjectClassID	CLong	
RelatedToTriggeringObjectString	BSTR	
ReportingDate	CDateTime	
ReportingShift	CUnsignedInteger	
ReportingSite	CSiteReference	
SkipDEPNotify	CBoolean	
TargetTriggeringObjectClassID	CUnsignedLong	
TargetTriggeringObjectObjectID	CUnsignedLong	
TransferList	CTransferListReference	
Updates	long	
UserCommentString	BSTR	
UserDefActySummaryDelayedEvents		
IPROTEANUserDefActySummaryDelayedEvents		
UserDefActySummaryLineItems		
IPROTEANUserDefActySummaryLineItems		
UserDefinedActivityProfile	CUserDefinedActyProfileReference	
UserReferenceNumber	BSTR	

3. IPROTEANUserDefActySummaryLineItems Interface Class

Manges IPTOREANUserDefinedActivityLine objects. The ProteanUserDefActySummar LineItems object is created by getting the UserDefActySummaryLineItems property on the IPROTEANUserDefinedActivitySummary object.

Create using the following class

Property that creates this class: **UserDefActySummaryLineItems**

Methods:

Method Name	Comment
Add	
Item	
Remove	

Properties:

Property Name	Data Type
Count	Integer

4. IPROTEANUserDefActyLineCharacteristics Interface Class

Manages the IPROTEANUserDefinedLineItemCharacteristic object. The IPROTEANUserDef ActyLineCharacteristics object is created by getting the UserDefLineItemCharacteristics property on the IPROTEANUserDefinedActivityLine object.

Create Using:

Property that creates this class: **UserDefActyLineCharacteristics**

Methods:

Method Name	Comment
Item	

Properties:

Property Name	Data Type	Comment
Count	Integer	

5. IPROTEANUserDefinedLine Interface Class

Corresponds to a line in an activity form for user defined activities. IPROTEANUserDefined Line objects are managed by the IPROTEANUserDefActySummaryLineItems object.

Getting the UserDefActyLineCharacteristics property creates a manager for an IPROTEANUserDefinedLineItemCharacteristic object.

The From Site and To Site are based on the sites specified in the Resource and ToResource properties. In order to set the From Site correctly, make sure that the CResourceReference SiteName Property for the reference for the Resource is set to the correct From Site. In order to set the To Site correctly, make sure that the CResourceReference SiteName Property for the reference for the ToResource is set to the correct To Site.

Create Using:

Methods on **IPROTEANUserDefActySummaryLineItems**

Methods:

Method Name	Comment
ChangeAll	

CopyLineItem	
GenerateUnits	
GetAlternateFactor	
GetAlternateQuantity	
GetPrimaryQuantity	
GetQuantityCalcFlag	
RetrieveCharacteristics	
RetrieveDefaults	
SetAlternateFactor	
SetAlternateQuantity	
SetPrimaryQuantity	
SetQuantityCalcFlag	
SuspendLineItem	

<u>Properties</u>:

Property Name	**Data Type**	**Comment**
AgeDate1	CDateTime	
AgeDate2	CDateTime	
AgeDate3	CDateTime	
AgeDate4	CDateTime	
AgeDate5	CDateTime	
Classification	CClassificationReference	
Comments	BSTR	
ContainerID	BSTR	
CorrLineID	BSTR	
CostCenter	CCostCenterReference	
CreateLooseQuantity	CBoolean	
DateCode	BSTR	
Errors	CBoolean	
ETDDate	CDateTime	
ExpirationDate	CDateTime	
GrossQuantity	CQuantity	
HasSpecialReqmts	CBoolean	
InstructionType	LONG	
LineID	BSTR	
Location	CLocationReference	
Lot	CLotReference	
ManufacturingDate	CDateTime	
MiscField1	BSTR	
MiscField2	CDouble	
MiscField3	BSTR	
MiscField4	CDouble	
OriginalLineID	BSTR	

PickingOverrideSequence	CUnsignedInteger	
ReasonCode	CReasonCodeReference	
Resource	CResourceReference	
ResourceDescription	BSTR	
RetestDate	CDateTime	
RevisionLevel	BSTR	
RotationDate	BSTR	
SampleNumber	BSTR	
ScaleAdjustmentQuantity	CQuantity	
Status	LONG	
TareQuantity	CQuantity	
ToCostCenter	CCostCenterReference	
ToLocation	CLocationReference	
ToLot	CLotReference	
ToResource	CResourceReference	
ToResourceDescription	BSTR	
ToUnitID	BSTR	
ToUnitType	CUnitTypeReference	
UnitID	BSTR	
UnitType	CUnitTypeReference	
UpdateGeneralLedger	CBoolean	
Updates	long	
UserDefActyLineCharacteristics		
IPROTEANUserDefActyLineCharacteristics		
UserDefinedActivityType	CUserDefinedTransTypeRef	
VehicleID	BSTR	

6. IPROTEANUserDefinedLineItemCharacteristic Interface Class

Holds information about a characteristic for this IPROTEANUserDefinedLine object. The characteristic you assign to this object must already exist as a CCharacteristicReference.
The IPROTEANUserDefinedLineItemCharacteristic object is managed by the IPROTEANUserDefActyLineCharacteristics object.

Create Using:

Methods on **IPROTEANUserDefActyLineCharacteristics**

Methods:

Method Name	**Comment**
None	

Properties:

Property Name	**Data Type**	**Comment**
AlphanumericValue	BSTR	
BaseNumericValue	CDouble	
BaseUM	CUMReference	
BooleanValue	CBoolean	
Characteristic	CCharacteristicReference	
InventoryCharacteristic	CBoolean	
LimitedTOValueTable	CBoolean	
LimittoValueTable	BOOL	
NumericValue	CDouble	
Operator	LONG	
UM	CUMReference	
Updates	long	
ValueTable	CEditTableReference	
ValueType	LONG	

7. IPROTEANUserDefActySummaryDelayedEvents Interface Class

Note: This class is reserved for Internal Use Only by FlexProcess Development.

Create Using:

Property that creates this class: **UserDefActySummaryDelayedEvents**

Methods:

Method Name	**Comment**
Item	
Move	
Remove	

Properties:

Property Name	**Data Type**	**Comment**
Count	Integer	

8. CUserDefinedActyProfileReference Interface Class

Uniquely identifies a user-defined activity profile.User defined activities are used on the IPROTEANUserDefinedActivitySummary object.

Create Using:

PROTEAN.UserDefinedActyProfileReference

<u>Methods</u>:

Method Name	**Comment**
Copy	
IsNull	
Locate	
SetNull	

<u>Properties</u>:

Property Name	**Data Type**	**Comment**
DisplayName	BSTR	
InstanceType	LONG	
ProfileName	BSTR	

9. CUserDefinedActivityReference Interface Class

Uniquely identifies an IPROTEANUserDefinedActivitySummary object.The CUserDefinedActivitySummarys object manages IPROTEANUserDefinedActivitySummary objects.

Create Using:

PROTEAN.UserDefinedActivityReference

<u>Methods</u>:

Method Name	**Comment**
Copy	
IsNull	
Locate	
SetNull	

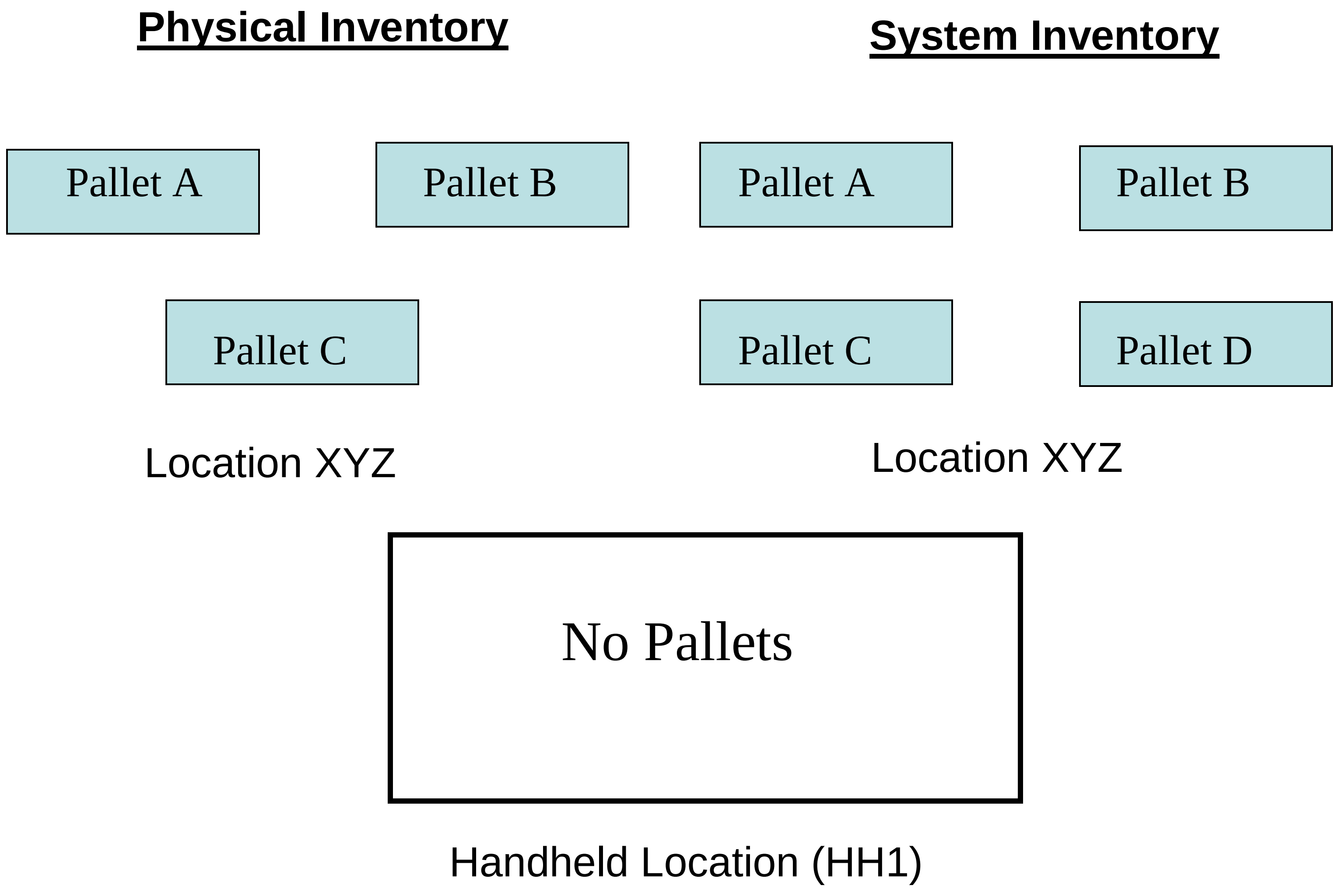
<u>Properties</u>:

Property Name	Data Type	Comment
DisplayName	BSTR	
InstanceType	LONG	
SystemReferenceNumber	BSTR	

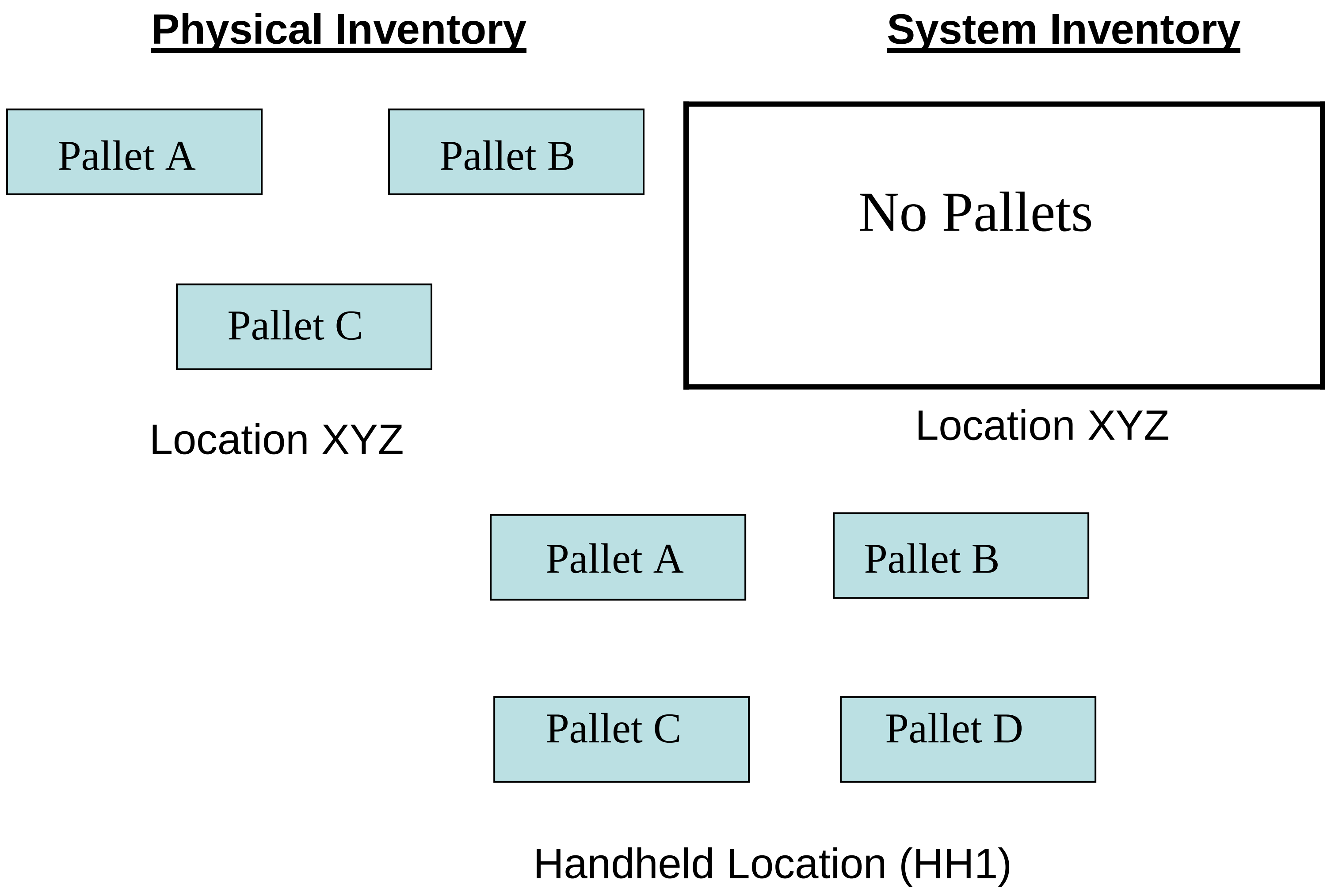
The following screen shots are created for better understanding of the purpose of this application. This will also help the user to follow the application flow. There are two scenarios in this application and I will be discussing them via the following pictures.

Scenario I

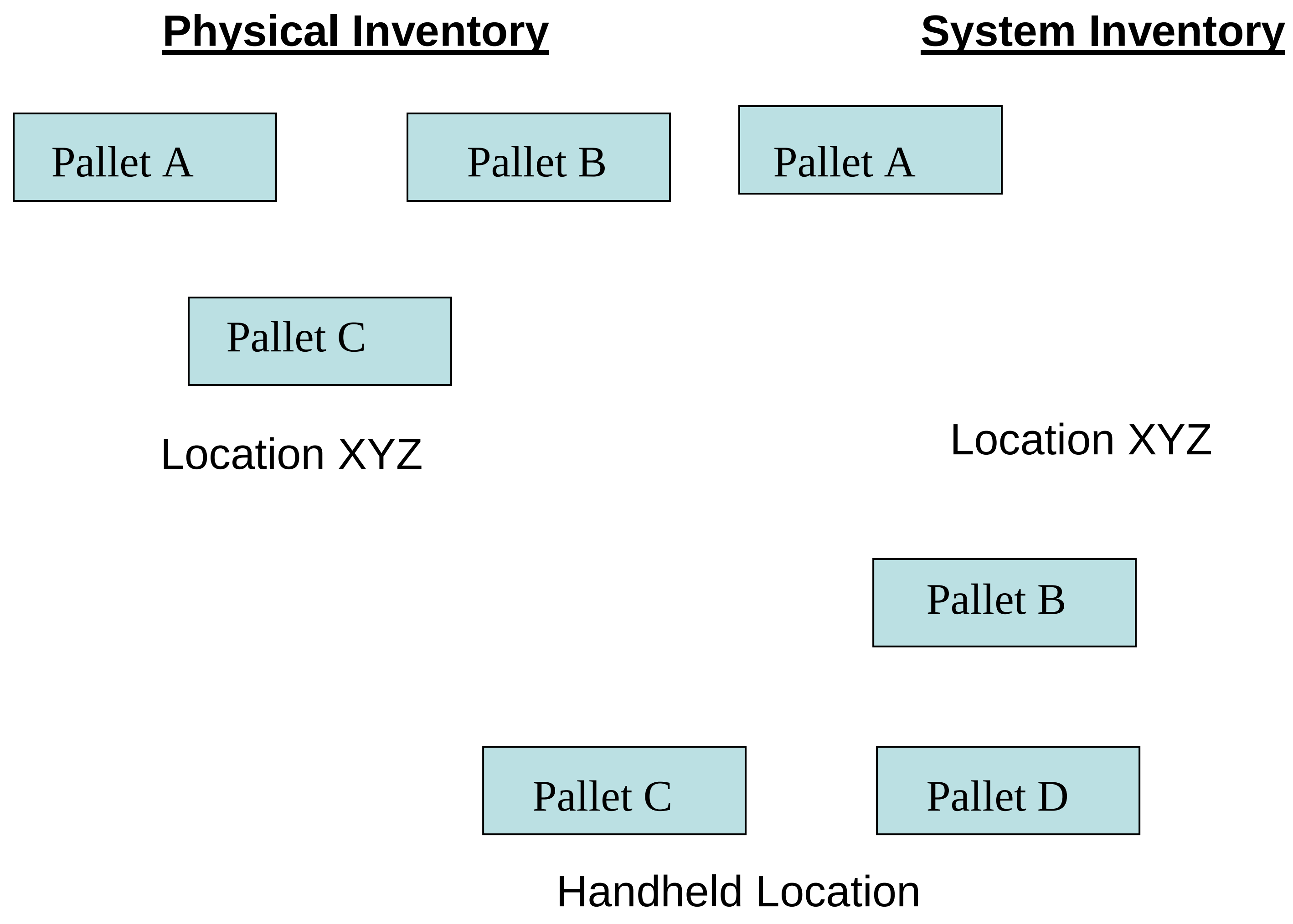
This scenario will describe the application flow when there is a discrepancy between the system inventory and the physical inventory and there are more pallets in the system and less in the physical location. As the user counts the location, the system inventory and physical inventory will be matched.



In the above case, the system shows that there are four pallets in XYZ location. Physically, there are only three pallets. As the user scans location XYZ all the pallets will be moved from XYZ location to the handheld (HH1) location. This move is described in the following screen shot.

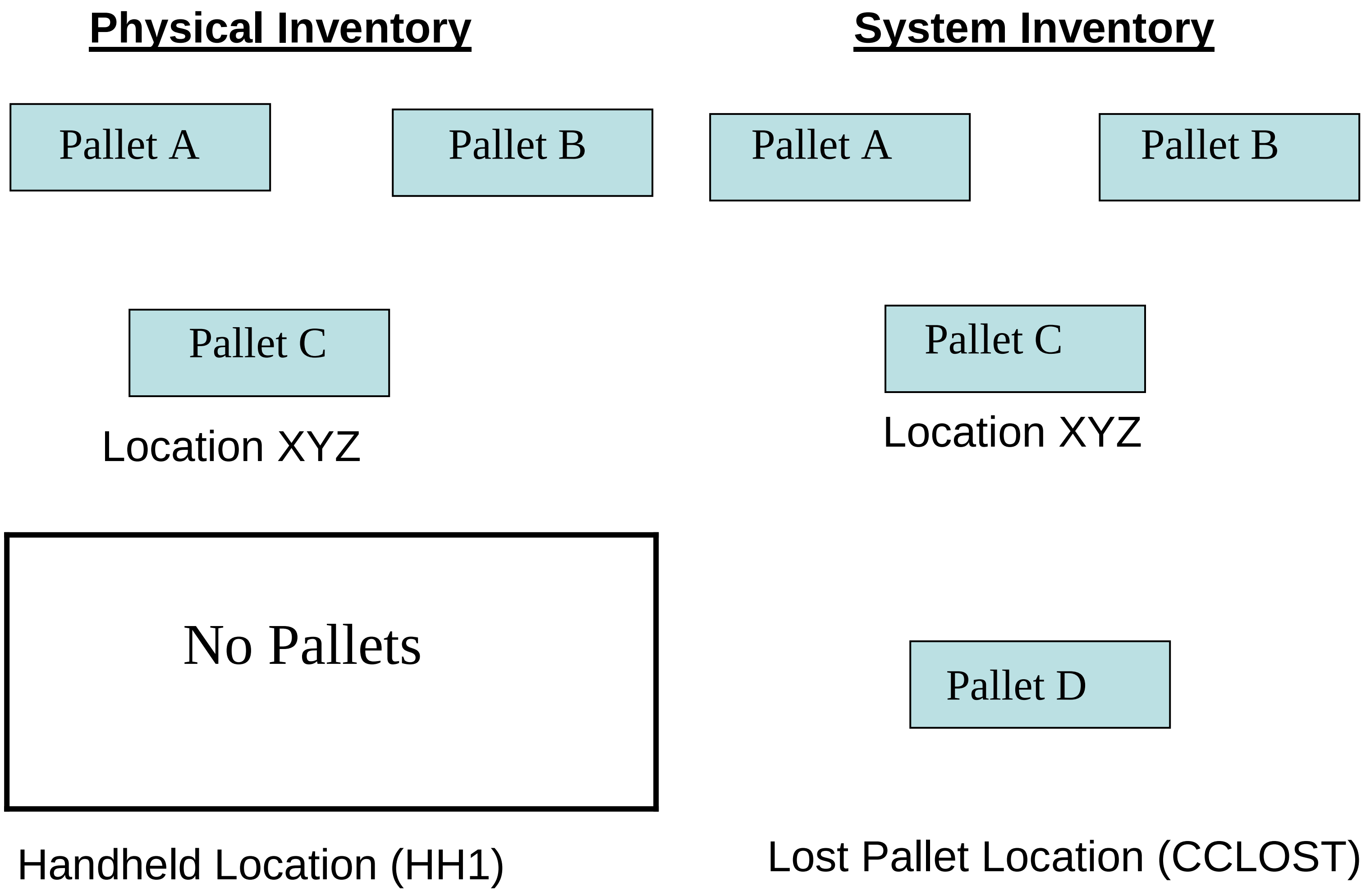


As the user starts scanning the physical pallets (A, B, C etc), the pallets will be moved back to the original location ABC.



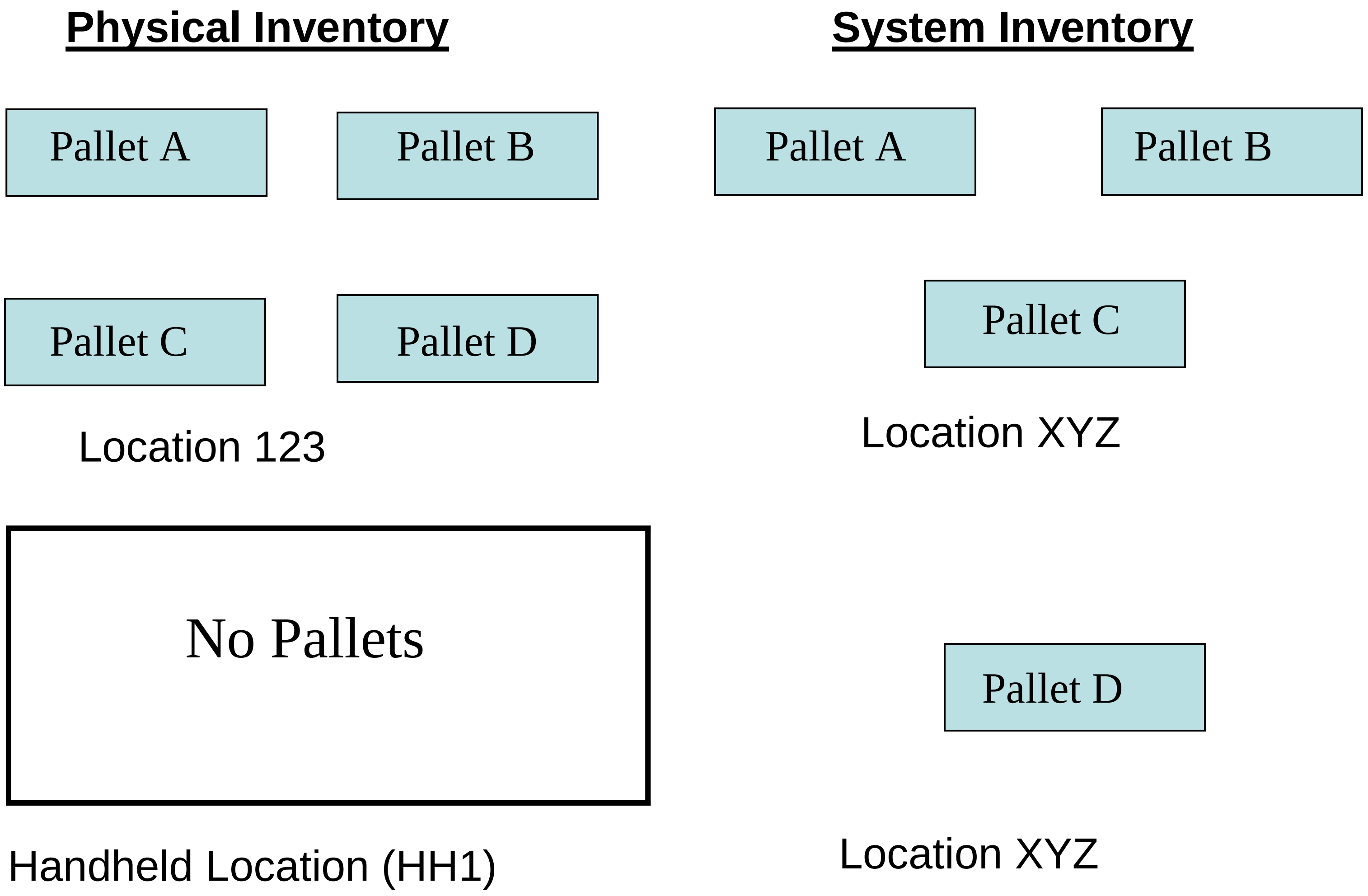
User scanned pallets A and it was moved from the handheld location to its original location (XYZ). The user continues to scan the physical pallets and they keep moving back the location XYZ.

At this point the application has moved pallets A, B, and C back to the original location but we still have Pallet D sitting on the handheld. Since pallet D is physically not at that location the user will not be able to scan it and is unaware that there is still a pallet residing at the handheld location. The user will exit out of the system thinking that he/she is done counting that location. The application will move the pallet to fictitious location "CCLOST". All the pallets sitting in the lost location will be accounted for later by using this application or manually by the warehouse supervisor.



Scenario II

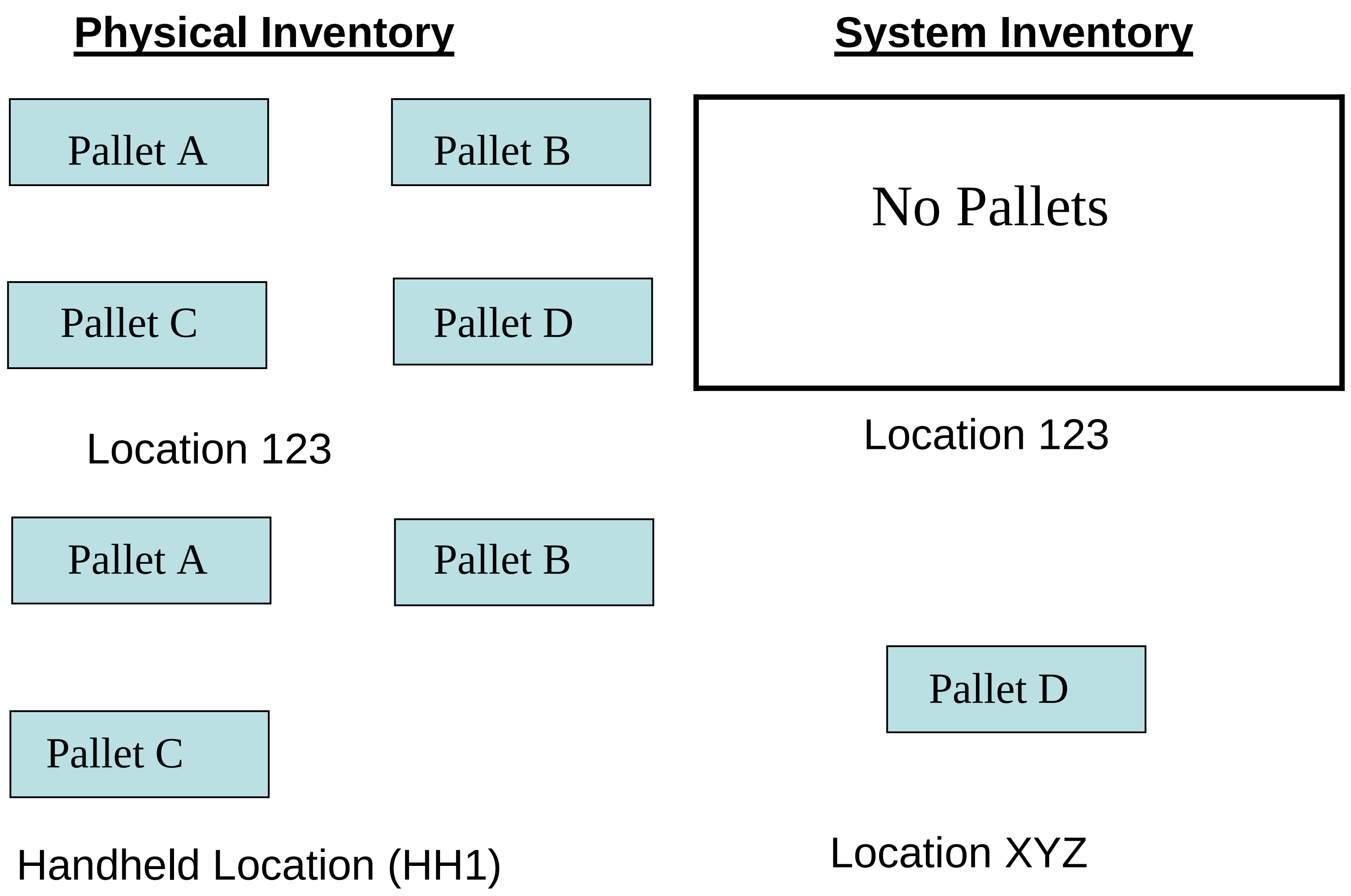
This scenario will describe the application flow when there is a discrepancy between the system inventory and the physical inventory and there are more pallets in the physical location and less in the system. As the user counts the location, the system inventory and physical inventory will be matched.



In the above case, pallet D is physically located in location 123 but in the system it resides at location XYZ.

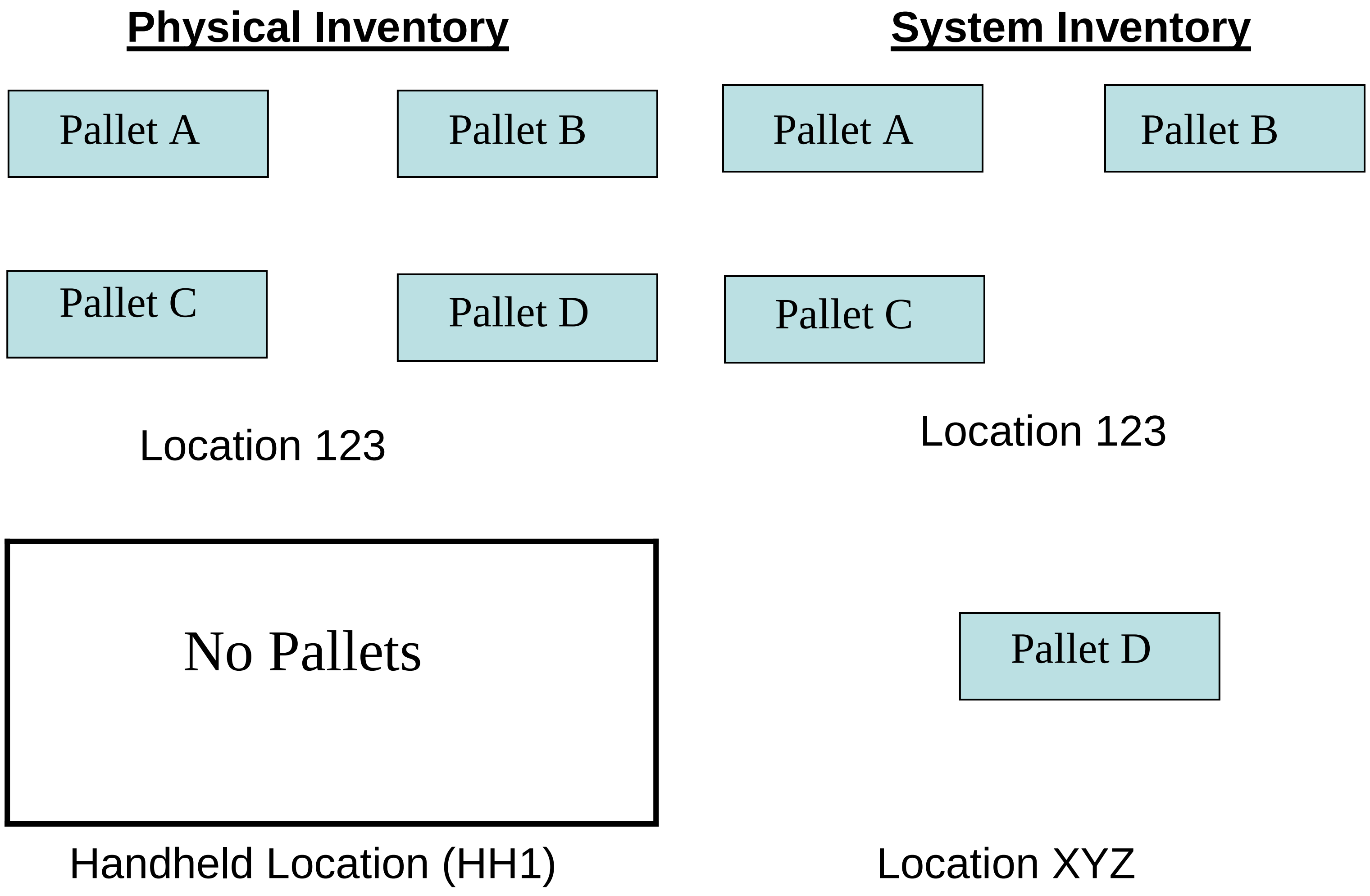
As the user scans location 123, pallet A, B, and C will be moved to handheld location.

This move is described in the following slide.

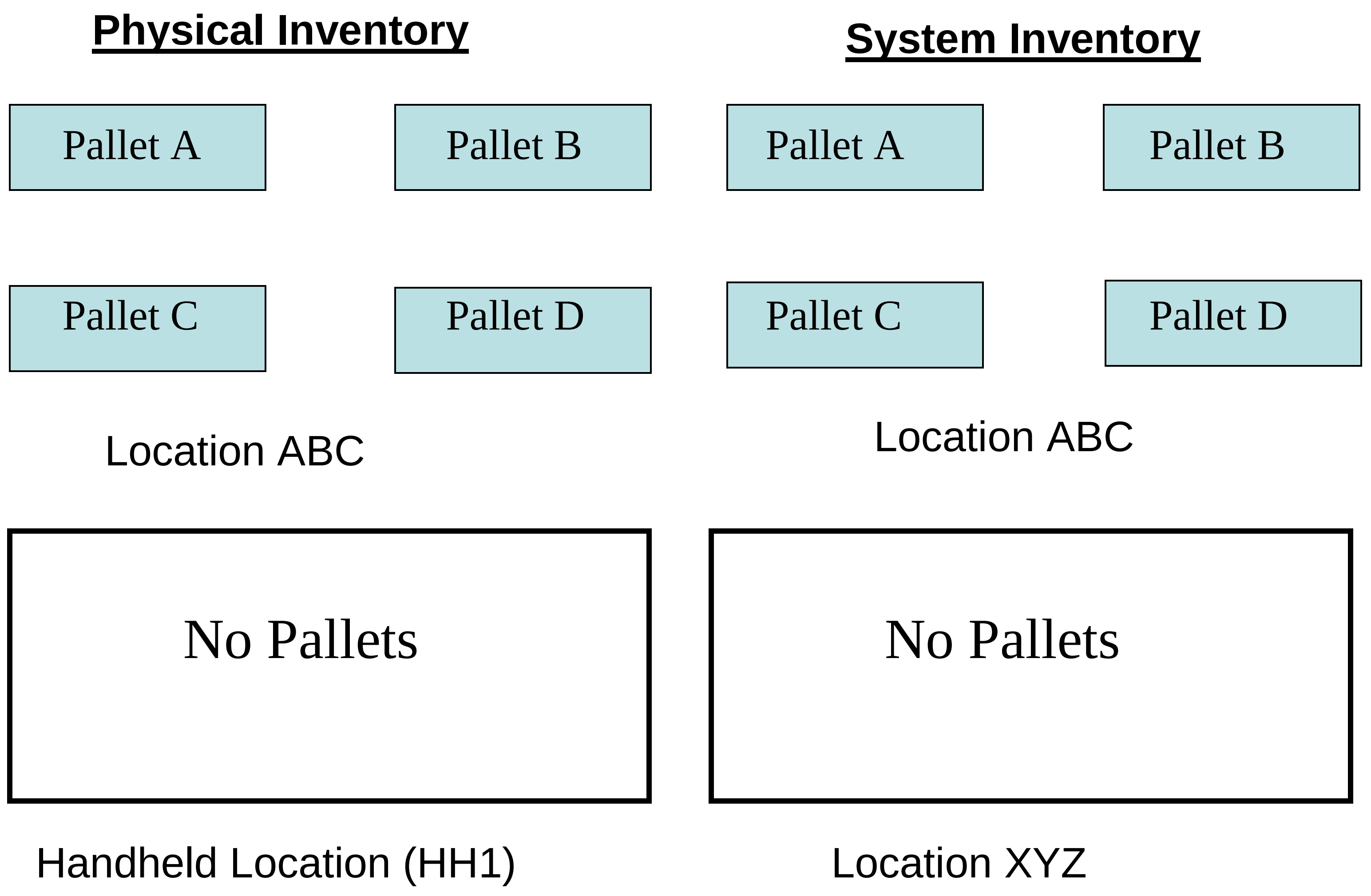


Pallet A, B, and C have been moved to handheld location by the application.

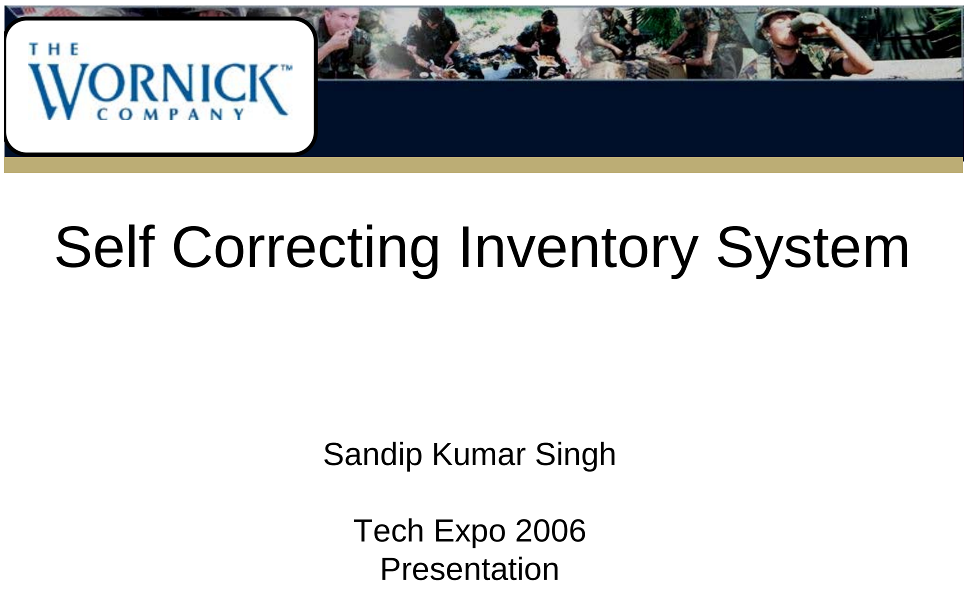
User scanned pallets A, B, and C they were moved from the handheld location to its original location (123).



As soon as the user scans Pallet D from the its physical location (ABC), the application will search the entire inventory in the system to try to find Pallet D. The application will find that Pallet D is actually located in XYZ location but physically it is in ABC location. Application will move Pallet D from XYZ to ABC and thus the physical inventory and system inventory.


Physical Inventory
System Inventory
Pallet A
Pallet B
Pallet A
Pallet B
Pallet C
Pallet D
Pallet C
Pallet D
Location ABC
Location ABC
No Pallets
No Pallets
Handheld Location (HH1)
Location XYZ

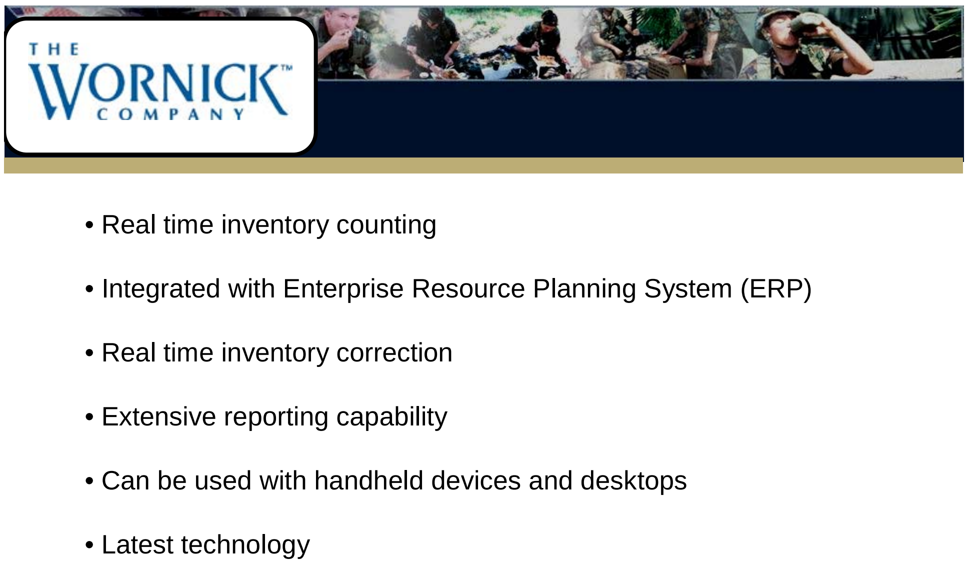
Tech Expo Presentation



Self Correcting Inventory System

Sandip Kumar Singh

Tech Expo 2006
Presentation



- Real time inventory counting
- Integrated with Enterprise Resource Planning System (ERP)
- Real time inventory correction
- Extensive reporting capability
- Can be used with handheld devices and desktops
- Latest technology

Company Information

• Headquarters in Blue Ash, OH

• Sister company in McAllen, TX

• Leading provider of Meals Ready To Eat (MRE’s) for United States Military

• Established in 1985

• Employs around 500 permanent and 1000 temporary employees

Area of Need

• Seeking automated system to count inventory in several warehouses

• Integration with the existing ERP system is a **MUST**

• Avoid manual inventory count at the end of year to satisfy audit requirements

• Simplified user interface on Flex Process ERP system

• Real time transactions

• Present meaningful data via crystal reports to the user

The Wornick Company
4701 Creek Rd
Suite 200
Cincinnati, OH 45242
Tel: (513) 552-7400
Fax: (513) 552-7600

www.wornick.com

UNIVERSITY OF Cincinnati

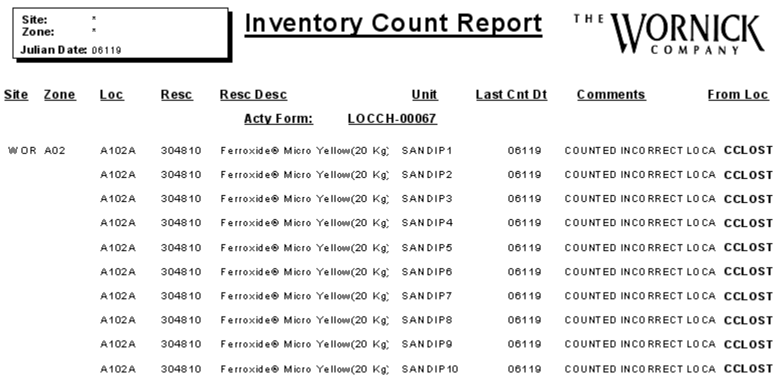
College Of Applied Science

Self correcting inventory system eliminates all the discrepancies in your inventory. This product is being developed for The Wornick Company to address the inventory issues they are experiencing due to human errors. When material handlers misplace product in the warehouse it cannot be found and an inventory adjustment is made.

This causes discrepancies between the perpetual and physical inventory. The misplaced product is reported as lost in the system. Material handlers are forced to write the transactions and manually enter them in the system. Self correcting inventory system is a wireless application and will allow the users to count the inventory in real time. Reports will be generated for Warehouse Supervisors for review.

This product interfaces with **Flex Process ERP** (Enterprise Resource Planning) system and utilizes ERP API's (Application Program Interface) to populate the database.

Advanced reporting with Crystal Reports XI



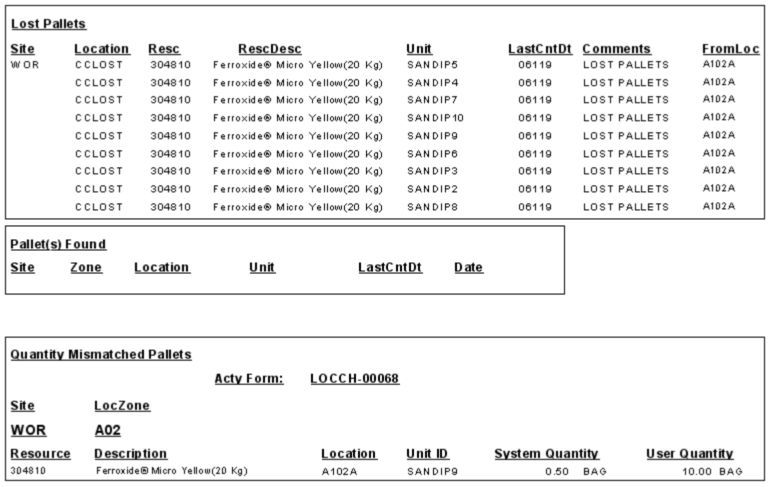
Site: *
Zone: *
Julian Date: 06119

Inventory Count Report

THE WORNICK COMPANY

Acty Form: **LOCCH-00067**

Site	Zone	Loc	Resc	Resc Desc	Unit	Last Cnt Dt	Comments	From Loc
WOR	A02	A102A	304810	Ferroxide® Micro Yellow(20 Kg)	SANDIP1	06119	COUNTED INCORRECT LOCA	CCLOST
		A102A	304810	Ferroxide® Micro Yellow(20 Kg)	SANDIP2	06119	COUNTED INCORRECT LOCA	CCLOST
		A102A	304810	Ferroxide® Micro Yellow(20 Kg)	SANDIP3	06119	COUNTED INCORRECT LOCA	CCLOST
		A102A	304810	Ferroxide® Micro Yellow(20 Kg)	SANDIP4	06119	COUNTED INCORRECT LOCA	CCLOST
		A102A	304810	Ferroxide® Micro Yellow(20 Kg)	SANDIP5	06119	COUNTED INCORRECT LOCA	CCLOST
		A102A	304810	Ferroxide® Micro Yellow(20 Kg)	SANDIP6	06119	COUNTED INCORRECT LOCA	CCLOST
		A102A	304810	Ferroxide® Micro Yellow(20 Kg)	SANDIP7	06119	COUNTED INCORRECT LOCA	CCLOST
		A102A	304810	Ferroxide® Micro Yellow(20 Kg)	SANDIP8	06119	COUNTED INCORRECT LOCA	CCLOST
		A102A	304810	Ferroxide® Micro Yellow(20 Kg)	SANDIP9	06119	COUNTED INCORRECT LOCA	CCLOST
		A102A	304810	Ferroxide® Micro Yellow(20 Kg)	SANDIP10	06119	COUNTED INCORRECT LOCA	CCLOST



Lost Pallets

Site	Location	Resc	RescDesc	Unit	LastCntDt	Comments	FromLoc
WOR	CCLOST	304810	Ferroxide® Micro Yellow(20 Kg)	SANDIP5	06119	LOST PALLETS	A102A
	CCLOST	304810	Ferroxide® Micro Yellow(20 Kg)	SANDIP4	06119	LOST PALLETS	A102A
	CCLOST	304810	Ferroxide® Micro Yellow(20 Kg)	SANDIP7	06119	LOST PALLETS	A102A
	CCLOST	304810	Ferroxide® Micro Yellow(20 Kg)	SANDIP10	06119	LOST PALLETS	A102A
	CCLOST	304810	Ferroxide® Micro Yellow(20 Kg)	SANDIP9	06119	LOST PALLETS	A102A
	CCLOST	304810	Ferroxide® Micro Yellow(20 Kg)	SANDIP6	06119	LOST PALLETS	A102A
	CCLOST	304810	Ferroxide® Micro Yellow(20 Kg)	SANDIP3	06119	LOST PALLETS	A102A
	CCLOST	304810	Ferroxide® Micro Yellow(20 Kg)	SANDIP2	06119	LOST PALLETS	A102A
	CCLOST	304810	Ferroxide® Micro Yellow(20 Kg)	SANDIP8	06119	LOST PALLETS	A102A

Pallet(s) Found

Site	Zone	Location	Unit	LastCntDt	Date

Quantity Mismatched Pallets

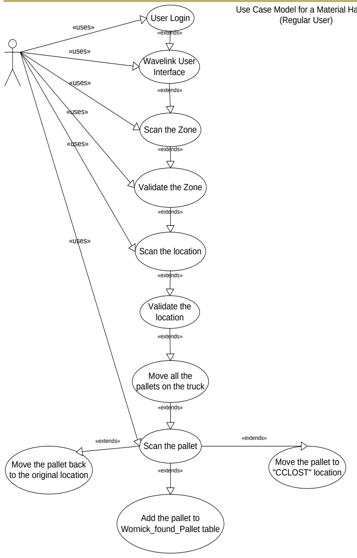
Acty Form: **LOCCH-00068**

Site	LocZone
WOR	**A02**

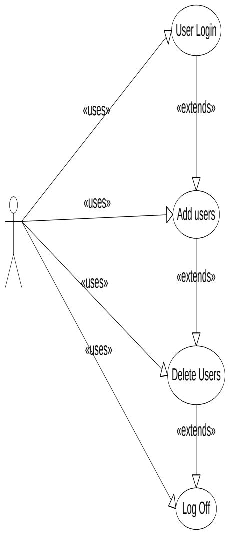
Resource	Description	Location	Unit ID	System Quantity	User Quantity
304810	Ferroxide® Micro Yellow(20 Kg)	A102A	SANDIP9	0.50 BAG	10.00 BAG

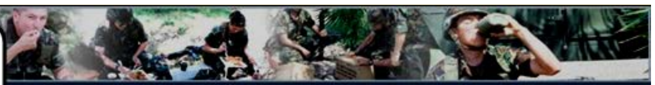
Use Cases

Use Case Model for a Material Handler (Regular User)

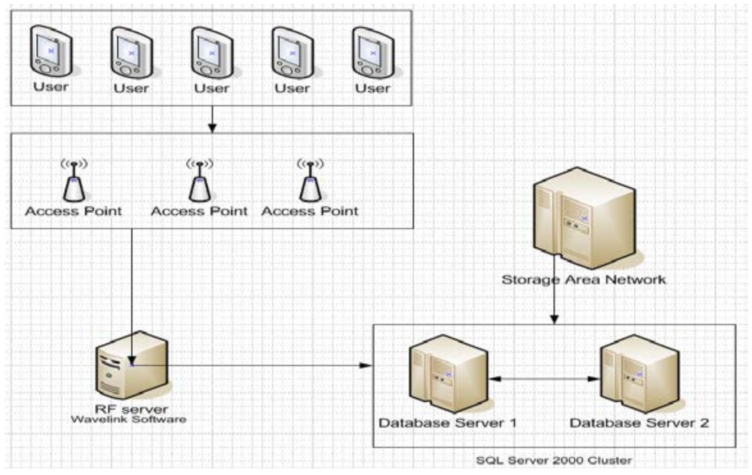

User Login
«uses»
«extends»
Wavelink User Interface
Scan the Zone
Validate the Zone
Scan the location
Validate the location
Move all the pallets on the truck
Scan the pallet
Move the pallet back to the original location
Move the pallet to "CCLOST" location
Add the pallet to Wornick_found_Pallet table

Use case model for supervisors

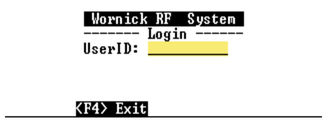




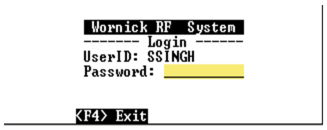
Technical Architecture



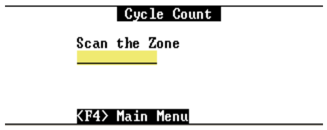
User Interface



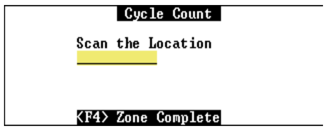
```
Wornick RF  System
------- Login ------
UserID:

<F4> Exit
```



```
Wornick RF  System
------- Login ------
UserID: SSINGH
Password:

<F4> Exit
```

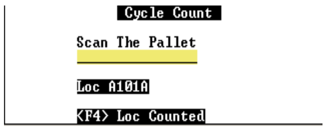


```
Cycle Count

Scan the Zone

<F4> Main Menu
```

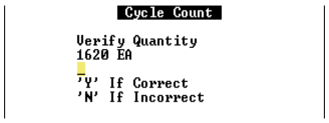


```
Cycle Count

Scan the Location

<F4> Zone Complete
```

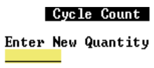
User Interface



```
Cycle Count

Scan The Pallet

Loc A101A
<F4> Loc Counted
```

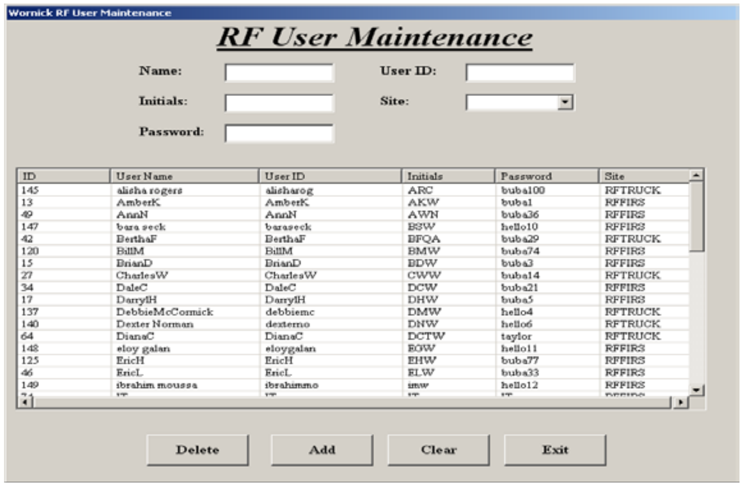


```
Cycle Count

Verify Quantity
1620 EA

'Y' If Correct
'N' If Incorrect
```



```
Cycle Count

Enter New Quantity
```

User Interface



RF User Maintenance

Name:
User ID:
Initials:
Site:
Password:

ID	User Name	User ID	Initials	Password	Site
145	alisha rogers	alisharog	ARC	buba100	RFTRUCK
13	AmberK	AmberK	AKW	buba1	RFFIRS
49	AnnN	AnnN	AWN	buba36	RFFIRS
147	bara seck	baraseck	BSW	hello10	RFFIRS
42	BerthaF	BerthaF	BFQA	buba29	RFTRUCK
120	BillM	BillM	BMW	buba74	RFFIRS
15	BrianD	BrianD	BDW	buba3	RFFIRS
27	CharlesW	CharlesW	CWW	buba14	RFTRUCK
34	DaleC	DaleC	DCW	buba21	RFFIRS
17	DarrylH	DarrylH	DHW	buba5	RFFIRS
137	DebbieMcCormick	debbiemc	DMW	hello4	RFTRUCK
140	Dexter Norman	dexterno	DNW	hello6	RFTRUCK
64	DianaC	DianaC	DCTW	taylor	RFTRUCK
148	eloy galan	eloygalan	EGW	hello11	RFFIRS
125	EricH	EricH	EHW	buba77	RFFIRS
46	EricL	EricL	ELW	buba33	RFFIRS
149	ibrahim moussa	ibrahimmo	imw	hello12	RFFIRS

References:

1. Hanco Enterprise specializes in RF development.
http://www.hanco-ent.com

2. SSA Global is the provider of the ERP system.
http://www.ssaglobal.com

3. Symbol devices (handhelds, truck mounts, and scanners) will be used.
http://www.symbol.com

4. Wavelink COM software will be used to make the communication possible between the handheld devices and the RF server.
http://www.wavelink.com

5. Prof. Russ McMahon
6. Mr. Jason Watson
7. Ms. Kimberly Harmeyer
8. Mr. Manuel Cota