

최 종
연구보고서

모바일 컴퓨팅을 이용한 산지유통 전문조직의 유통관리시스템 개발

A Development of Marketing Management
System of Producer Marketing Associations
Using Mobile Computing Technologies

연구 기관

안양대학교
(주) 이지팜

농 립 부

최 종
연구보고서

모바일 컴퓨팅을 이용한 산지유통 전문조직의 유통관리시스템 개발

A Development of Marketing Management
System of Producer Marketing Associations
Using Mobile Computing Technologies

연구 기관

안양대학교
(주) 이지팜

농 립 부

제 출 문

농림부 장관 귀하

본 보고서를 “모바일 컴퓨팅을 이용한 산지유통 전문조직의 유통관리시스템 개발” 과제의 최종보고서로 제출합니다.

2004년 10월 일

주관연구기관명 : 안양대학교

총괄연구책임자 : 김동환

연 구 원 : 변신의

연 구 원 : 한국택

연 구 원 : 조양호

협동연구기관명 : (주)이지팜

협동연구책임자 : 김한민

연 구 원 : 정윤용

연 구 원 : 장진웅

연 구 원 : 김평송

연 구 원 : 권준수

연 구 원 : 윤미경

요 약 문

I. 제목

모바일 컴퓨팅을 이용한 산지유통전문조직의 유통관리시스템 개발

II. 연구개발의 목적 및 필요성

1. 연구개발의 목적

본 과제의 목적은 산지유통의 핵심체로 육성하고자 하는 산지유통전문조직의 수행업무에 모바일 컴퓨팅 기술을 접목하여 생산성 향상 및 효율성 제고를 위한 현실적인 대안을 제시할 수 있는 유통관리시스템을 개발하는데 있다.

2. 연구개발의 필요성

정부는 '98년까지 농산물 규격출하 촉진자금지원, 생산자 유통지원사업 및 품목별 생산자조직육성지원을 실시, '99년에는 각종 산지유통관련 유사사업을 통합하여 주산지 중심, 전략품목에 대해 집중 지원, 2000년에는 산지유통 개혁방안에 따라 각종 운영자금을 통합 지원하여 산지유통 활성화 사업을 전개해 오고 있다.

선정된 전문조직은 공동출하, 품목별 전문화, 해당 품목 전국협의회 가입 및 지역협의회 결성, 공동계산, 팔렛트 출하 등이 높은 조직으로 ① 공동선별, 공동수송, 공동판매를 통한 유통효율 증진 ② 조합당 유통사업 취급품

목을 3개 품목 이내로 제한, 주산지화·전문화 촉진 ③ 지역협의회 운영활성화를 통해 자율적인 수급조절 능력 및 시장교섭력을 강화하는데 목적이 있다.

따라서 주산지화·전문화와 공동선별, 공동수송, 공동판매 및 자율적인 수급조절과 시장교섭력을 증진시키기 위한 하나의 대안으로써 정보통신기술(ICT)를 도입하여 산지유통전문조직의 유통 효율성과 수급조절 및 시장교섭력을 제고할 필요가 있다.

Ⅲ. 연구개발 내용 및 범위

1. 산지유통전문조직의 모바일 컴퓨팅 환경도입을 위한 기초연구와 시스템 평가 및 활용방안

○ 기존 산지유통전문조직의 정보화 실태 조사 및 분석을 통한 향후 시스템 도입 방안 연구

- 산지유통전문조직의 정보화 현황 및 실태 분석
- 해외의 산지유통 정보화 및 향후 정보화 방향 분석
- 국내·외 모바일 컴퓨팅기술의 적용 사례 조사

○ 산지유통전문조직 구축 시스템의 효과 및 활용방안 연구

- 시스템 구축효과 조사 및 분석
- 시스템 구축 후 활용방안 연구

2. 모바일 컴퓨팅을 이용한 산지유통전문조직의 유통관리시스템 개발

○ 모바일 컴퓨팅기술을 이용한 농산물 생산·입고에서부터 출하·판매에 이르기까지 입고관리, 저장관리, 선별/포장관리, 채고관리, 출고관리 등의 원격지 데이터를 수집이 가능한 유통관리시스템 구축

- 입·출고관리, 저장관리, 선별/포장관리 등 모듈별 'Real time information' 생성 및 'Information availability'시스템 어플리케이션 개발
 - 입고관리 : 등급·규격별 입고 등록/조회
 - 저장관리 : 등급·규격별 저장고 입고 등록/조회
 - 선별, 포장관리 : 선별 포장 등록/조회, 미선별 재고관리, 포장 재고관리
 - 출고관리 : 출고내역 등록/조회, 상품 재고관리

IV. 연구개발 결과 및 활용에 대한 건의

1. 연구개발 결과

농산물 산지유통은 영세한 수집상에 의해 주도되어 정보화가 제대로 이뤄지지 않고 있으며, 작목반, 농협, 영농조합법인 등 생산자조직의 정보화도 미진한 실정이고, 상인들은 법인화되지 않은 개인사업자로서 규모가 영세하고 자료의 공개를 극도로 꺼려 정보화를 회피하고 있다.

산지유통전문조직은 투자여력이 부족하고, 정보화에 대한 필요성을 크게 인식하지 못하는 등 규모의 영세성으로 인해 정보화 마인드가 상당히 낮은 수준이다. 아울러 정보의 분석에 따른 경영의 과학화가 미진하여 출하결과를 포함한 수집 정보를 분석 및 데이터베이스화하여 출하처 및 출하량 선정 등 판매정책을 효과적으로 수립하는 등의 전략적인 업무수행이 미비한 실정이다.

산지유통전문조직은 on-line 유통체계의 개선을 위해서 전자수발주시스템, 공급망관리(supply chain management)등 정보기술 활용 및 전자상거래 기반조성으로 효율성을 제고해야 하며, ERP시스템, 모바일시스템 등의

도입으로 산지의 정보화를 촉진시켜야한다.

따라서 산지유통전문조직은 모바일 정보시스템을 개발, 이용하여 운영의 효율성을 높이고 유통정보 제공 등 부가기능을 수행할 필요가 있다. 모바일시스템은 산지유통전문조직의 종사자들에게 있어서 언제 어디서나 데이터베이스에 접근하여 원하는 정보를 등록, 수정, 삭제, 조회가 가능하며 체계적인 관리 및 분석이 가능하게 하고, 전반적인 유통업무 환경의 질적 개선을 통해 생산성 향상과 효율성 증대를 꾀할 수 있으며, 데이터의 실시간 처리 및 고객 대응력 제고 등 산지유통전문조직의 경쟁력과 통제력을 향상시킬 수 있는 획기적인 대안이 될 것이다.

산지유통전문조직의 모바일 시스템은 시스템 계획, 시스템 분석, 시스템 설계, 구현, 유지·보수 및 평가 등의 개발 프로세스를 거쳐 개발되었다. 구매입고 프로세스는 구매입고 조회, 구매입고 등록/수정 등의 하부 프로세스로 구성되어 있고, 선별관리 프로세스는 선별관리 조회, 선별관리 등록/수정 등의 하부프로세스로 구성되었다. 선별상세관리 프로세스는 선별상세 조회, 선별상세 등록/수정 등의 하부 프로세스로 구성되었으며, 포장관리 프로세스는 포장관리 조회, 포장관리 등록/수정 등의 하부프로세스로 구성되었다. 마지막으로 출하관리 프로세스는 출하관리 조회, 출하관리 등록/수정 등의 하부프로세스로 구성되었다.

2. 활용에 대한 건의

본 연구를 통해 향후 산지유통전문조직의 유통업무 관리가 용이해 질뿐만 아니라 데이터를 기반으로 경영이 가능해 진다. 특히 산지유통전문조직뿐만 아니라 농산물 가공업체, 식품업체, 유통업체 등에 확산 가능하다. 다만 본 산출물인 정보시스템을 도입하기 위해서 각 조직들의 특성과 시장 환경

을 반영해야 하기 때문에 그대로 사용하기 보다는 커스터마이징 (Customizing)이 필요하며, 아직 정보화에 대한 인식이 산지에서 폭넓게 자리 잡지 못하고 있기 때문에 이에 대한 홍보 및 교육 등이 절실하게 요구되기도 한다.

산지유통전문조직의 경쟁력을 높이기 위해서는 지금까지의 시설위주 하드웨어적인 정부지원에서 벗어나 본 연구에서 개발한 모바일 유통관리시스템 등 정보시스템 보급에 대한 지원을 적극적으로 고려해야 할 것이다.

SUMMARY

The objective of this study is to develop a marketing management system of producer marketing associations using mobile computing technologies. For this purpose, this study conducted surveys on information systems of producer marketing associations. The development of mobile system was made based upon the general development process of information systems.

Producer marketing associations lack of information mind because the scale of producer marketing associations is generally small as they can not be aware of information needs on account of deficiency of remaining strength of investment. In addition, they are insufficient to apply skills of database management like shipping information, because they are not used to apply scientific management skill.

In order to improve the on-line distribution system, producer marketing associations should reconsider efficiency with building up infrastructures of e-commerce and applying the information technologies like Supply Chain Management (SCM) system. Besides, they should promote information mind to introduce information systems such as ERP Systems, Mobile Systems, etc.

Therefore, producer marketing associations need to perform additional functions, which are to provide distributing information through developing and applying mobile system. For workers of producer marketing associations, mobile system makes them available to record, modify, delete, and search informations that they want to access at any places and time as well as to manage and analyse informations. Moreover, mobile systems get

the surroundings of pandemic distribution businesses improved via promoting productivity and increasing efficiency. Consequently, mobile system is such an outstanding measure to improve competitiveness of producer marketing associations being able to process real time data and uplift confrontation.

The mobile system developed in this study consists of sub-processes such as, storage management, sorting management, packing management, and shipping management.

By this study, producer marketing associations will be convenient and possible to manage based upon data. It is also expected the mobile system developed in this study to be applied to food processing and distribution companies. In order to apply the current system to other types of organization, it has to be customized to reflect characteristics of each associations and marketing circumstance. Moreover, the members of producer marketing associations are definitely required to get educated and promoted in order to stimulate information mind.

Table of Contents

I . Introduction

1. Necessity for Research and Development
 - 1) Necessity for Research and Development
 - 2) Current Situation of Domestic and Overseas Technology Trends
2. Targets and Contents of Research and Development
3. Research Methodologies

II. Present Condition and Information Strategy of Producer Marketing Associations

1. Current Situation and Performance of Producer Marketing Associations
 - 1) Current Situation of Producer Marketing Associations
 - 2) Performance of Producer Marketing Associations
2. Current Situation of Agricultural Marketing Information
 - 1) Significance of Marketing Information
 - 2) Current Situation of Marketing Information Systems in Producing Areas
 - 3) Present Condition of Marketing Information System at the Wholesale and Retail Level
 - 4) Agricultural Marketing Information System
3. Current Situation of Information System of Producer Marketing Associations
 - 1) Outline of Investigation
 - 2) Current Situation of Information System of Producer Marketing Associations
 - 3) Evaluation of Information System of Producer Marketing Associations
 - 4) Information System Strategies of Producer Marketing Associations
4. Case Study and Current Situation of Agricultural Product Information Management Systems in the U.S.A.
 - 1) Current Situation of Agricultural Marketing Softwares

- 2) Spokane Software System (SSS) of Packing Houses
- 3) Application Softwares to Agribusiness Industries
- 4) Softwares for Physical Distribution Warehouse Management: ATS
- 5) Packing House Management Program in Wyoming state
- 5. Concept of a Mobile System and the Promotion Strategy
 - 1) Concept of Wireless Internet and Ubiquitous
 - 2) Concept of Mobile Computing
 - 3) Effects and Problems of Mobile Computing
 - 4) Application Examples of Mobile systems
 - 5) Perspectives of Mobile System Application to Agricultural Marketing
 - 6) Mobile Promotion Strategies of Producer Marketing Associations

III. Research and Development, and Evaluation Method

- 1. Methods of Research and Development
- 2. Outputs by Development Stage
- 3. Methods according to Detailed Development Stage
 - 1) System Plan
 - 2) System Analysis
 - 3) System Design
 - 4) Realization
 - 5) Maintenance Repair and Evaluation
- 4. Evaluation Method of Information System
 - 1) Theoretical Background of The Information System Evaluation
 - 2) The Derivation Method of Investment Effects
 - 3) Main Issues and Procedure of Investment Effects Analysis on Computerization

IV. System Design and Development

- 1. Definitions of the System Development Environments
 - 1) Outline for Development
 - 2) Technical Trends of the System for Development
 - 3) Something to Consider for the System Development
- 2. Outline of the System
 - 1) System Configuration

- 2) Technical Characteristics of Embodiment System
- 3) Installation Items of S/W
- 4) Concept of System Employment
3. An Analysis on Business and System
 - 1) Analysis of System Configuration
 - 2) Reengineering of Present Business Process
 - 3) Definition of Need Analysis
 - 4) Data flow Diagram (DFD)
4. System design
 - 1) System DB Tables
 - 2) Physical DB Definitions
 - 3) Production Distribution Management System
 - 4) Physical Entity Relationship Diagrams (Physical ERD)
 - 5) Menu Composition
 - 6) Code Composition
5. Screen Development of Input–Output
 - 1) Definition of Input–Output Screen
 - 2) Screen Development of Marketing Management System
6. Composition of System File
 - 1) System Execution File
 - 2) Mobile File (Module)
 - 3) Module Sauce

V. Expected Effects and Evaluation

1. Expected Effects
2. Method of Practical Use
3. Effect Analysis
4. User Evaluation

VI. Conclusion

목 차

제 1 장 서 론

제 1 절 연구개발의 필요성	1
1. 연구개발의 필요성	1
2. 국내·외 관련기술의 현황과 문제점	6
3. 앞으로 전망	8
제 2 절 연구개발의 목표 및 내용	10
1. 연구개발의 목표	10
2. 연구개발의 내용	10
제 3 절 연구 방법	11

제 2 장 산지유통전문조직 현황 및 정보화전략

제 1 절 산지유통전문조직 현황 및 성과	14
1. 산지유통전문조직 현황 및 실태	14
2. 산지유통전문조직의 성과	17
제 2 절 농산물 유통정보화 현황	21
1. 유통정보의 중요성	21

2. 산지유통 정보화 현황	22
3. 소비자유�� 정보화 현황	28
4. 농산물유통정보시스템	30
제 3 절 산지유통전문조직의 정보화 실태 및 추진 전략	33
1. 조사개요	33
2. 산지유통전문조직의 정보화 실태	33
3. 산지유통전문조직 정보화에 대한 평가	47
4. 산지유통전문조직 정보화 추진 전략	48
제 4 절 미국의 농산물 산지 정보관리시스템 운영 실태와 사례	50
1. 산지농산물 유통 소프트웨어 실태	50
2. 선과장의 정보관리시스템 소프트웨어 : Spokane SoftWare System(SSS)	50
3. 농업관련산업 적용 소프트웨어: Kirkey 소프트웨어시스템	55
4. 물류창고 관리 소프트웨어 : ATS(applied Technical Systems)	60
5. 와이오밍주 선과장 운영 프로그램(VEGMARC)	61
제 5 절 모바일시스템의 개념 및 추진전략	64
1. 무선인터넷 및 유비쿼터스의 개념	64
2. 모바일 컴퓨팅의 개념 및 등장배경	70
3. 모바일 컴퓨팅의 효과 및 문제점	74
4. 모바일시스템의 적용사례	77

5. 농산물 유통에의 모바일시스템 응용 전망	83
6. 산지유통전문조직의 모바일 컴퓨팅 추진전략	85
제 2장의 부록1 사과선과방식	82
제 2장의 부록2 산지유통전문조직 정보화 실태 조사표	87
제 3 장 연구개발 및 평가 방법	
제 1 절 연구 개발방법	95
제 2 절 개발 단계별 산출물	97
제 3 절 세부개발 단계별 방법	98
1. 시스템 계획	98
2. 시스템 분석	99
3. 시스템 설계	101
4. 구현	102
5. 유지보수 및 평가	103
제 4 절 정보시스템 평가방법	104
1. 정보시스템 평가의 이론적 배경	104
2. 정보화 투자효과 유형 및 효과지표 도출 방법	105
3. 정보화 투자효과 분석절차 주요 이슈	107

제 4 장 시스템 설계 및 개발

제 1 절 시스템 구축 환경정의	110
1. 구축대상의 개요	110
2. 구축대상 시스템의 기술적 동향	113
3. 시스템 개발시 고려사항	119
제 2 절 시스템 개요	124
1. 시스템 구성도	124
2. 구현 시스템의 기술적 특성	125
3. S/W의 설치내역	127
4. 시스템 운용 개념도	128
제 3 절 업무 및 시스템 분석	130
1. 시스템 구축을 위한 업무분석	130
2. 현행 업무 흐름의 문제점 개선 방향	135
3. 요구분석정의서	136
4. 자료흐름도(DFD)	139
제 4 절 시스템 설계	140
1. 논리적 엔터티 관계도(Logical ERD)	140
2. 물리 DB 설명	142

3. 물리 DB 정의서	142
4. 물리적 엔티티 관계도(Physical ERD)	148
5. 메뉴구성도	149
6. 코드구성	150
제 5 절 입출력 화면 개발	160
1. 입출력 화면정의서	160
2. 유통관리시스템 화면 개발	173
제 6 절 시스템 파일의 구성	181
1. 시스템 실행 화일	181
2. 모바일 파일(Module)	181
3. Module 소스	183
제 4 장의 부록 ARPC 모바일 유통관리 시스템	312
제 5 장 기대효과 및 평가	
제 1 절 기대효과	326
제 2 절 활용방안	327
제 3 절 효과분석	328
제 4 절 사용자만족도 조사 결과	331
제 6 장 결론	

표 목 차

<표 1-1> 산지유통전문조직 선정 현황	2
<표 1-2> 모바일 컴퓨팅의 특징	3
<표 2-1> 산지유통전문조직 현황	5
<표 2-2> 산지유통전문조직의 지역별 분포 현황	6
<표 2-3> 산지유통전문조직의 직거래 실적	6
<표 2-4> 산지유통전문조직의 물류기기 공동이용 현황	7
<표 2-5> 산지유통전문조직의 유형	7
<표 2-6> 산지유통전문조직의 마케팅 구조 변화	8
<표 2-7> 산지유통전문조직의 조합원, 이사회 조직 및 의식구조 변화	9
<표 2-8> 유통정보의 평가기준	22
<표 2-9> 전자경매시스템의 정보내역	28
<표 2-10> 기관별 유통정보 현황	23
<표 2-11> 지역별 조사업체 분포	33
<표 2-12> 조직 형태별 현황	43
<표 2-13> 산지유통전문조직 선정 연도별 현황	43
<표 2-14> 컴퓨터 및 프린터 보유 현황	53
<표 2-15> 네트워크 통신망 현황	53
<표 2-16> 소프트웨어 사용 현황	63
<표 2-17> 자체 정보시스템 보유 현황	73
<표 2-18> 경제종합정보시스템의 사용 효과	83
<표 2-19> 경제종합정보시스템의 문제점	04
<표 2-20> 종합관리시스템(ERP)의 도입 의향	14
<표 2-21> 종합관리시스템(ERP)의 도입 시 예상효과	14
<표 2-22> 전사적자원관리(ERP) 도입의향이 없는 이유	24

<표 2-23> 모바일시스템의 도입 의향	31
<표 2-24> 모바일시스템의 도입 의향이 있는 이유	44
<표 2-25> 모바일시스템 도입 시 적용 분야	44
<표 2-26> 모바일시스템의 도입 시 예상효과	54
<표 2-27> 모바일시스템 도입의향을 가지고 있는 않은 이유	64
<표 2-28> 컴퓨팅 개념의 진화 단계	76
<표 2-29> M-commerce가 확산될 수 있는 동기 요인	27
<표 2-30> 모바일 컴퓨팅 환경 도래	47
<표 2-31> 모바일시스템 도입과정의 장애요인과 해결방안	67
<표 2-32> 롯데제과 모바일 시스템의 효과	97
<표 3-1> 시스템 개발의 단계별 추진내역 및 산출물	79
<표 3-2> 투자효과의 유형	10
<표 4-1> 산지유통전문조직 선정현황	11
<표 4-2> 순천농협 농산물 산지유통센터 일반현황	12
<표 4-3> 순천농협 농산물 산지유통센터 사업실적	12
<표 4-4> 순천농협 농산물 산지유통센터 공동유통시설 보유현황	13
<표 4-5> 모바일 기술적 구조계층	17
<표 4-6> 코드구성내역	10
<표 4-7> 입출력화면정의서 설명	11
<표 4-8> 모바일 산지유통 프로그램 목록	11
<표 5-1> 모바일 컴퓨팅 도입의 효과	27
<표 5-2> 업무감소율 상세내역 분석	28
<표 5-3> 업무감소율 사업적 가치분석	29
<표 5-4> 효과적인 경영 의사결정 시간 단축 상세내역	29
<표 5-5> 효과적인 경영 의사결정 시간 단축 사업적 가치분석	30
<표 5-6> 정량적 효과지표에 의한 투자효과(천원)	3
<표 5-7> 사용자 만족도 조사항목	32

그림 목 차

<그림 1-1> 새로운 농산물 유통정보시스템 구축 요인	5
<그림 1-2> 모바일 컴퓨팅의 구성요소	7
<그림 1-3> PDA를 이용한 모바일컴퓨팅 환경의 구축	7
<그림 1-4> 최근 기업정보화의 추이	8
<그림 1-5> 휴대단말기의 향후 전망	9
<그림 2-1> 경제종합정보시스템 구성 현황	2
<그림 2-2> 한국참다래유통사업단 ERP시스템 구성	2
<그림 2-3> 농산물유통정보시스템의 흐름도	3
<그림 2-4> 인터넷 및 이동전화가입자 현황	5
<그림 2-5> 유비쿼터스 컴퓨팅의 구현 방향	5
<그림 2-6> 모바일 컴퓨팅의 도입 예정기업 분포 및 예정 솔루션	7
<그림 2-7> 모바일시스템의 효과	7
<그림 2-8> 모바일 영업지원시스템 구축사례	8
<그림 2-9> 택배사 모바일 정보 시스템 사용 사례	8
<그림 2-10> 롯데제과 모바일 시스템 구축 사례	8
<그림 2-11> 롯데제과 PDA 활용사례	8
<그림 2-12> 모바일 물류관리 시스템의 구축사례	18
<그림 3-1> 관리기법/1 적용 절차	9
<그림 3-2> 정보화 효과지표 도출방법	10
<그림 3-3> 정보화 투자효과 분석절차	10
<그림 4-1> 모바일 하드웨어(H/W) 플랫폼	11
<그림 4-2> 시스템 구성도	15
<그림 4-3> 소프트웨어 구성도	15
<그림 4-4> 시스템 운용개념도	19
<그림 4-5> 전체 업무 흐름도	19

<그림 4-6> 수매/입고 업무 흐름도	11
<그림 4-7> 선별/소포장 업무 흐름도	12
<그림 4-8> 출하업무 흐름도	13
<그림 4-9> 현행 업무 개선방향	16
<그림 4-10> 데이터흐름도 모바일부분 자료흐름	19
<그림 4-11> 논리적 엔터티 관계도	11
<그림 4-12> 물리적 엔터티 관계도	18
<그림 4-13> 메뉴 구성도 유통관리시스템	19
<그림 4-14> 모바일 로그인 화면	14
<그림 4-15> 수매입고 조회화면	15
<그림 4-16> 모바일 산지유통 출하관리 화면	17
<그림 4-17> 모바일 산지유통 포장관리 화면	17
<그림 4-18> 모바일 산지유통 선별마스터관리 화면	18
<그림 4-19> 선별마스터, 선별상세 등록/수정 화면	19
<그림 4-20> 자료 입력용 키패드 구축화면	19
참고문헌	338

제 1 장 서 론

제 1 절 연구개발의 필요성

1. 연구개발의 필요성

정부는 '98년까지 농산물 규격출하 촉진자금지원, 생산자 유통지원사업 및 품목별 생산자조직육성지원을 실시하였으며, 1999년에는 각종 산지유통 관련 유사사업을 통합하여 주산지 중심, 전략품목에 대해 집중 지원하였고 2000년에는 산지유통 개혁방안에 따라 각종 운영자금을 통합 지원하여 산지유통 활성화사업을 전개해 오고 있다.

그동안 산지유통은 개별 농업인을 비롯하여 20,000여개의 작목반, 5,000여개의 영농법인, 1,200여개의 협동조합 등 영세규모의 조직에 의해 주도되어 왔으나, 이제 이와 같은 영세규모 조직으로는 상품의 균일성, 공급의 지속성, 브랜드 상품 선호 등 시장의 요구를 도저히 수용하기 어려운 실정이다.

특히 소비지에 대형유통업체가 속속 등장하고, 전자상거래가 활성화되는 등 농산물 판매여건이 크게 달라짐에 따라 산지의 농산물 마케팅조직의 규모화를 집중 지원하기로 하였다. 2003년도까지 288개 협동조합 및 영농조합법인이 산지유통전문조직으로 선정되었으며, 총 10,516억원의 자금을 집중 지원하여 산지유통전문조직으로 육성하고 있다.

<표 1-1> 산지유통전문조직 선정현황

(단위 : 억원, 개소)

구분	2000	2001	2002	합계
조직수	99	62	127	288
총사업비	2,641	2,875	5,000	10,516

선정된 전문조직은 공동출하, 품목별 전문화, 해당 품목 전국협의회 가입 및 지역협의회 결성, 공동계산, 파렛트 출하 등이 높은 조직으로 ① 공동선별, 공동수송, 공동판매를 통한 유통효율 증진 ② 조합당 유통사업 취급품목을 3개 품목 이내로 제한, 주산지화·전문화 촉진 ③ 지역협의회 운영활성화를 통해 자율적인 수급조절 능력 및 시장교섭력을 강화하는데 목적이 있다.

따라서 주산지화·전문화와 공동선별, 공동수송, 공동판매 및 자율적인 수급조절과 시장교섭력을 증진시키기 위한 하나의 대안으로써 정보통신기술(ICT)인 모바일 컴퓨팅을 도입하여 산지유통전문조직의 유통 효율성과 수급조절 및 시장교섭력을 제고할 필요가 있다.

가. 기술적 측면

모바일 컴퓨팅 환경의 농업정보화는 PDA 등 휴대형 단말기를 이용한 농업경영체의 전산시스템을 말하는 것으로 원격지 데이터 및 상품관리, 판매처관리, 재고관리, 운영인력 등의 효과적인 관리체계를 구축하기 위한 농업경영체 정보화 기술로 정의될 수 있다.

아울러 모바일 컴퓨팅기술은 산지유통전문조직의 현실적인 문제인 불특정 원격지에서 빈번하게 발생하는 데이터관리, 현장 수배송관리, 판매관리

등 효과적인 관리체계 부족 및 현장 정보를 분석정보로써 효율적으로 활용하지 못하는 문제점 등을 원활하고 실시간에 해결하기 위한 정보통신기술(ICT)이다.

<표 1-2> 모바일 컴퓨팅의 특징

구 분		주 요 내 용
Mobility	Ubiquity	지정된 장소 어디에서나 실시간으로 정보 등록, 수정, 삭제, 검색 가능
	Reachability	언제, 어디서나 데이터베이스에 접속 가능
	Convenience	간단한 통신 도구 이용, 휴대에 용이
Localization		특정 시점에 사용자의 현 위치를 파악, 활용 가능
Security		보안상의 장점 보유
Personalization		휴대 단말기를 이용, 개별적으로 사용 가능
Instant Connectivity		빠른 시간내에 필요한 정보를 탐색 및 등록 가능

산지유통전문조직은 전문 품목을 중심으로 규격화, 체계적인 물류가 이뤄지고 있어 정보통신기술(ICT)의 적용이 매우 용이하고, 사무실 같은 지정된 장소가 아니라 여러 장소를 이동하며 현장에서 다양한 정보를 관리해야 하는 농업특성상 모바일 컴퓨팅기술의 도입은 매우 획기적인 효율성 제고의 대안이 될 수 있다.

나. 경제·산업적 측면

산지유통은 그간 조직 확충, 시설과 장비 및 자금지원 확대 등으로 조직화 기반은 갖추어져 있으나 운영 면에서 효율성과 경쟁력 제고에는 미흡하여 농산물 산지유통의 핵심체로 산지유통전문조직 육성이 농산물 유통정책의 중점과제로 선정되었다.

산지유통의 핵심체로 육성하기 위한 산지유통전문조직은 바코드 도입과

공동규격출하, 공동계산, 시설자동화 등 정보화 추진여건이 성숙되어 있으며, 소비자 유통환경에 적절하게 대응하기 위해서는 모바일 컴퓨팅 환경의 도입이 매우 중요하다고 볼 수 있다.

모바일을 이용한 산지유통전문조직의 유통관리시스템은 종사자들에게 있어서 언제 어디서나(지정된 장소) 데이터베이스에 접근하여 원하는 정보를 등록, 수정, 삭제, 조회가 가능하며 체계적인 관리 및 분석이 가능하게 하고, 전반적인 유통업무 환경의 질적 개선을 통해 생산성 향상과 효율성 증대를 꾀할 수 있으며, 데이터의 실시간 처리 및 고객 대응력 제고 등 산지유통 전문조직의 경쟁력과 통제력을 향상시킬 수 있다.

최근 FTA등 개방화 시책이 가시화되고, 구조적 과잉생산 및 소비행태의 변화, 유통환경변화 등 해외농산물과의 경쟁은 물론 국내 농산물간 경쟁도 치열해질 전망으로 물류체계의 효율화 및 위험관리 등 산지유통전문조직의 운영 면에서 효율성과 경쟁력을 제고하는 방안이 필요하다.

결과적으로 농산물유통 특성을 반영하여 가장 적절한 정보통신기술을 적용할 수 있는 모바일 컴퓨팅을 도입한다면 현장에서 업무처리 및 신속 정확한 데이터 처리로 생산성 향상 및 작업효율 증대, 물류/주문/재고관리의 효율화, 품질관리 기능강화가 가능하게 되어 비용 절감 및 업무능률 향상, SPEED한 의사결정 지원, 신속한 고객 대응 등이 가능하게 될 것이다.

다. 사회·문화적 측면

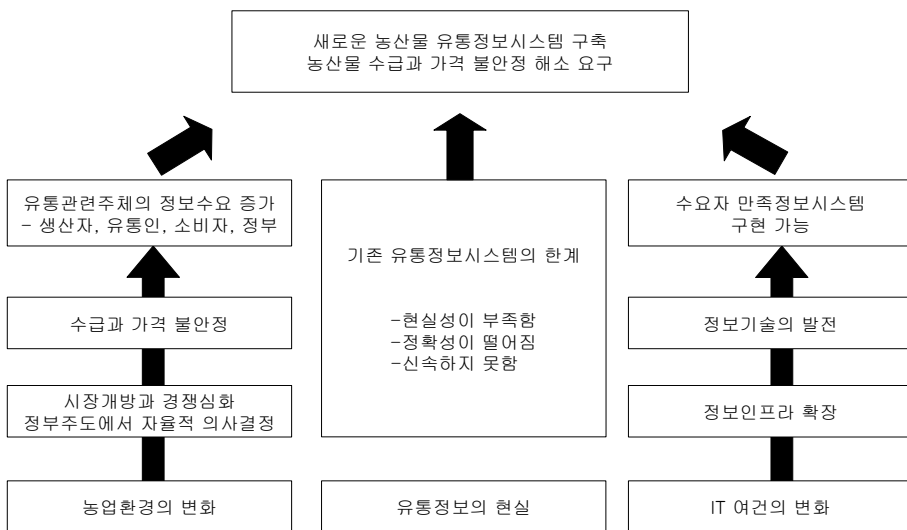
사무실 같은 지정된 장소가 아니라 여러 장소를 이동하며 근무해야 하는 농업특성상 현장에서의 다양한 정보를 언제나 편리하게 꺼내보고 저장할 수 있는 매우 현실적인 요구가 증대되고 있다.

산지유통전문조직처럼 정보화가 필수적인 조직에서의 모바일 컴퓨팅 환경의 도입은 개인용 컴퓨터를 통한 종이기반의 작업을 개선하고, 현장작업 구조 자체를 개선하여 생산성을 높일 수 있을 것으로 예상된다.

또한 소비지 유통환경변화에 대응하여 농산물 품질규격화가 촉진되고 품질위주 출하체계가 구축되며, 공동선별 활성화 및 출하물량의 규모화가 가능하게 되어 수급안정을 통한 농가소득 지지, 소비자 만족의 유통체제 정착, 실질적인 농업정보화를 통한 정보격차(Digital Divide) 해소 등의 역할을 담당할 것으로 예상된다.

특히 공급과잉의 시대로의 수급구조 변화, 외국농산물 수입 증가, 소비행태 변화, 외식산업 성장, 대형유통업체의 급속한 증가, 농산물 전자상거래의 급성장 등 유통부문의 변화가 전방위적으로 일어나고 있는 상황에서 모바일 컴퓨팅을 이용한 유통관리시스템은 산지유통전문조직의 본래의 목적달성과 생산성 제고, 유통비용을 절감시킬 수 있는 적절한 방안이 될 수 있다.

<그림 1-1> 새로운 농산물 유통정보시스템 구축 요인



2. 국내 · 외 관련기술의 현황과 문제점

모바일 컴퓨팅의 활용은 영업과 물류, 운송, 택배 부문에서 가장 활발하게 이루어지고 있으며, 현재 동양제과, 롯데제과, 농심, 코카콜라 등 식음료 업체들이 영업현장에서 PDA를 이용하여 생산 및 재고, 유통에서 효율성을 증진하고 있다.¹⁾

모바일 영업지원시스템은 식음료 업체와 보험업계를 중심으로 활용이 증가하고 있으며, 동양제과는 영업소, 공장, 물류창고를 하나의 시스템으로 연결한 첨단 물류창고를 구축하여 물류의 흐름에서 발생하는 입고, 재고, 출고, 판매, 택배 등 생성되는 데이터를 실시간으로 수집하고 분석할 수 있게 됨으로써 효율적인 관리가 가능하게 되었다.

농심도 PDA를 이용해 재고나 출고상황을 조회하고 영업정보를 검색할 수 있을 뿐 아니라 제품 주문을 즉시 처리할 수 있는 모바일 영업지원시스템을 구축하여 실제 판매현황 및 유통재고 파악에 의한 영업, 생산, 물류 전략수립의 기초정보 지원 등에 활용하여 연간 30억원 가량의 영업활동비용 절감효과를 기대하고 있다.

롯데칠성음료는 생산 집계에서부터 제품 출하에 이르기까지 모바일 컴퓨팅을 구축하여 기존업무의 효율성을 제고시키고 있으며, 모든 보험업계도 영업지원시스템을 구축하여 경쟁력을 높이고 있다.

외국의 경우 물류회사인 Fedex가 1986년부터 수하물의 위치, 선적상황, 기타 배송관련 정보를 고객과 회사에 실시간으로 전송할 수 있도록 '슈퍼 트랙커'라는 초소형 컴퓨터를 이용하고 있으며, DHL과 같은 택배업체와

1) 배찬권, 「PDA를 이용한 모바일 오피스 활용 사례」, KISDI IT FOCUS, 2001. 11

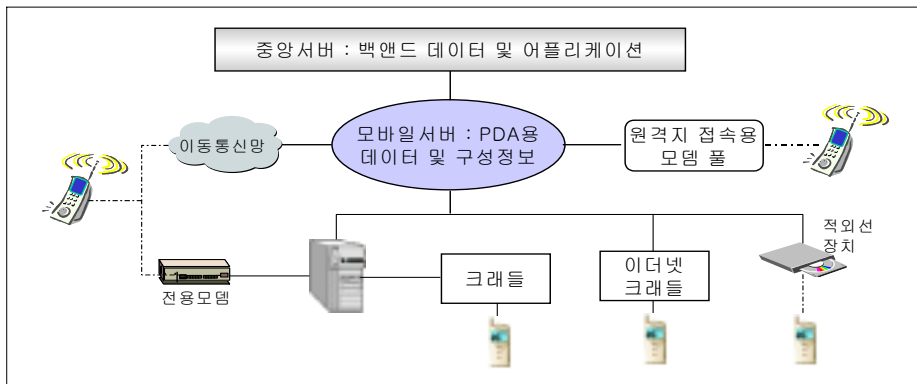
맥케슨, 오피스데포 등 대형 물류업체들은 일찍이 바코드 스캐너를 장착한 무선 PDA를 업무에 활용해 오고 있다.

<그림 1-2> 모바일 컴퓨팅의 구성요소



맥케슨에 의하면 모바일 컴퓨팅의 도입으로 생산성 8% 향상, 제품 선적 오류 80% 감소, 재고부족 50% 감소, 재고목록의 정확성 99.5% 도달이 나타났으며, 이는 미국 창고업체가 잘못된 제품선적과 회수에 드는 비용으로 평균 1회당 100달러 가량을 소요하고 있는 것으로 볼 때 대단한 성과라고 할 수 있다.

<그림 1-3> PDA를 이용한 모바일컴퓨팅 환경의 구축

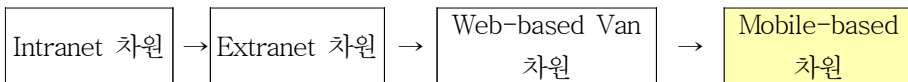


농업관련분야에는 농협유통이 하나로클럽의 매장관리 등에 일부 활용하고 있고, 농산물 수배송 차량위치추적 등 농산물 유통부문에 일부 활용되고 있다. 그러나 지금까지의 모바일 컴퓨팅을 활용한 정보기술은 대기업형 유통 및 물류기업을 위해 개발되어졌으며, 산업적 측면에서 특히 농업관련 기술로 개발된 사례가 부족한 실정이다. 또한 산지유통에 대한 독자적인 분석이 부족하여 산지유통전문조직은 브랜드관리 시스템 자체가 지향해야 될 방향 및 목표를 정하고, 생산 → 입고 → 예냉 → 저장 → 선별/포장 → 출하 등 표준적인 프로세스를 정립하여 적절한 시스템을 구축함으로써 기존 유통 및 물류관련 기술과는 다른 독창성과 혁신성이 반영되는데 어려움이 있다.

3. 앞으로의 전망

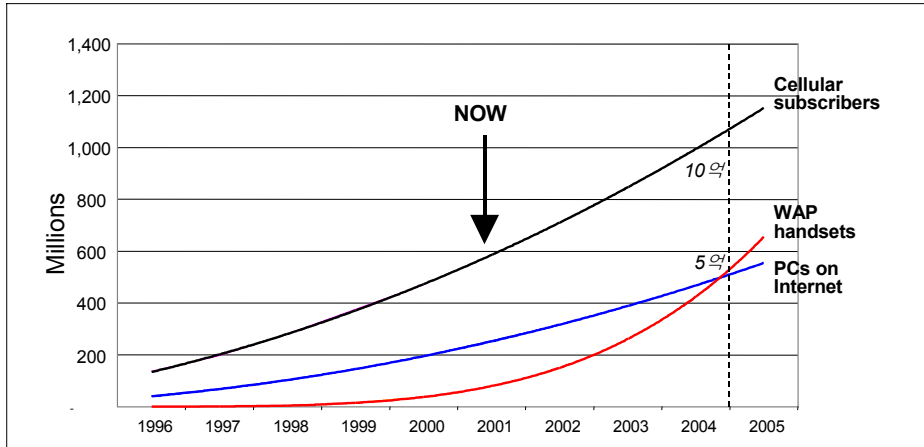
현재 일반 산업분야에서는 업무생산성 향상과 비용절감을 효과적으로 이루어내기 위해서는 모바일 인프라로의 전환을 서두르고 있으며, 영업직원 또는 생산직원 등 현장 근로자가 현장에서 직접 회사의 데이터베이스에 접근할 필요성이 증가하면서 모바일 컴퓨팅 사무환경에 대한 수요가 증가하고 있다.

<그림 1-4> 최근 기업정보화의 추이



Forrester Research(2001. 3)는 미국의 경우 2001년 초 기준 전체 업체의 약 30%정도만이 SFA(Sales Force Automation)과 같은 보다 향상된 기업 어플리케이션을 이용하고 있으나, 향후에는 대부분의 기업들이 보다 고급의 기업 어플리케이션을 이용할 것으로 예상하였다.

<그림 1-5> 휴대단말기의 향후 전망



자료 : Andersen Consulting, 2001.

모바일 컴퓨팅 환경은 영업, 재고관리 등 기존 업무의 생산성 최적화는 물론 CRM 등 다양한 부문의 핵심도구로 활용될 것이며, 특히 농업의 특수성에 비추어 볼 때 그 활용은 매우 다양하다.

농업분야에서 우선 적용이 가장 용이한 분야는 농산물 유통부문으로 본 연구에서는 향후 농산물 유통을 주도적으로 선도할 산지유통전문조직의 유통관리시스템 구축에 관한 연구로 그 시장은 이미 선정되어 운영하고 있는 288개(2003년 기준)의 전문조직에 시스템을 구축하여 활용이 가능하며, 유사업무를 수행하고 있는 산지유통센터 208개소(2003년 기준)에도 수행하고 있는 기능에 따라 적용이 가능하다.

사업이 완료되어 활용이 활성화되면 단순관리에서 벗어나 생산 및 출하 예측에 의한 자율 수급조절이 가능토록 시스템을 확장하여 산지에서 농산물 출하의 중심적 역할을 담당할 수 있도록 유도하고, 고객관리(CRM)까지 가능한 가치사슬(Value Chain)상의 모든 업무프로세스를 관리하기 위한 확장ERP(Extended Enterprise Resource Planning)로 발전시킬 계획이다.

아울러 다른 산업분야에 비해 산지유통전문조직의 정보화가 뒤떨어져 있는 점을 감안하여, 산지유통 조직의 내부정보화(ERP)와 연계하여 합리적인 경영 및 의사결정이 가능한 시스템으로 확장하고, 향후 전자상거래(B to B, B to C) 등과 연동, 산지유통전문조직의 핵심전산화를 이끌어 갈 계획이다.

이 외에도 농산물 유통업체 및 계열화단체, 식음료회사 등에 응용하여 활용이 가능하며, 농가에는 농장생산 및 경영관리 데이터를 수집·분석하는 데 활용이 가능하다.

제 2 절 연구개발의 목표 및 내용

1. 연구 개발의 목표

본 과제의 목적은 산지유통의 핵심체로 육성하고자 하는 산지유통전문조직의 수행업무에 모바일 컴퓨팅 기술을 접목하여 생산성 향상 및 효율성 제고를 위한 현실적인 대안을 제시할 수 있는 유통관리시스템을 개발하는데 있다.

2. 연구 개발의 내용

가. 산지유통전문조직의 모바일 컴퓨팅 환경도입을 위한 기초연구와 시스템 평가 및 활용방안 연구

○ 기존 산지유통전문조직의 정보화 실태 조사 및 분석을 통한 향후 시스템 도입 방안 연구

- 산지유통전문조직의 정보화 현황 및 실태 분석
- 해외의 산지유통 정보화 및 향후 정보화 방향 분석
- 국내·외 모바일 컴퓨팅 기술의 적용 사례 조사

○ 산지유통전문조직 구축 시스템의 효과 및 활용방안 연구

- 시스템 구축효과 조사 및 분석
- 시스템 구축 후 활용방안 연구

나. 모바일 컴퓨팅을 이용한 산지유통전문조직의 유통관리시스템 개발

○ 모바일 컴퓨팅기술을 이용한 농산물 생산·입고에서부터 출하·판매에 이르기까지 입고관리, 저장관리, 선별/포장관리, 재고관리, 출고관리 등의 원격지 데이터를 수집이 가능한 유통관리시스템 구축

○ 입·출고관리, 저장관리, 선별/포장관리 등 모듈별 'Real time information' 생성 및 'Information availability'시스템 어플리케이션 개발

- 입고관리 : 등급·규격별 입고 등록/조회
- 저장관리 : 등급·규격별 저장고 입고 등록/조회
- 선별, 포장관리 : 선별 포장 등록/조회, 미선별 재고관리, 포장 재고관리
- 출고관리 : 출고내역 등록/조회, 상품 재고관리

제 3 절 연구 방법

○ 기초연구를 위해 문헌조사 및 업체 직접 조사에 의거 연구 자료를 수집하고 이를 토대로 분석하여 모바일 컴퓨팅 도입환경 조성

1. 조사내용

100개의 산지유통전문조직을 선정하여 정보화 실태를 조사하였으며, 주요 조사 내용은 다음과 같다.

- 정보인프라 현황 : 컴퓨터 및 프린터 현황, 네트워크통신망 현황, 소프트웨어 현황, 자체 정보시스템 현황
- 경제종합정보시스템 현황 : 사용분야, 효과 및 문제점
- 종합관리시스템(ERP) 도입여부 : 도입의 필요성, 도입의향 여부, 도입 시 기대효과
- 모바일시스템 도입여부 : 도입의 필요성, 도입의향 여부, 도입 분야, 도입 시 기대효과

2. 모바일 시스템 개발 프로세스

산지유통전문조직에 대한 모바일 시스템은 순천농협농산물산지유통센터를 대상으로 개발되었으며, 시스템 개발은 시스템 계획, 시스템 분석, 시스템 설계, 구현, 유지 및 보수 등 5단계 절차에 따라 진행되었다.

가. 시스템 계획

- 1) 기초연구
- 2) 시스템 개발 착수
- 3) 업계 및 환경 분석
- 4) 개발 톨 설정
- 5) 정보 인프라 분석

나. 시스템 분석

- 6) 사용자 요구 조사
- 7) 사용자 요구 분석
- 8) 기존 정보시스템 분석
- 9) 업무 흐름 분석
- 10) BPR
- 11) 개발목표 설정

다. 시스템 설계

- 12) 세부 업무 분석
- 13) 데이터 사전 작성
- 14) 데이터 베이스 스키마 작성
- 15) 메뉴 설계
- 16) 화면 설계
- 17) 보고서 설계

라. 구현

- 18) 모듈 설계
- 19) 프로그래밍
- 20) 사용자 매뉴얼 작성
- 21) 인스톨
- 22) 베타 테스트 실시

마. 유지·보수 및 평가

- 23) 디버깅
- 24) 개발 및 납품 완료
- 25) 사용자 만족도 조사
- 26) 유지 보수
- 27) 활용방안

제 2 장 산지유통전문조직 현황 및 정보화전략

제 1 절 산지유통전문조직 현황 및 성과

1. 산지유통전문조직 현황 및 실태

농수산물유통공사의 2003년도 산지유통전문조직의 운영현황 자료에 의하면 전국의 농업협동조합과 영농조합법인 중 산지유통전문조직으로 선정되어 유통활성화사업을 수행한 조직은 288개소이고, 그중에서 농협이 263개소, 영농조합법인이 25개소이다(표 2-1).

산지유통전문조직의 사업실적을 살펴보면, 산지유통전문조직으로 선정된 농협조직은 사업실적이 일반조합에 비해 조합 당 2배 수준으로 판매사업이 활성화되고 있고, 영농법인은 3배 수준으로 판매사업이 활성화되고 있다.

아울러 산지유통전문조직의 사업실적 성장률은 일반조직들에 비해 2.1% 높게 나타나고 있다. 이는 농산물공급과잉과 경쟁격화로 인한 농산물가격 하락 추세 속에서도 산지유통전문조직은 일반조합 및 일반영농법인에 비해 수익률 면에서 유리한 위치에 있다는 것을 보여준다.

<표 2-1> 산지유통전문조직 현황

(단위: 개소, 억원, %)

구 분	2003년 조직수	사업실적			평 균			평 균 성장률 (%)
		2001년	2002	2003	2001	2002	2003	
농 협 (일반조합)	261 (1,165)	30,048 (71,657)	32,331 (78,239)	35,041 (83,414)	112 (60)	121 (65)	134 (65)	10.7 (10.7)
영농법인 (APC전체)	24 (84)	1,226 (1,386)	1,570 (1,887)	2,112 (2,723)	65 (19)	83 (24)	88 (32)	6.0 (33.3)
합 계	285	31,274	33,901	37,153	110	119	130	9.2

자료: 농수산물유통공사 유통정보부, '대형유통업체와 산지출하조직의 농산물거래실태 조사 분석,' 2004.

주: 1) 영농법인은 매출액 기준임.

2) 사업실적에 공판장 운영 및 가공품 매출액 포함.

3) 양곡, 수산 등 사업 대상에서 제외되는 품목과 그 가공품은 실적에서 제외함.

현재 산지유통전문조직으로 선정된 조직은 전남, 경북, 경남지역에 전체의 50% 정도가 선정되어 있으며 특별시 및 광역시(서울, 부산, 대구, 인천, 광주, 울산)에도 5.9%개소가 선정되어 있다(표 2-2). 이는 산지유통시설(산지유통센터, 저온저장고, 선별장, 간이집하장 등)이 전남, 경북지역에 가장 많이 분포되어 있기 때문에 전남, 경북지역에 산지유통전문조직이 가장 많이 분포되어 있는 것으로 나타났다.

<표 2-2> 산지유통전문조직의 지역별 분포 현황

(단위 : 개소, %)

총계	특별시 및 광역시	경기	강원	충북	충남
288 (100)	17 (5.9)	22 (7.6)	23 (8.0)	14 (4.9)	30 (10.4)
	전북	전남	경북	경남	제주
	19 (6.6)	48 (16.7)	54 (18.8)	41 (14.2)	20 (6.9)

자료 : 농수산물유통공사.

산지유통전문조직의 직거래 실적을 살펴보면, 산지유통전문조직은 마케팅기능의 강화로 대형유통업체와의 직거래가 증가추세에 있다(표 2-3). 2003년 대형유통업체와의 직거래실적은 6,135억원 규모로 전체 판매실적의 16.5% 수준이며, 직거래 비중이 전년대비 12.2% 가량 증가하였다. 직거래 대상업체도 종전의 농협종합유통센터 위주에서 할인점, 백화점 등 일반 유통업체로 확대되고 있고 소포장, 수확후 관리 등을 통해 상품성이 향상되고 있다.

<표 2-3> 산지유통전문조직의 직거래 실적

(단위 : 억원, %)

구 분	2002년	2003년	성장률
판매실적	33,901	37,153	9.6
직거래실적	4,979	6,135	23.2
직거래비중	14.7	16.5	12.2

자료 : 농수산물유통공사

산지유통전문조직의 물류기기 공동이용 현황을 살펴보면, 2003년도 산지유통전문조직의 물류기기 공동이용사업에 대한 참여율은 66.0%로 전년대비 9.6%정도 증가한 것으로 나타났다(표 2-4).

<표 2-4> 산지유통전문조직의 물류기기 공동이용 현황

(단위 : 백만원, %)

구 분	조직수(A)	참여현황		지원현황	
		조합수(B)	참여율(B/A)	금 액	평균
'02	287	162	56.4	3,479	21.5
'03	계(285)	188	66.0	2,067	11.0
	261(농협)	175	67.0	1,910	10.9
	24(영농법인)	13	54.2	157	12.1

자료 : 농수산물유통공사

산지유통전문조직은 '선도농협형', '연합판매형', '마케팅법인형' 등 3가지로 구분하여 운영되고 있다(표 2-5). 그중에서 '선도농협형'이 254개소, '연합판매형'이 9개소, '마케팅법인형'이 25개소로 지정되어 운영되고 있다.

<표 2-5> 산지유통전문조직의 유형

유 형	특 징
선도농협형	선도농협을 마케팅 창구로 하여 작목반, 영농조합법인, 인근조합 등이 농산물을 공동으로 판매하되, 같은 품질관리 기준을 적용하는 방식
연합판매형	지역적으로 인접한 다수의 조합이 연합판매팀을 결성, 시장에 공동으로 판매하는 방식
마케팅법인형	영농법인 또는 농업인이 주주인 상법상 회사형태의 단일 마케팅법인을 설립하여 전문가에 의한 책임경영체로 운영하는 방식

2. 산지유통전문조직의 성과

가. 조합의 유통관리 접근시각의 변화

산지유통전문조직으로 선정되기 이전에는 대부분의 조합들이 소비자의 요구는 무시하고 오직 생산과 판매에만 관심을 기울이는 생산지향적인 혹은 판매지향적인 사고방식을 고수하였다.

그러나 산지유통전문조직으로 선정된 이후 조합들은 팔아주는 유통에서 벗어나 차별화된 농식품을 사전적으로 기획 생산하여 파는 소비자 지향적, 고객 지향적 접근 시각을 통하여 안정적인 판로를 확보하고 높은 가격을 받아 고소득을 확보하는 조합이 점차 증가하고 있고, 조합의 마케팅 구조가 바람직한 방향으로 전환되는 과정에 있다.

<표 2-6> 산지유통전문조직의 마케팅 구조 변화

구 분	산지유통전문조직 선정 이전	산지유통전문조직 선정 이후
유통접근 방식	· 생산지향적 유통사업	· 고객지향적 유통사업
유통 사업의 관리	· 단순 수집출하 · 단순한 선별, 포장, 저장, 출하 기능	· 전략적 마케팅 · 생산자를 계열화시키는 등 새 로운 기능 수행
유통 사업의 성격	· 수환보직에 의한 무수익사업 · 환원서비스적 판매사업	· 유통전문가에 의한 수익사업 · 농협과 조합원의 생존을 가능 하게 하는 버팀목
상품출하	· 상인과 개별적으로 직접 접촉 · 도매시장 중심	· 유통협약, 공동계산의 출현 · 직거래 및 대형유통업체와의 거래 증가 · 수확후관리에 대한 관심 제고

자료 : (사)농식품신유통연구원, 산지유통전문조직 유통활성화 사업의 성과와 개선방안, 2003.

나. 조합의 조직 및 의식구조 변화

산지유통전문조직 선정 후 조합원, 이사회 사이에서 조직의 변화, 의식구조의 변화가 나타나고 있고, 변화는 일반적으로 부정적에서 긍정적으로 전화되는 과정에 있다(표 2-7).

<표 2-7> 산지유통전문조직의 조합원, 이사회 조직 및 의식구조 변화

구 분	산지유통전문조직 선정 이전	산지유통전문조직 선정 이후
조합원	<ul style="list-style-type: none"> · 조합원들의 조합사업에 대한 불신(사업능력, 사업의 지) · 조합과 작목반간의 연계성 부족 · 조합원들은 조합에 대하여 일방적으로 요구 	<ul style="list-style-type: none"> · 조합사업에 대하여 조합원들이 긍정적으로 생각 · 조합과 조합원들이 합심하여 공동 노력하면 사업을 성공시킬 수 있다는 가능성 확인 · 조합과 작목반간, 조합과 조합원간의 연계성 강화 · 공동선별과 공동계산의 중요성 인식, 참여 확대 · 조합과 조합이 필요에 따라 서로 요구하는 쌍무적 관계
이사회	<ul style="list-style-type: none"> · 이사회에의 유통사업에 대한 이해부족 · 사업을 방해하지 않으나, 협조적인 분위기도 아님 	<ul style="list-style-type: none"> · 유통사업을 이해하고 협조적인 분위기로 반전

자료 : (사)농식품신유통연구원, 산지유통전문조직 유통활성화 사업의 성과와 개선방안, 2003.

다. 조합의 마케팅 능력 변화

산지유통전문조직 선정 이후 조합과 조합원들이 마케팅에 인식 및 마인

드는 급격하게 긍정적으로 변화하였고, 이에 따라 일부 선진 조합에서는 수확후 품질관리, 포장개선, 신상품개발, 브랜드관리 등을 통하여 상품화 능력을 제고시키고 있다. 이 점에서 많은 조합들이 산지유통전문조직 선정 이전 보다 발전하였다고 스스로 평가하고 있다.

아울러 이전에는 유통관련 업무가 3D업종으로 분류되었으나 일부 선진 조합의 직원들 사이에서는 해볼만한 업무로 부각되고 있고, 품질관리사 자격증을 취득하려는 자발적 움직임이 나타나고 있으며, 할인점이나 백화점 등 대형업소와 직접 거래를 하는 조합이 증가하고 있고, 일부 조합에서는 판로개척 능력의 부족을 실감하고 이에 대한 지도와 알선을 요청하는 등 시장을 개척하려는 노력이 나타나고 있다.

또한 농산물을 구입하는 바이어들이 유통활성화사업 이전에는 중도매인 으로부터 구입하는 경우가 많았으나, 사업 이후에는 산지유통센터 혹은 공동선별을 잘 하는 조합에서 구입하는 물량이 점차 증가하고 있다.

한편 상품을 알리려는 홍보관측에 대한 의식은 있으나 아직 노력 부족, 예산의 미비, 홍보방법의 미숙 등으로 많은 개선이 요구되고 있는 실정이고, 유통관리능력(판로개발 및 관리, 유통정보관리, 물류관리)과 가격관리능력(가격교섭력 확대, 독자적인 가격결정 등)도 의식은 있으나 방법과 능력의 측면에서 매우 미숙한 실정이다.

라. 기업적 성격의 도입

산지유통전문조직에 선정된 조합은 판매사업 부문이 기존의 관행적 업무가 아니고 개인기업에서 발휘되는 전략, 순발력 등이 필요한 고도의 생존 게임이라는 것을 점차 인식하고 있다.

아울러 유통활성화사업은 관료적 성격의 조합에서 탈피하여 모험적, 도전적인 기업가적 성격이 협동조합에 도입되기 시작하는 계기를 마련해 주었으며, 선진조합에서는 조합의 주인인 조합원들의 이윤추구 욕구를 만족시키기 위해 주체적인 마케팅을 수행하는 유통기관으로 탈바꿈되고 있고, 선진조합의 변화는 후발조합의 조합원들에게도 인식이 전환되는 계기를 제공하고 있다.

마. 물류합리화

산지유통전문조직은 소비지의 대형유통업체(도매시장의 도매상, 백화점, 대형할인점, 슈퍼마켓, 종합유통센터)와 함께 농식품 유통의 양대 축을 형성하고 있다. 유통활성화사업은 기존의 도매시장 의존적인 유통경로에서 벗어나 물류센터 혹은 대형유통업체와의 직거래 개설 및 농식품 유통단계를 대폭 축소하고, 물류흐름을 단순화 시키고 있어 물류합리화를 도모하고 있다.

제 2 절 농산물 유통정보화 현황

1. 유통정보의 중요성

유통정보는 생산자 및 유통업자의 의사결정을 지원하는 중요한 기능을 하며, 유통정보의 분산은 시장참여자간 공정경쟁을 촉진하는 데도 중요한 역할을 수행한다. 생산자 및 유통업체들은 유통정보의 도움으로 생산, 출하 시기 및 출하처를 결정하고 마케팅 전략을 수립하게 된다.

유통정보는 정확하고, 신뢰성이 있으며, 적합하고, 익명성이 있으며, 시의 적절하고, 신속해야 높은 가치가 있다(표 2-8). 농산물 유통에 있어서도 유

통정보의 중요성을 과소평가할 수 없으며, 유통정보의 정확성, 신뢰성, 적합성, 익명성, 시의적절성, 신속성 등을 높여 유통정보의 가치를 높여야 할 것이다.

<표 2-8> 유통정보의 평가기준

구 분	평가기준
정확성(accuracy)	실제 상황을 반영
신뢰성(trustworthy)	정보를 어느 한쪽에 치우침 없이 객관적으로 수집
적합성(relevance)	이용자의 필요에 따라 수집, 가공, 분산되어야 함
익명성(confidentiality)	정보공개 소스의 보호
시의적절성(timeliness)	이용자가 필요한 시기에 적합하게 제공
신속성(speed)	정보분산의 신속성

2. 산지유통 정보화 현황

우리나라의 농산물 유통은 정보 공개에 폐쇄적이며 규격화, 표준화 수준이 낮고 공동출하가 활성화되지 않아 고비용, 저효율 구조를 보여 왔으며, 소비지 시장의 요구를 반영해 산지에 상품을 만들어 공급하는 시장 변화에 수동적인 움직임을 보이고 있다. 농산물 산지유통은 영세한 수집상에 의해 주도되어 정보화가 제대로 이뤄지지 않고 있으며, 작목반, 농협, 영농조합법인 등 생산자조직의 정보화도 미진한 실정이고, 상인들은 법인화되지 않은 개인사업자로서 규모가 영세하고 자료의 공개를 극도로 꺼려 정보화를 회피하고 있다.

농협은 도매시장과 물류센터 등의 사후적인 정산자료를 확보하고 있으나 출하시점에서 정보화 시스템이 구축되지 않아 정보의 실시간 수집이 어려운 실정이고, 도매시장 출하시 송품장을 작성하고 있으나 문서가 전자화(EDI)되어 있지 않아 실시간 정보수집이 어렵고, 송품장도 도매시장별로 각기 다른 양식을 사용하여 정보의 표준화도 미비하다. 전산전문인력 부족,

정보화기기에 대한 투자여력 미흡 등 산지의 정보화 여건도 매우 취약하여 산지유통의 정보화를 제약하고 있으며, 이러한 정보화의 미비로 산지유통의 경우 가격, 출하량 등 출하의사결정에 필요한 정보의 실시간 수집과 분석이 어려운 실정이다. 산지공판장에서는 경매로 가격이 결정되기 때문에 산지가격으로서 대표성을 가지고 있으나 공판장 및 집하장의 전산시스템이 완전히 갖추어지지 않아 정보의 신속한 수집, 분산에 어려움이 있다.

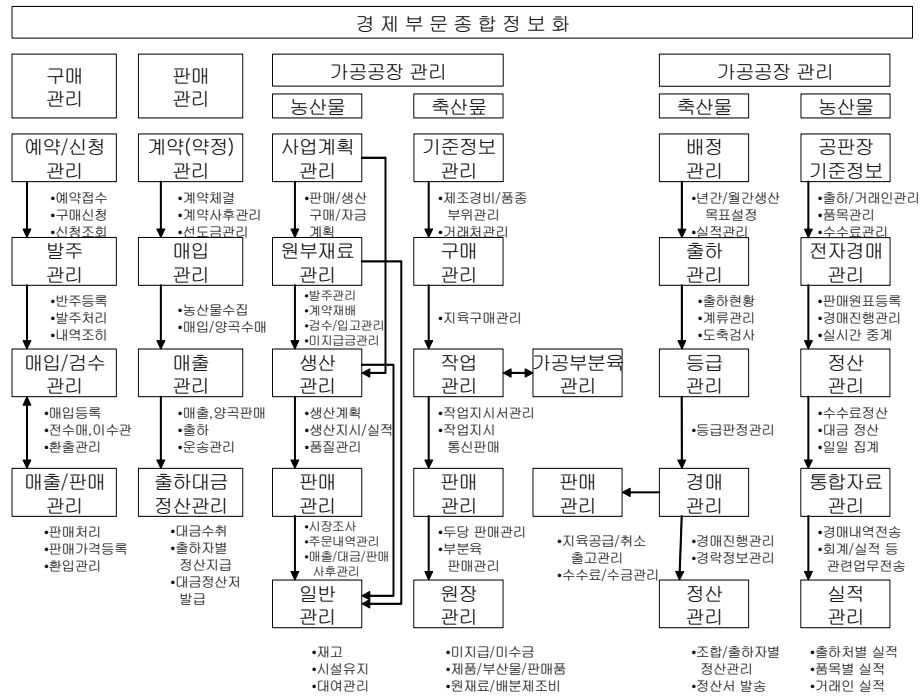
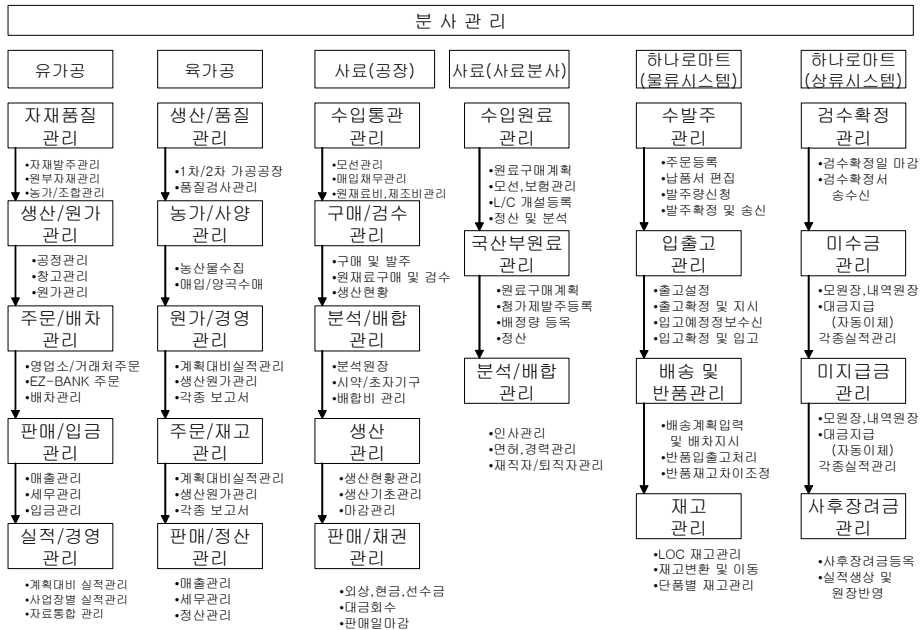
산지유통센터가 산지유통개선의 중추적인 역할을 수행할 것으로 기대되고 있으나 생산자의 출하관행, 운영주체의 운영시스템 미비, 전통적인 유통시설과의 경합, 소비지수용태세의 미비 등의 요인으로 운영이 아직까지 활성화되어 있지 않은 실정이다. 가동률이 낮은 문제점과 더불어 산지유통센터의 운영관리가 주로 수작업에 의존되어 있으며, 전산화의 수준이 매우 미약한 실정으로 운영관리의 효율성이 낮고 출하정보의 수집·분석이 미진하고 도매시장·물류센터 등 소비지 정보 및 식부면적, 작황 등 산지정보가 생산자들에게 전파되고 있지 못하고 있다. 아울러 생산자에 대한 정산 등이 신속하고 효율적으로 이루어지지 못하고 산지유통센터의 손익관리 등도 잘 이루어지지 않고 있다.

농협 등의 일부 산지유통센터에서는 대형유통센터의 등장에 대비한 업무프로세스 표준화/통합화 등의 종합 정보화를 구축하기 위한 많은 노력을 기울이며 업무 생산성 제고 및 종합적이고 일원화된 서비스 제공을 추구하고 있다.

가. 농협 경제종합정보시스템

농협에서는 산지유통정보화의 문제점을 인식하고 다음과 같은 '경제부문 종합정보화' 계획을 수립하여 회원농협 판매사업의 정보화를 추진하고 있다(그림 2-1).

<그림 2-1> 경제종합정보시스템 구성 현황



농협에서는 웹을 기반으로 하는 경제종합정보시스템을 추진하여 사용 중이며, 경제종합정보시스템을 운영함으로써 각 회원농협은 본/지소 및 각 사업장 계수를 자동 집계, 통합관리가 가능하여 업무처리의 효율성을 높일 수 있고, 표준 업무를 적용함으로써 업무 생산성을 이룰 수 있다. 그리고 조합원에게는 신용 부분과의 연계 처리로 개인별 관리 및 농산물시세정보, 농업관측정보 등의 정보를 제공할 수 있다.

농협의 경제종합정보시스템은 판매관리, 구매관리, 가공공장관리, 미곡처리장(RPC)관리, 공판장관리, 자금관리, 경영관리, 회계관리, 공통관리 등의 시스템으로 구성되어 있고, 산지유통의 주업무에 사용하는 판매관리시스템은 공동출하업무, 일반유통업무, 수출업무, 군납업무, 양곡수매업무, 채소수급 안정사업업무, 산지유통센터업무, 창고/저온저장고업무 등으로 구성되어 있다.

중앙회에서 추진하고 있는 경제사업 온라인망 구축사업은 전국적으로 일률적인 사업모델을 적용함에 따라 조합별 여건이 제대로 반영되지 못하는 등의 문제점이 있으며, 조합의 여건을 반영하여 시스템을 구축, 활용하기에는 시간이 걸릴 전망이다.

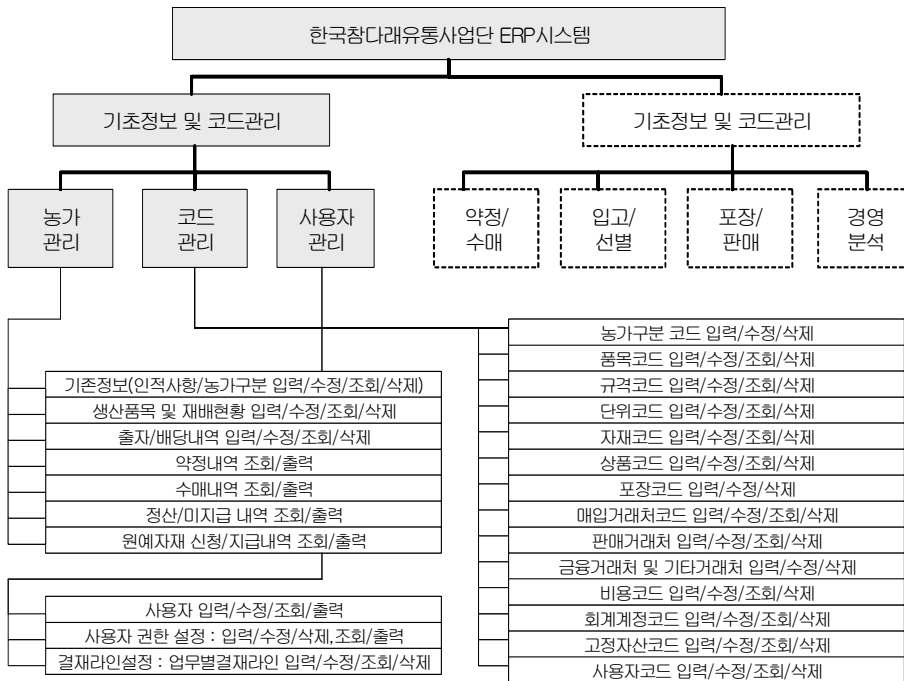
나. 산지유통센터 종합관리시스템(ERP)

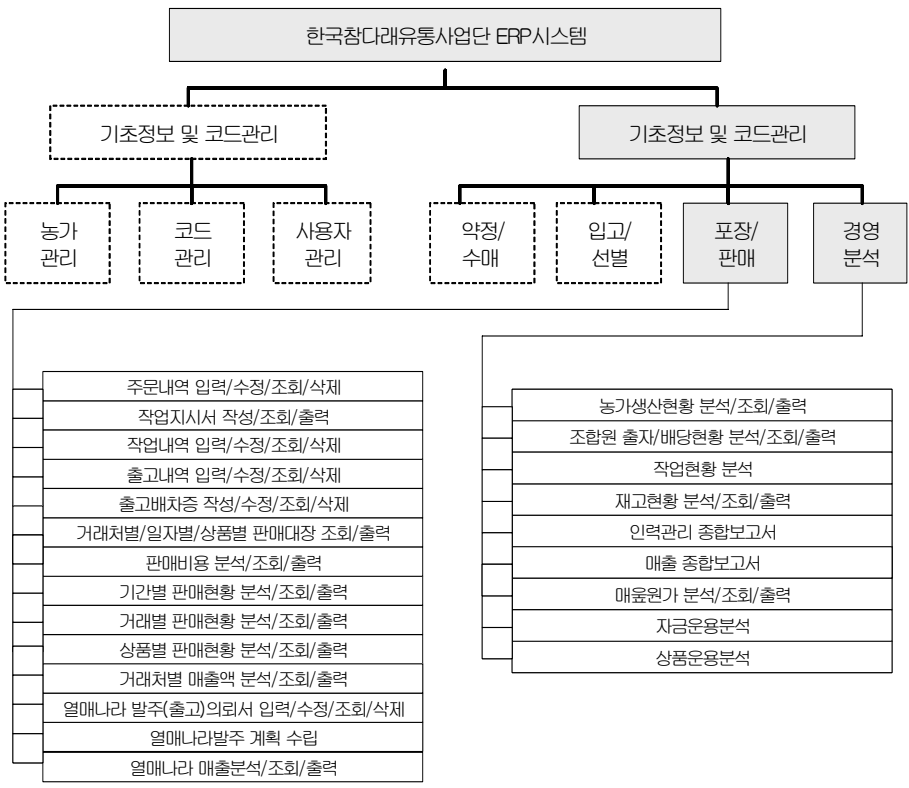
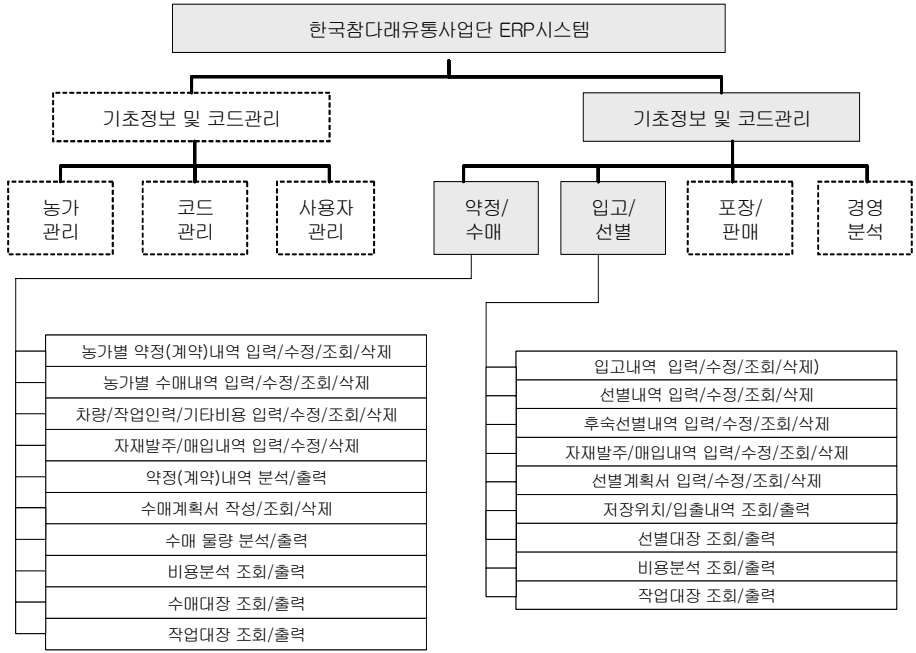
산지유통센터의 효율적인 정보화시스템 구축 예로는 한국참다래유통사업단을 말할 수 있다. 한국참다래유통사업단의 종합관리 시스템은 단순한 거래의 처리시스템(TPS), 업무의 자동화시스템(MIS)뿐만 아니라 경영관리자의 의사결정을 도와주는 의사결정시스템(DSS)차원의 농산물유통센터(APC)의 종합적인 관리 정보시스템으로서 웹서버를 통하여 전자상거래, 조합원에 대한 지도·유통정보서비스, 소비자에 대한 홍보서비스 등을 제

공하는 홈페이지 구축으로 생산, 가공, 유통, 경영 등 APC업무의 효율적 관리는 물론 신유통패러다임을 선도하는 정보시스템으로서의 역할을 수행하고 있다(그림 2-2).

한국참다래유통사업단이 종합관리시스템(ERP)을 사용함으로써 얻는 효과로는 업무처리시간의 단축과 업무효율의 극대화, 재고 및 파지 감소로 인한 비용절감, 정확한 업무분석에 따른 합리적인 계획수립, Lead-Time의 단축으로 고객만족도 제고, 자재 및 재고관리 혁신으로 관리비용절감 및 품질관리 향상 등이 있다.

<그림 2-2> 한국참다래유통사업단 ERP시스템 구성





3. 소비자유통 정보화 현황

가. 도매시장 정보화

1) 공영도매시장 정보화

공영도매시장 입주 도매법인 및 공판장들은 정산업무 및 관리업무를 전산화하였으나 도매시장 내 중도매인들은 규모가 영세하여 정보화가 미비한 실정이다. 가격, 물동량 등 유통정보도 자체적으로 수집되고 있으나 도매시장 관리사무소와의 네트워크 체계 미구축, 경매 후 조정의 필요성 등의 요인으로 정보의 실시간 수집체계가 불가능하고, 도매시장 법인별로 송품장의 형식이 다양하며, 상품코드가 표준화되어 있지 않아 정보의 활용성이 제한되고 있다.

정부는 경매과정의 투명성 제고와 경매에서 발생하는 정보를 신속하게 분산시키기 위해 전자경매의 도입을 강력히 추진하고 있다(표 2-9).

<표 2-9> 전자경매시스템의 정보내역

시스템명	제공서비스	정보통신기술
입하 정보	입하내역 등록	· 출하예약 · EDI/EC/CALS 기반시스템
	입하내역 조회	
	입하내역 통계	
경매 정보	경매순서 관리	· 수지식 무전표시스템 · 전자식 경매시스템 · 대형전광판, 터치스크린, 공중통신망, FAX, ARS를 통한 실시간 경매정보 조회
	경락가 입력	
	경매상황 조회	
	전광판 처리	
거래 정보	판매원표 조회/출력	· 인터넷/초고속통신망 활용 기반시스템
	일일시황 조회	
	일일경락 정보	
	통계 및 그래프	
	일일거래 동향분석	

기초 자료	표준송장	· 물류표준화
	농수산물코드표준화	
	유통정보 업무개선	
자료 송수신	실시간 경락결과	· 전용선망 활용(LAN, TCP/IP)
	송수신	
	일일 배치 경락결과 송수신	

2001년 말 63개 도매시장 법인에 전자경매시스템이 구축되어 경매과정의 투명성 제고에는 커다란 기여를 하고 있고, 가락시장 홈페이지 및 기타도매시장 홈페이지를 통한 실시간 정보전파시스템으로 경매결과를 실시간으로 전파하고 있다.

2) 유사도매시장 정보화

유사도매시장 위탁상인들은 전근대적인 상관행을 답습하여 자료공개를 극도로 꺼리고 있고 규모가 영세하여 정보화 수준이 극히 낮은 실정이고, 거래시 이용하는 양식은 간이영수증, 송품장, 판매일지 등이며, 이러한 양식을 사용하는 상회는 소수이고 대부분 메모장 등에 그날의 판매실적 등을 기록하는 원시적인 수준이다.

나. 종합유통센터 정보화

종합유통센터(물류센터)는 대형매장으로서 관리업무의 정보화는 물론 POS 등 판매시점 정보관리체계가 갖추어져 있다. 종합유통센터가 정보화를 통해 산지출하조직과의 연계성을 강화하여 운영의 효율성을 높여야 하나 정보화가 미흡한 실정이고, 소매 측면에서도 Data Warehouse, CRM 등 첨단유통정보시스템 활용에 의한 소비자 기호변화 분석 및 관련정보의 전파가 미흡한 실정이다.

농협이 운영하는 물류센터의 경우 유통관리시스템을 운영하고 있어 유통 관련 정보의 수집, 분산이 가능하나 시스템간 네트워크 체계가 구축되지 않아 종합적인 정보관리가 되지 않고 있고, 운영조직이 분사와 자회사로 이분화 되어 정보의 네트워크가 제약되고 있다.

종합유통센터는 소비자 거래처와는 전산수발주 시스템(EDI)이 활용되고 있으나, 산지출하조직과는 활용실태가 전무한 실정이다.

4. 농산물유통정보시스템

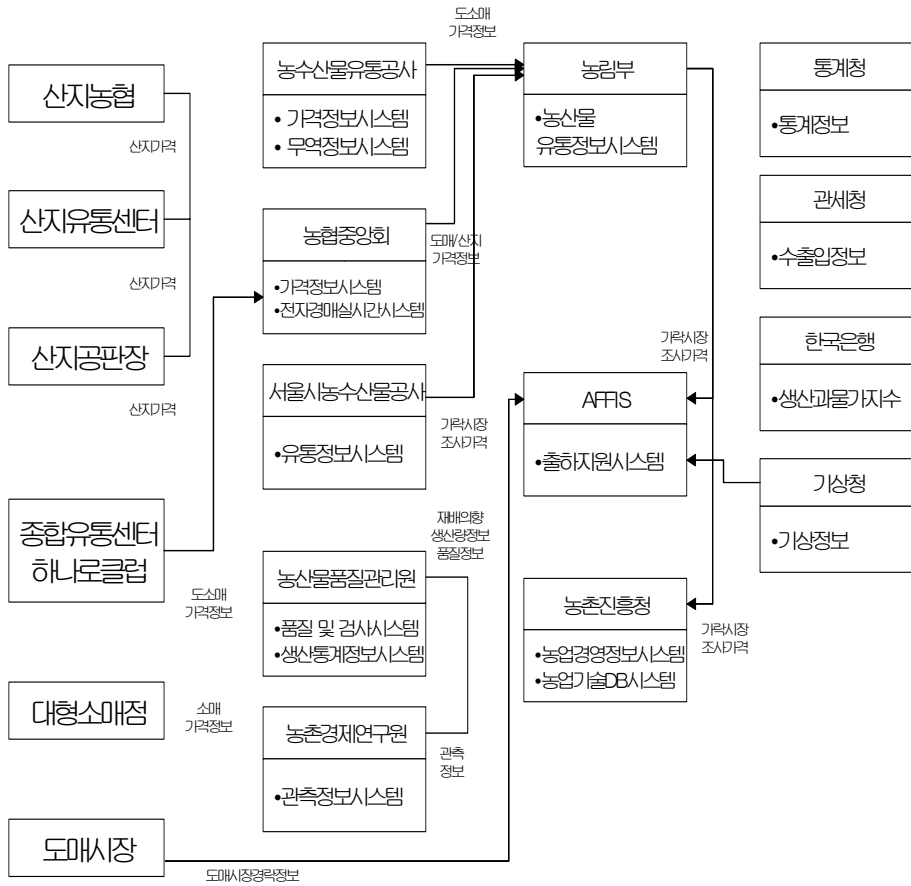
농수물유통정보시스템은 국내의 농산물 생산과 소비에 관한 정보를 신속하게 수집·분석하여 농업생산자에게 전달함으로써 농산물의 수급안정과 가격안정을 도모하려는 것이며, 농산물의 수급불안 요인 중의 하나로 시장에서 정보가 효율적으로 생성·분산되지 못한다는 데 있다는 인식 하에 보다 양질의 정보공급을 통해 수급과 가격을 안정시키려는 것이다.

농산물유통정보시스템은 농림부가 주관이 되어 산지 및 소비자 공판장은 농협, 도매시장 및 소매시장은 농수산물유통공사가 조사를 담당하며 이를 농림부가 집계하여 다양한 경로를 통해 분산시키고 있다. 기초 정보 수집은 농수산물유통공사, 농협중앙회, 서울시 농수산물공사, 농산물품질관리원, 한국농촌경제연구원, 기상청, 관세청, 통계청, 한국은행 등 9개 기관이 맡고 있으며, 정보를 취합하는 기관은 농림부 1개 기관이며, 분산기관은 농림수산정보센터, 농협, 농수산물유통공사, 농촌진흥청, 품질관리원 등으로 다양하다.

농림부의 유통정보시스템과 더불어 농협과 농협유통은 자체의 필요성에 따라 종합유통센터(물류센터)의 도매가격과 하나로클럽의 소매가격을 수집, 분산하고 있으며, 농수산물유통공사는 백화점, 할인점 등 대형유통업체의

가격자료를 수집, 분산하고 있다.

<그림 2-3> 농산물유통정보시스템의 흐름도



<표 2-10> 기관별 유통정보 현황

기관명	주요 정보내역
농림부/AFFIS	농업정책정보, 통계정보, 관측정보, 병충해정보, 해외 농업정보, 유통정보
농수산물유통공사	국내가격/시황, 해외가격, 수출입통제, 국내생산통계, 해외생산통계, 해외시장속보, 수출입정보, 국가별정보, 일본정보, 대형유통업체가격정보
농협중앙회	농업경영정보, 농산물가격정보, 날씨정보
농협유통	농산물시세정보(도매가격 및 하나로클럽 소매가격), 종합유통센터안내, 우리농산물 정보
서울특별시 농수산물공사	가격정보, 조사가격정보, 정산가격정보, 물량정보, 출하지정보, 판매실시간중계
농촌진흥청	영농정보(기상정보, 방제기상, 병해충예찰정보 등), 신농업기술정보, 해외농업정보, 농업경영정보, 영농기술보급, 농업기술 DB
농촌경제연구원	주요 농산물 관측정보 및 농업경제 연구정보
국립농산물 품질관리원	우리농산물 식별정보, 품질인증농산물정보, 농산물표준규격, 농약 잔류허용기준, 관련법규, 통계정보
서울시	장바구니물가정보, 직거래장터 정보
기상청	특보 및 경보, 예보, 현재의 날씨, 생활기상정보
한국은행	생산자 물가지수
통계청	소비자물가동향

제 3 절 산지유통전문조직의 정보화 실태 및 추진 전략

1. 조사개요

산지유통전문조직의 정보인프라 현황과 경제종합정보시스템의 사용 현황, 종합관리시스템(ERP)의 도입 의향, 모바일시스템의 도입 의향을 조사하기 위해 표본 업체 100개를 선정하여 전화조사를 실시하였다.

2. 산지유통전문조직의 정보화 실태

가. 조사업체 개황

1) 지역별 현황

조사업체의 지역별 분포는 경북 20%, 전남 18%, 강원 13%, 충남 13%, 경남 11%, 인천 및 경기 8%, 전북 7%, 충북 5%, 제주 5%의 순이다(표 2-11).

<표 2-11> 지역별 조사업체 분포

구 분	조합수	비율(%)
인천 및 경기	8	8.0
강원	13	13.0
충북	5	5.0
충남	13	13.0
전북	7	7.0
전남	18	18.0
경북	20	20.0
경남	11	11.0
제주	5	5.0
합 계	100	100

2). 조직형태별 현황

지역농협 형태 97개소, 연합사업 형태 3개소 순으로 농협을 중심으로 하여 조사하였다(표 2-12).

<표 2-12> 조직 형태별 현황

구 분	조합수	비율(%)
지역농협	97	97.0
연합사업	3	3.0
합 계	100	100

3) 산지유통전문조직 선정 연도별 현황

조사업체 중 산지유통전문조직으로 선정된 연도별 현황으로 2000년에 산지유통전문조직으로 선정된 조합이 37개소, 2001년도에 선정된 조합이 23개소, 2002년에 선정된 조합이 40개소이다(표 2-13).

<표 2-13> 산지유통전문조직 선정 연도별 현황

구 분	조합수	비율(%)
2000년	37	37.0
2001년	23	23.0
2002년	40	40.0
합 계	100	100

나. 산지유통전문조직의 정보인프라 현황

1) 컴퓨터 및 프린터 보유 현황(농산물 판매업무용)

산지유통전문조직은 조합마다 농산물 판매 업무만을 위해 사용하는 컴퓨터를 약 2.9대 보유하고 있으며, 팬티엄Ⅲ와 팬티엄Ⅳ의 비율은 각각 52.2%와 47.8% 이다(표 2-14).

산지유통전문조직은 조합마다 농산물 판매 업무만을 위해 사용하는 프린터를 약 1.7대 보유하고 있으며, 레이저 프린터와 잉크젯 프린터의 비율은 각각 98.2%와 1.8% 이다.

<표 2-14> 컴퓨터 및 프린터 보유 현황

구 분		대수	비율(%)	합계	평균
컴퓨터	팬티엄Ⅲ	153	52.2	293	2.93
	팬티엄Ⅳ	140	47.8		
프린터	레이저	165	98.2	168	1.68
	잉크	3	1.8		

2) 네트워크 통신망 현황

모든 조합에서 농협중앙회통신망을 사용하고 있으며, 2가지 통신망을 같이 사용하고 있는 조합은 12개로 나타났다. 2가지 통신망을 같이 사용하는 이유로는 본사와 작업장이 별도로 떨어져 있어 본사와 작업장 간의 통신망 연결을 위해 사용하는 경우와 농협중앙회통신망의 시스템 오류시 타통신망을 통해 업무를 보조하기 위해서 사용하는 것으로 나타났으며, 통신망 사용 용도는 경제사업 업무와 인터넷으로 나타났다. 조합자체 LAN을 사용하는 조합은 8개소로 조합자체 LAN의 사용은 미진한 것으로 나타났다(표 2-15).

<표 2-15> 네트워크 통신망 현황

구 분		조합수	사용 용도
외부	농협중앙회통신망	100	경제사업 및 인터넷
	KT 매가패스	4	경제사업 및 인터넷
내부	조합자체LAN	8	경제사업 및 인터넷

3) 소프트웨어 사용 현황

소프트웨어 사용 현황 중 엑셀은 100개 조합 모두 사용 중이며 보고서 작성, 정산, 집계와 입고·출하·선별·포장관리 등에서 이용되고 있고, 훈민정음도 100개의 조합에서 사용 중이며 보고서작성에 이용되고 있다. 한글은 35개의 조합에서 사용 중이며, 보고서작성에 이용되고 있다. 파워포인트는 20개의 조합에서 사용 중이며 업무보고에 이용되고 있다. 주로 사용하고 있는 소프트웨어는 오피스 위주의 소프트웨어로 ACCESS 등 데이터베이스 소프트웨어의 사용 현황이 미약한 실정이다(표 2-16).

<표 2-16> 소프트웨어 사용 현황

구 분	조합수	사용 용도
엑셀	100	보고서 작성, 정산, 집계 등
훈민정음	100	보고서 작성
한글	35	보고서 작성
파워포인트	20	업무 보고

4) 자체 정보시스템 보유 현황

자체정보시스템을 보유하고 있는 조합은 3개소 뿐이며 그 사용분야는 용역관리와 공판장관리에 적용하여 사용하고 있다. 용역관리시스템은 일용직 용역들의 일당 지급에 대한 정산을 관리하기 위해 사용하고 있고, 공판장 관리시스템은 공판장 현황과 운영을 관리하기 위해 사용하고 있다(표 2-17).

자체정보시스템을 보유하고 있지 않은 조합들은 기존에 사용하고 있는 경제종합정보시스템에 큰 불편함을 느끼지 못하고 있고, 자체정보시스템 도입 시 농협중앙회, 회원농협, 농협자회사 등과 업무 연계가 되지 않기 때

문에 자체정보시스템 도입의 필요성을 느끼지 못하고 있는 실정이다.

<표 2-17> 자체 정보시스템 보유 현황

구 분	조합수	사용 분야
자체정보시스템 보유	3	용역관리(1) 공판장관리(2)
자체정보시스템 미보유	97	.

다. 경제종합정보시스템의 사용 현황

1) 경제종합정보시스템의 사용 분야

산지유통전문조직에서 경제종합정보시스템은 경제사업에 필요한 모든 업무분야에 사용되고 있고, 현재 모든 조합에 100% 정착되어 있는 상태이며, 모든 업무에 경제종합정보시스템이 적용되고 있다.

경제종합정보시스템으로 모든 업무처리가 가능하지만, 현재 12개 조합에서 입고·출하·선별·포장관리 등에서 경제종합정보시스템과 엑셀을 같이 병행하여 사용하고 있다. 경제종합정보시스템이 정산위주의 시스템으로 되어 있어 입고·출하·선별·포장관리 등의 업무를 잘 반영하지 못하고 있다.

2) 경제종합정보시스템의 사용 효과

경제종합정보시스템의 사용 효과에 대한 응답 현황으로 “정산이 편리함”이 전체 응답의 28.8%로 가장 높았고, 모든 시스템이 정산과 연계되어 있어서 정산 업무에서 가장 많은 편리성을 가지고 있는 것으로 나타났다(표 2-18).

“자료의 입·출력이 편리함”에 대한 응답 현황은 전체 응답의 16.5%로 과거 수기로 작업을 하던 것과 비교하여 볼 때 상당한 편리함을 가지고 있는 것을 알 수 있다.

“실적관리가 편리함”에 대한 응답 현황은 전체 응답의 11.7%로 과거 실적들을 편리하게 조회할 수 있고, 실적결과를 한눈에 알아볼 수 있기 때문이라고 응답하였다.

“자료(장부)관리가 편리함”에 대한 응답은 10.9%로 과거 종이문서로 작성되어 관리해오던 장부들이 경제종합정보시스템 안에서 관리되기 때문에 편리하다고 응답하였다.

“사용효과 모르겠음”에 대한 응답은 4.7%로 이들 응답자들은 경제종합정보시스템 사용 시 업무가 번거롭고, 타거래처와 연계가 안 된다는 등의 문제점을 가지고 있다고 응답하였다.

기타 사용 효과로는 "업무의 간소화", “모든 업무와 연계됨”, “재고과약이 용이함”등이 있으며 각각 전체 응답의 2.7%, 1.3%, 0.7%를 차지하였다.

<표 2-18> 경제종합정보시스템의 사용 효과

구 분	응답현황	비율(%)
정산이 편리함	42	28.8
자료의 입·출력이 편리함	24	16.5
실적관리가 편리함	17	11.7
업무 속도가 빨라짐	17	11.7
자료(장부)관리가 편리함	16	10.9
정보조회가 편리함	16	10.9
사용효과 모르겠음	7	4.7
기 타	7	4.7
합 계	146	100

3) 경제종합정보시스템의 문제점

“조합의 특성을 맞추지 못함”에 대한 응답은 19.2%로 조합마다 주 품목, 규격, 경영방법 등이 다른데 각각의 조합이 일률적인 시스템에 맞추어서 작업을 해야 하기 때문에 각 조합의 특성을 살리지 못하고 시스템에 맞추어서 작업한다고 응답하였다. 따라서 각각의 산지조합들의 특성에 맞는 시스템 개발과 도입이 시급한 실정이다(표 2-19).

“속도 느림”에 대한 응답은 19.2%로 경제종합정보시스템은 웹을 기반으로 하고 있기 때문에 마감시간 등 업무량이 많이 집중되고, 사용자가 많은 시간에는 시스템 연결과 속도가 느린 것으로 나타났다.

“업무가 번거로움”에 대한 응답은 19.2%로 공동계산 시 대표출하주로 정산 후 조합에서 다시 개별조합원들에게 정산을 해야 하고, 경제종합정보시스템과 연계가 되지 않는 품목들은 엑셀에서 작업 후 다시 경제종합정보시스템에 적용하여 작업하는 등 2중 작업을 하기 때문에 업무가 번거로운 것으로 나타났다.

“사용이 어려움”에 대한 응답은 9.6%로 경제종합정보시스템에 대한 직접적인 교육 부족으로 사용자가 이용에 어려움을 겪고 있다고 응답하였다. 따라서 전체적인 사용 매뉴얼의 보급과 중앙회에서 직접적인 교육이 요구되고 있다.

“버그(오류)가 있음”에 대한 응답은 9.6%로 웹상에서 경제종합정보시스템을 사용할 때 시스템 다운 등의 버그(오류)가 있는 것으로 나타났다.

“중앙회에서 정보 입력이 늦음”에 대한 응답은 5.8%로 경제종합정보시스템으로 농산물 가격에 대한 시세 등의 정보를 조회할 수 있으나 중앙회에서 정보 입력이 늦기 때문에 정보를 충분히 활용할 수 없다고 응답하였다.

“통계자료에 신뢰성이 없음”에 대한 응답은 3.9%로 경제종합정보시스템이 제공하는 자료가 현실과 잘 맞지 않기 때문에 경제종합정보시스템의 통계자료를 사용하기 보다는 주로 인터넷을 통해 정보 조회를 하는 것으로 나타났다.

<표 2-19> 경제종합정보시스템의 문제점

구 분	응답현황	비율(%)
조합 특성을 맞추지 못함	10	19.2
속도 느림	10	19.2
업무가 번거로움	10	19.2
사용이 어려움	5	9.6
버그(오류)가 있음	5	9.6
정보분석 및 활용 미흡	3	5.8
중앙회에서 정보 입력이 늦음	3	5.8
통계자료가 신뢰성이 없음	2	3.9
기 타	4	7.7
합 계	52	100

라. 종합관리시스템(ERP)의 도입 의향

1) 종합관리시스템(ERP)의 도입 의향

종합관리시스템(ERP)의 도입 의향에 대해 69개의 조합에서 도입 의향을 가지고 있지 않다고 응답하고, 10개의 조합에서만 종합관리시스템(ERP)의 도입 의향을 가지고 있다고 응답하였다(표 2-20). 대부분의 조합들이 새로운 시스템의 도입에 대해 부정적인 견해를 가지고 있는 것으로 나타났으며, 이는 정보화 마인드 부족과 정보시스템에 대한 인식이 낮기 때문이다. 따라서 중앙회 전산본부에서는 시스템 도입으로 얻는 실질적인 기대효과 등의 정보를 제공하고 집적적인 교육을 통하여 조합들의 정보화 마인드를 높이는 것이 요구되고 있다.

<표 2-20> 종합관리시스템(ERP)의 도입 의향

구 분	조합수	비율(%)
필요하다	10	10.0
보 통	22	22.0
필요없다	69	69.0
합 계	100	100

2) 종합관리시스템(ERP)의 도입 시 예상 효과

종합관리시스템(ERP)의 도입 시 예상효과로 “부서간, 담당자간 정보 공유 가능”에 대한 응답이 전체응답의 40%로 가장 높은 것으로 나타났고, 정보 공유를 통해 경영의 합리화가 진전될 것이라고 응답하였다(표 2-21).

“분석과 예측이 용이함”에 대한 응답은 15%로 기존정보시스템으로는 수집된 정보에 대한 분석이 미비하고, 수요예측에 따른 출하 등 경영의 과학화가 잘 이루어지지 않는 것으로 나타났다. 따라서 정보의 분석과 예측을 통한 경영의 과학화를 이룰 수 있는 시스템 구축이 요구되어 진다.

<표 2-21> 종합관리시스템(ERP)의 도입 시 예상효과

구 분	조합수	비율(%)
부서간, 담당자간 정보공유 가능	8	40
업무의 연계성 확보	4	20
업무시간의 단축	3	15
분석과 예측이 용이	3	15
업무의 중복과 지체 탈피	2	10
합 계	20	100

3) 종합관리시스템(ERP)의 도입 의향이 없는 이유

종합관리시스템(ERP) 도입의향이 없는 이유 중“필요성 못느낌”에 대한 응답이 60%로 가장 높았다. 현재 사용하고 있는 경제종합정보시스템에 대부분 만족하며 사용하고 있고 정보화 마인드의 부족으로 다른 시스템 도입에 대한 필요성을 느끼지 못하고 있다(표 2-22).

“시스템에 대한 이해 부족”에 대한 응답은 32%로 조합들이 아직까지 종합관리시스템에 대한 인식이 낮은 것으로 나타났다. 이들 응답자 중에서 경제종합정보시스템도 사용 권한을 부여 받은 경우 부서별 정보를 조회 할 수 있고, 시스템 이용도 가능하기 때문에 두 시스템의 차이점을 잘 모르겠다고 응답하였다.

기타 응답 현황으로는“도입이 번거로움”, “구축 후 효율성에 대한 불확실성”이 있으며, 각각 1.1%를 차지하였다.

<표 2-22> 전사적자원관리(ERP) 도입의향이 없는 이유

구 분	조합수	비율(%)
필요성 못느낌	54	60.0
시스템에 대한 이해 부족	28	31.2
거래처와 연계 미흡	2	2.2
조합의 규모가 작아서	2	2.2
구축비용과 시간에 대한 투자	2	2.2
기 타	2	2.2
합 계	90	100

마. 모바일시스템의 도입 의향

1) 모바일시스템의 도입 의향

모바일시스템의 도입 의향에 대해 65개의 조합에서 모바일시스템의 도입이 필요 없다고 응답하였고, 26개의 조합에서 모바일시스템의 도입 의향을 가지고 있다고 응답하였다(표 2-23). 종합관리시스템(ERP)의 도입 의향과 마찬가지로 다른 시스템의 도입에 부정적인 견해를 가지고 있고, 조합 규모가 영세하기 때문에 정보화에 대한 필요성을 인식하지 못하고 있으며 정보화 마인드도 미흡한 것으로 나타났다. 따라서 중앙회 전산본부에서는 시스템 도입으로 얻는 실제적인 기대효과 등의 정보를 제공하고 집적적인 교육을 통하여 조합들의 정보화 인식을 높이는 것이 시급한 실정이다.

<표 2-23> 모바일시스템의 도입 의향

구 분	조합수	비율(%)
필요하다	26	26.0
보 통	10	10.0
필요없다	65	65.0
합 계	100	100

2) 모바일시스템의 도입 의향이 있는 이유

모바일시스템의 도입 이유로 “현장에서 이용 가능”에 대한 응답이 39.3% 가장 높았다. 이는 산지 특성상 현장에서 업무하는 시간이 많기 때문에 모바일시스템 도입으로 현장에서 작업하는 작업자가 바로 정보 입력에서부터 조회, 결제까지 가능하여 업무의 효율성을 높일 수 있을 것으로 나타났다(표 2-24).

“정보과약이 용이함”에 대한 응답은 28.5%로 농산물 가격 시세와 날씨 등의 정보를 언제 어디서나 즉시 조회할 수 있기 때문에 편리할 것이라고 응답하였다.

“있으면 편리할 것 같음”에 대한 응답은 17.8%로 시스템에 대한 이해는 부족하지만 있으면 편리할 것 같다고 응답하였다. 이 응답자들은 정보

시스템에 대한 인식은 낮지만 긍정적인 사고를 가지고 있다. 따라서 이 응답자들에게 모바일시스템의 효과에 대한 교육을 통해 확실한 인식을 심어 주는 것이 필요하다.

<표 2-24> 모바일시스템의 도입 의향이 있는 이유

구 분	조합수	비율(%)
현장에서 이용 가능	11	39.3
정보파악이 용이함	8	28.5
있으면 편할 것 같음	5	17.8
정산이 편리함	1	3.6
빠른 대처	1	3.6
도입이 용이함	1	3.6
추세에 따라서	1	3.6
합 계	28	100

3) 모바일시스템 도입 시 적용 분야

모바일시스템의 도입 시 적용 분야에 대한 응답으로 생산관리, 판매관리, 출하관리, 입고관리, 선별관리 등 현장에서 업무를 적용할 수 있는 분야가 비교적 높은 비중을 차지하였다(표 2-25). 모바일시스템 사용 시 언제 어디서나 시스템에 접속하여 업무를 할 수 있는 효과 등으로 인하여 현장에서 업무를 적용할 수 있는 분야에 대한 응답이 높게 나타났다.

<표 2-25> 모바일시스템 도입 시 적용 분야

구 분	조합수	비율(%)
생산관리	23	12.0
판매관리	23	12.0
출하관리	22	11.6
입고관리	20	10.6
선별관리	17	9.0

포장관리	16	8.3
주문관리	15	7.8
창고관리	14	7.3
조합원관리	14	7.3
거래처관리	14	7.3
정산관리	13	6.8
합 계	191	100

4) 모바일시스템의 도입 시 예상효과

모바일시스템의 도입 시 예상되는 효과로 “현장에서 발생하는 정보수집이 가능함”에 대한 응답이 전체 응답의 68.8%로 가장 높았다(표 2-26). 작업자들이 현장에서 작업하는 시간이 많기 때문에 현장에서 발생하는 정보 수집에 대한 필요성을 많이 가지고 있는 것으로 나타났다.

“시스템 연결이 신속함”에 대한 응답은 22.9%로 모바일시스템 도입으로 언제 어디서나 정보시스템에 바로 접속할 수 있기 때문에 정보시스템에 연결이 신속하고, 그에 따라 업무시간의 단축 등 업무의 효율성이 증대될 것으로 나타났다.

<표 2-26> 모바일시스템의 도입 시 예상효과

구 분	조합수	비율(%)
현장에서 발생하는 정보수집 가능함	24	68.6
시스템 연결이 신속함	8	22.9
농산물 수배송차량 위치추적 가능함	2	5.7
대금결제가 용이함	1	2.8
합 계	35	100

5) 모바일시스템 도입 의향이 없는 이유

“사용 시 불편함”에 대한 응답은 17.9%로 모바일시스템은 PDA, 핸드폰 등을 통해 작업 해야 하는데 무선단말기가 출력할 수 있는 화면 크기가 작고, 자료 입력 등에서 불편할 것이라고 응답하였다(표 2-27).

“인프라구축이 안되어 있음”에 대한 응답은 17.9%로 공산품 제조업과는 달리 농산물 유통은 전체적인 표준화가 이루어지지 않았기 때문에 표준화 등 인프라 구축이 되어있지 않은 상황에서 모바일시스템의 도입은 효율성이 없을 것이라고 응답하였다.

“조합원들이 사용하기 어려움”에 대한 응답은 12.8%로 조합원들의 노령화로 모바일시스템에 대한 인식이 부족하여 사용하기 어려울 것이라고 응답하였다.

기타 응답 현황으로는 “보안상의 문제점”, “업무처리가 느림”, “현장에서 모바일시스템 필요성 못 느낌”등이 있으며, 각각 5.2%, 2.5%를 차지하였다.

<표 2-27> 모바일시스템 도입의향을 가지고 있는 않은 이유

구 분	조합수	비율(%)
사용시 불편함	7	17.9
인프라구축이 안되어 있음	7	17.9
조합원들이 사용하기 어려움	5	12.8
아직은 이른 것 같음	5	12.8
시스템에 대한 이해 부족함	5	12.8
생각 안해봤음	3	7.7
구축비용과 시간에 대한 투자	2	5.2
도입 후 효율성에 대한 불확실성	2	5.2
기 타	3	7.7
합 계	39	100

3. 산지유통전문조직 정보화에 대한 평가

투자여력이 부족하고, 정보화에 대한 필요성을 크게 인식하지 못하는 등 산지유통전문조직은 규모의 영세성으로 인해 정보화 대한 인식이 상당히 낮고 부족한 상태이며, 자체적으로 정보시스템의 필요성을 느껴서 적극적으로 정보시스템을 도입하려고 하기 보다는 상부 조직의 행동에 따라 움직이는 등 정보시스템의 도입에 있어서 수동적인 움직임을 보이고 있는 실정이다.

산지유통전문조직에서 사용하고 있는 소프트웨어 현황은 엑셀, 훈민정음, 한글, 파워포인트 등 주로 오피스 위주의 소프트웨어로서 수집된 정보를 가공 및 분석하여 합리적인 경영을 하도록 도와주는 데이터베이스 소프트웨어와 입고, 출하, 선별, 포장 등의 업무에 적용되는 소프트웨어의 사용이 미비하고 취약한 것으로 나타났다.

산지유통전문조직의 회원농협들은 정보의 분석에 따른 경영의 과학화가 미진한 실정이다. 출하결과를 포함한 수집 정보를 분석 및 데이터베이스화 하여 출하처 및 출하량 선정 등 판매정책을 효과적으로 수립하는 등의 전략적인 업무수행이 미비한 실정이다. 아울러 경제종합정보시스템을 정보의 기록 및 저장 기능 위주로 사용하고 있어 정보가공 기능에 대한 사용이 미비한 실정이다. 또한 경제종합정보시스템에서 제공하는 정보들이 현실과 잘 맞지 않고, 중앙회 전산조직에서 정보 입력이 늦는 등 정보 활용을 잘 하지 못하고 있다.

농협은 웹아리오피스라는 EDI시스템을 추진하여 회원농협에 보급하였으나 EDI시스템의 사용 현황은 대부분 미흡한 실정이다. EDI시스템이 농협 중앙회, 회원농협, 농협자회사와 연계가 되어있지만 타 거래처와는 연계되

어 있지 않기 때문에 타 거래처와는 전화나 팩스, e-mail 등으로 수발주 및 문서 송·수신을 할 수 밖에 없는 상태이고, 더욱이 농협중앙회, 회원농협, 농협자회사와의 수발주 및 문서 송·수신 등 업무 수행에 있어 대부분의 조합들은 EDI시스템을 활용하기 보다는 전화, 팩스, e-mail의 사용을 선호하고 있다. 이는 직원의 전산능력이 취약하고 중앙회에서 시스템에 대한 교육과 유지관리를 철저히 하지 않는 것이 주요인이라고 할 수 있다.

산지유통전문조직의 일부 회원농협들은 매취사업 보다는 수탁(일반유통) 사업에 치중하고 있어 입고, 출하, 선별, 포장 등의 업무를 작목반과 영농법인, 개별농가에서 직접 관리하고 있다. 이들은 규모가 영세하고 자료의 공개를 꺼려 정보화를 회피하고 있으며, 전산전문인력이 부족하고, 정보화 기기에 대한 투자여력 미흡 등 정보화가 미진하여 산지유통의 정보화를 제약하고 있다.

4. 산지유통전문조직 정보화 추진 전략

산지유통전문조직의 정보화는 산지유통전문조직의 운영효율성을 높이고 다양한 부가서비스를 생산자에게 제공하여 생산자들의 편익을 제고시킬 수 있는 시스템으로 개발한다.

산지유통전문조직의 구성원들은 낮은 정보화 마인드로 인하여 정보화에 대한 필요성을 크게 인식하지 못하고 있고, 정보시스템 도입에 있어 수동적인 움직임을 보이고 있어 이는 산지유통전문조직의 정보화에 큰 걸림돌로 작용하고 있다. 이를 위하여 정보시스템에 전문성을 갖춘 인력 보강 및 정보화 교육 프로그램 등을 통하여 조합장 및 조합원들의 정보화 마인드를 제고시키는 것이 매우 시급하다.

산지유통전문조직은 하드웨어적인 유통시설의 확충으로 인해 정보화에 대한 투자가 미비한 실정이다. 앞으로는 지식정보화 기반을 활용하는 소프트웨어 중심으로 운영효율화를 도모해야 한다. 현재 산지유통전문조직에서 사용하고 있는 소프트웨어는 오피스 위주의 간단한 소프트웨어로서 앞으로는 데이터베이스 소프트웨어 등 응용 소프트웨어 사용으로 수집된 정보를 가공 및 분석하여 합리적인 경영을 할 수 있도록 도모해야 한다.

또한 농협이 추진하고 있는 경제사업 온라인망 구축사업은 전국적으로 일률적인 사업모델을 적용함에 따라 조합별 여건이 제대로 반영되지 못하고 있는 실정이다. 따라서 각각의 조합여건을 반영할 수 있는 시스템 구축이 필요하다.

산지유통전문조직은 on-line 유통체계의 개선을 위해서 전자수발주시스템, 공급망관리(supply chain manarement)등 정보기술 활용 및 전자상거래 기반조성으로 효율성을 제고해야 하며, ERP시스템, 모바일시스템 등의 도입으로 산지의 정보화를 촉진시켜야한다.

따라서 산지유통전문조직은 모바일 정보시스템을 개발, 이용하여 운영의 효율성을 높이고 유통정보 제공 등 부가기능을 수행할 필요가 있다. 모바일시스템은 산지유통전문조직의 종사자들에게 있어서 언제 어디서나 데이터베이스에 접근하여 원하는 정보를 등록, 수정, 삭제, 조회가 가능하며 체계적인 관리 및 분석이 가능하게 하고, 전반적인 유통업무 환경의 질적 개선을 통해 생산성 향상과 효율성 증대를 꾀할 수 있으며, 데이터의 실시간 처리 및 고객 대응력 제고 등 산지유통전문조직의 경쟁력과 통제력을 향상시킬 수 있는 획기적인 대안이 될 것이다.

제 4절 미국의 농산물 산지 정보관리시스템 운영 실태와 사례

1. 산지농산물 유통 소프트웨어 실태

미국의 청과물 선과장(packing house)들은 회계, 판매, 재무, 재고관리 등 경영정보의 통합관리를 위해 특별히 고안된 소프트웨어를 설치운영하고 있다.

대부분의 관련프로그램은 기본 소프트웨어로서 마이크로소프트 윈도우와 IBM의 AS/400 중급(midrange)시스템을 이용하여 데스크탑 컴퓨터상에서 이루어지고 있으며, 소프트웨어 프로그램은 적용이 쉬울 뿐만 아니라 포괄적이어서 선과장의 경영관리에 유용하게 사용되고 있다.

2. 선과장의 정보관리시스템 소프트웨어 : Spokane SoftWare System(SSS)

SSS는 워싱턴주 사과 선과장에서 대표적으로 사용되고 있는 소프트웨어 시스템으로서 1975년 이래 선과장 뿐만 아니라 농산물 가공업자, 식품브로커, 수송업자 등 애그리 비즈니스분야에서 적용되어 왔는데 워싱턴 주내 100여 업체가 이 시스템을 사용하고 있다.

소프트웨어 시스템의 적용분야는 ①일반원장관리(general ledger), ② 발주 및 발송관리(order entry & invoicing), ③회계관리(accounts Payable), ④ 바코드관리(bar code) ⑤ 급여관리(payroll) ⑥ 공동회계관리(lot & Pool Processing) ⑦ 생산자 원장관리(Grower Ledger) ⑧ 판매관리 및 팔렛팅 등이 있다.

가. 핵심 소프트웨어 시스템

1) 발주 및 송장 모듈(Orde Entry & Invoicing Module)

송장모듈을 이용하여 재무자료 생성 하부시스템은 일반원장회계시스템과 통합관리되는데 송장처리(invoicing), 신용관리기록(credit memos), 발송취소 및 재발송(voids/ re-invoicing)이 하부구조시스템에 해당된다.

일반적으로 업체의 상황에 맞게 제공된 프로그램은 신축적으로 조정되는데 예를 들면 포장과일의 재고조사프로그램은 재고 데이터베이스를 다음과 같이 업체의 필요에 따라 ① 품목/품종/창고/지역/등급, ② 지역/품목/품종/등급, ③ 등급/품목/품종/창고/지역 등으로 다양하게 특성별로 우선순위를 두고 구축할 수 있도록 하고 있다.

2) 바코드 재고관리 모듈(Bar Code Inventory Control Module)

바코드 재고관리모듈은 생산자, 포장업자, 수송업자에게 완벽한 재고관리를 가능케하는 관리시스템으로 바코드 일련번호를 이용하여 생산물 선적에서부터 최종인수에 이르기까지 추적이 가능하다. 바코드의 일련번호는 상품, 품종, 등급, 크기, 포장형태, 생산년도 등의 정보를 담고 있으며, 포장지역번호, 인수자번호, 판매번호 등도 표기되어 있다. 이외에도 부착표에는 인수, 선별, 포장 등의 날짜 및 시간이 표시된다. 본 프로그램은 운영업체의 재량으로 검사번호, 수출품질, 포장라인번호 등 최대 6개까지 정보를 추가해서 입력하여 추적할 수 있도록 고안되어 있다.

농산물이 산지 수집과정과 포장단계에서 다른 사람 생산품목과 섞일 수도 있으나 특정 농가와 특정과수원에서 생산된 품목은 개별 상자에 담겨

있어 재고관리가 용이하다. 바코드시스템은 창고위치의 추적도 가능하게 하며, 선과장 운영자는 소형저장창고를 여러 곳에 두고 위치를 프로그램에 입력함으로써 대형창고 하나에만 의존하기보다는 다양한 형태의 창고를 확보함으로써 창고관리를 탄력적으로 할 수 있다.

바코드 시스템이 제 기능을 발휘하기 위해서는 무선주파수(Radio Frequency: RF) 터미널이 이용될 수 있다. RF 터미널은 포장, 수송 재고 관리 과정에서 정확한 취급이 이루어지는 것을 가능케 하고, 실시간으로 생산물이 포장될 때 과수원구획 번호, 농가번호 등을 실시간으로 식별하게 하는데 이러한 과정을 통해 정선된 품질의 상품만 포장이 된다.

오차 없는 발주를 위해서도 RF 터미널이 이용된다. 이 시스템은 발주품 목이 맞는지를 스캐닝을 통해서 일치하는지를 각각의 팔레트에 붙어 있는 인식표(tag)를 통해서 식별한다. 인식표는 바코드의 일련번호에 기억되어 있는데 산물(bulk)상태로 입고되는 과실의 빈(bin)상자, 가선별포장(pre-sized /pre-sorted)품, 즉시판매 비포장과실, 최종 포장과실에 각각 인식표가 부착된다.

이 시스템은 공동계산제의 운영을 가능케 하는데 기본적으로 공동계산방식은 농민이 포장단계에서 실제로 참여할 수 있고 가선별(presorting)상태에서도 참여할 수 있다.

과실을 인수하여 포장하는 형태는 다음 4가지로 구분될 수 있다.

첫째, 산물형태로 되어 즉시 포장하는 방식으로 이 경우 빈(bin)상자 인식표는 지우고 팔레트와 회수번호표를 부착한다.

둘째, 산물을 최종 포장 전에 가선별하는 방식으로 빈 상자 인식표와 가선별 인식표를 부착한다.

셋째, 가선별된 과실을 최종 포장하는 방식이 있다. 이 경우 팔레트 번호표를 붙이고 회수번호표는 지운다.

넷째, 사전에 포장된 과실을 재포장하는 경우로 이 경우 새로운 팔레트 인식표를 붙인다. 보다 정확히 공동계산제를 실행하기 위해서는 판매번호가 일관되게 일련번호로 추적될 수 있어야 하며, 선과장에서 팔레트와 빈상자의 인식표는 놀랜드 포클리프트 스캐너(Norand Forklift-mounted Scanner)를 이용해서 스캐닝 한다.

3) 공동계산모듈(Pool Accounting Module)

공동계산 정보시스템은 다음과 같은 형태로 운영된다. 첫째 배송후 귀속되지 않은 수입의 공동회계 입력, 둘째 공동계산 참여 특정농가에 직접적인 비용/수익 배당, 셋째 공동계산 그룹간 소득 이동이나 조정, 넷째 공동계산 중지 등으로 운영된다.

공동계산 정보시스템은 농가의 경영원장과 연결되어 있어 농가의 공동계산에 대한 모든 정보를 기초로 기능을 발휘하게 된다. 이밖에 공동계산 정보시스템은 배송(invoicing), 급여지급(payroll) 등과 관련한 정보시스템과도 연결되어 작동된다.

공동계산정보시스템은 개별고객의 여건에 따라 최대한 간편하게 사용할 수 있도록 고안되었는데 기본적으로 다음과 같은 정보를 담을 수 있다.

① 공동계산비용 부담, ② 포장비용, ③ 직접 거래/외상 거래, ④ 공동계산 회원관련정보, ⑤ 빈(bin) 상자 임대 및 임대료, ⑥ 생산농가의 과실수송비 부담, ⑦ 기타 공동계산관련 부담금 및 채권 등

본 시스템에서 다음 형태의 공동계산제 운영관련 비용을 설정할 수 있다.

첫째, 포장비용(packing charge)을 시스템에서 설계할 수 있다. 포장단가는 품목, 품종, 등급, 크기에 따라 다양한데, 공동계산에 참여하는 모든 농가의 동일한 포장품목에 대해서는 일률적으로 같은 포장요율이 적용된다.

둘째, 공동계산수수료(pool charge)를 설정할 수 있다. 이는 회비성격으로서 공동계산참여농가에게 출하물량, 포장물량 등에 근거하여 부과되는데 등급이나 크기는 영향을 주지 않는다.

셋째, 공동계산 시스템하에서 빈(bin)상자를 임차할 경우 이에 대한 수수료를 시스템에 설정할 수 있다. 이밖에 공동계산과 관련한 기타 수수료를 징수하는 프로그램을 시스템내에서 설정할 수 있다.

4) 작물수량 추정 프로그램

이 시스템은 최종 출하량을 미리 예측하기 위해 사전적으로 수확량을 추정하도록 고안된 프로그램으로 과원면적, 식재주수, 수확주수 등을 입력하면 자동으로 주당 수확량과 선과장에 도착할 총 과실량이 과원별로 지구별로 계산된다.

5) 농가 출하 빈(bin)상자 재고관리

이 시스템은 산물 상태로 출하된 과실이 선과라인에 투입되기 전 까지 추적이 가능하도록 고안되었다. 따라서 이 시스템을 통해 얼마나 많은 수의 산물운송 빈상자가 출하되었고 선과라인에 투입되었는지 알 수 있어 남아 있는 상자의 재고를 관리할 수 있다. 빈상자에 부착되어 있는 인식표를 통해 정보가 전산처리되고 인식표에는 출하농가, 출하원산지, 과실명, 품종, 출하일자 등이 기재되는데 창고별, 과실종류별, 농가/지역단위별, 공동계산 형태별로 빈상자 재고관리가 가능하다.

3. 농업관련산업 적용 소프트웨어: Kirkey 소프트웨어시스템

Kirkey 사는 미국내 농산물 생산, 포장, 판매, 가공 등 모든 시장참여자에게 비즈니스의 종합 전산운영을 위한 소프트웨어를 개발하였다. 이 소프트웨어는 IBM i시리즈(AS/400) 중급 컴퓨터에 맞게 고안 되었으며 프로그램은 농산물판매 관리, 선과장 관리, 생산물 가공, 일반경영관리에 적용할 수 있다.

가. 선과관리(선과장)

선과 및 포장과 관련한 관리프로그램은 다음과 같은 업무에 응용된다.

- ① 농가가 선과장에 과실을 출하하면 산물상태로 관리하는데 온라인으로 인식표를 전산화하고 관리한다. 과실입고는 출하주, 품종, 과원, 일자별로 관리된다.
- ② 빈(bin)상자는 농가별로 추적 관리되는데 “인식표프로그램(Receiving Ticket Program)” 에 의해 사실상 비어있던 생산물이 담겨 있는 상태이던 모두 추적관리가 가능하다.

- ③ 재고입출고관리의 전산화가 이루어지며, 저장고 셀 별로 재고관리를 할 수 있다. 선과프로그램은 팔레트의 출고시점에서부터 최종 판매에 이르기까지의 전과정 추적이 가능하며 예비선과(pre-sized)와 CA (controlled atmosphere)저장고의 관리운영에도 적용할 수 있다.
- ④ 팔레트 인식표에는 바(bar)코드가 입력되어 있는데 팔레트 인식표 포맷은 TL Ashford 바코딩 소프트웨어를 이용하여 만들 수 있다.
- ⑤ 과실을 선과장에서 통조림 공장으로 이동시킬 경우 추적이 가능하며, 통조림라인에서의 입출고 및 검사업무가 전산화된다. 통조림공장에서 농가별 수익배분도 자동으로 이루어진다.
- ⑥ 포장부자재에 대한 세부 재고관리가 이루어지며, 재료별 단가, 사용한 부자재 등도 전산화된다. 부자재를 다른 곳으로 이동시킬 경우 기록되며, 모든 부자재구입 영수증이 전산작업으로 제공된다.

나. 생산자 관리

생산자 적용모듈은 농가의 계약내용, 수수료부담, 수확량 추정, 비용정산 및 공동계산, 농가의 손익원장과 관련된 정보를 전산화한다.

- ① 농가별 과원규모, 농가의 기업(선과장 등)과 계약내용 및 조건 등이 포함된다.
- ② 농가별로 다양한 형태의 수수료를 정의하고, 필요에 따라 수수료를 통합 운영할 수 있다. 또한 작물 종류, 공동계산 유형, 계약형태에 따

라 수수료를 우선순위를 두고 지급할 수 있다.

- ③ 농가별, 지역별로 수확량을 추정할 수 있어 수확기에 농가가 직면한 상황을 수확량 추정을 통해 파악할 수 있다.
- ④ 공동계산과 정산업무를 전산화하여 농가별로 수입과 비용을 관리한다.
- ⑤ 농가의 월별 및 연간 수입명세서, 재무회계, 공동계산결과보고 등 농가의 경영전반에 대한 기록을 전산 관리한다.
- ⑥ 농가의 비료, 농약, 농기계 등 농자재 구입 대금, 외상거래 등이 정보화되며 농가가 라인으로 거래내역 및 계좌 잔고 등을 조회할 수 있다.
- ⑦ 프로그램을 이용하여 작물생산량 추정이 가능하며, 실제 수확기에는 3-4주동안 1일 또는 주 단위로 생산량, 출하량을 전산화하여 참고가 가능하다. 판매재고는 항상 업데이트되기 때문에 판매부서는 재고를 앞서서 관리 할 수 있다.

다. 가공공장

가공분야에서는 생산계획, 자재관리, 구매주문, 품질관리 등에 대한 정보시스템이 구축되다.

- ① 생산자재와 완제품 재고에 대한 정보가 입력되고, 2-3주전에 생산계획이 수립되며 비교목적으로 몇 가지 계획을 수립할 수 있다.

- ② 실제 조업일정을 특별한 생산계획에 적용할 수 있고, 사후에 계획조업량을 실제 조업량과 비교할 수 있다.
- ③ 온라인상에서 생산에 필요한 자재를 검토할 수 있으며, 주단위 또는 일단위로 자재소요량을 파악할 수 있고 실제 자재사용량은 표준사용기준에 의해서 명시될 수도 있다.
- ④ 자재 재고는 생산이 시작되면 온라인에서 실시간 파악이 가능하며, 자재구매제안서가 생산에 필요한 소요량 산출에 근거하여 작성되어 알려진다.
- ⑤ 시스템내에서 생산비용원장이 기록되며, 자재가 효율적으로 관리되고 있는가를 판단 할 수 있는 보고서도 볼 수 있다.
- ⑥ 본 시스템내의 구매주문시스템은 완제품, 소요자재, 기타제품의 구매 상황을 추적하며, 이를 기록한다.
- ⑦ 복수지 배달이나 배달 일자 변경 등의 상황변화를 온라인상에서 조절 가능하다.

라. 운송분야

본 시스템은 운송관리, 수송, 수출과 관련한 데이터의 정보화 작업이다.

- ① 화물 운송지역을 무한정 관리할 수 있으며, 운송시에 팔레트에 부착되어 있는 인식표를 스캐닝하여 발송정보를 관리한다.

② 운송시 컨테이너의 온도를 기록하고 보고한다.

③ 상하차 등 화물요금을 입력하는 등 화물요금관리를 전산화 한다.

마. 판매관리

판매관리관련 소프트웨어는 판매주문관리, 전산자료, 과거거래기록관리, 판매재고관리, 송장관리, 신용조회 등과 관련한 자료를 정보화 한다.

① 조회주문(inquiry orders), 잠정주문(tentative orders), 확정주문(booked orders)과 관련한 정보를 관리한다.

② 구매주문 정보, 송장관리, 가격인상, 특별주문 등에 관련한 정보처리가 가능하고, 판매가격 조정 프로그램도 설치된다.

③ 온라인으로 일자별 상품구매를 문의할 수 있으며, 고객의 상품별 구매에 관한 과거내역을 조회할 수 있다.

④ 일일판매 및 선과장 판매보고서, 가격변동내역, FOB 가격내역 등을 체크할 수 있다.

⑤ 선과가능물량 추정, 재고량, 예약판매량, 가격대별 주문량 등 재고분석이 가능하다.

⑥ 送狀의 수정, 직불(debit)/신용(credit) 정보를 처리한다.

⑦ 고객별, 상품별로 과거 구매내역을 조회할 수 있다.

- ⑧ 소비자의 반품, 교환 등과 관련한 요구가 있을 때 이를 신속히 처리할 수 있는 시스템을 구축한다.

4. 물류창고 관리 소프트웨어 : ATS(applied Technical Systems)

가. 개황

ATS는 물류창고의 효율적인 운영을 위해 고안된 소프트웨어로서 윈도우 NT 서버를 사용하여 작동하도록 설계되었으며, 이용고객은 윈도우 NT에서 작동하는 SQL 서버 또는 윈도우 NT나 UNIX에서 작동하는 Oracle을 데이터베이스로 활용할 수 있다.

물류창고는 보다 많은 상품을 보다 정확히 그리고 많이 조합해서 확보하는 것이 매우 중요하다.

소비자는 당일내에 배달, 포장, 운반되길 원하며 이러한 서비스에 매우 민감하게 반응한다. ERP(Enterprise Resource Planning)/호스트 시스템은 소비자의 구매에 적절히 대응하기 위해 정확한 재고량 파악, 창고위치 및 재고상태 등을 알 수 있도록 고안되어 있으며, 창고관리의 구체적인 목표는 주문량 증가, 운송의 정확성 제고, 노동생산성 향상, 공간이용의 극대화, 창고관립의 최소화에 있다.

나. ATS의 효과

물류창고에 ATS를 도입함으로써 다음과 같은 창고관리의 효율성을 기대할 수 있다.

- ① 실시간으로 재고관련정보를 직접 접할 수 있다.
- ② ERP 시스템과 상호 정보교환이 가능하다.
- ③ 바코드에 다양한 정보를 담을 수 있다.
- ④ 높은 정확성을 갖고 RF 터미널을 이용하여 자동데이터(automated data)를 수집할 수 있다.
- ⑤ 복잡한 운반, 포장 등의 물류기능을 관리할 수 있다.
- ⑥ 아무리 큰 물류창고도 관리가 가능하다.

다. ATS의 적용 분야

ATS는 다음과 같은 물류창고 관리업무에 적용될 수 있다.

- ① 상품의 입고, ② 품질관리, ③ 재고 추적, ④ 재고관리, ⑤ 주문관리, ⑥ 운송, ⑦ 하역관리, ⑧ 실시간 거래관리, ⑨ UCC 바코드 표준화 지원, ⑩ EDI 지원, ⑪ 2곳 이상의 물류창고 관리

5. 와이오밍주 선과장 운영 프로그램(VEGMARC)

VEGMARC는 농가의 출하기록, 고객의 구매기록 및 공동계산가격의 계산 등에 필요한 컴퓨터 프로그램으로서 미국내 협동조합 선과장이 이 프로

그램을 많이 이용하고 있다. 주요메뉴는 다음과 같다.

- ① 고객관리
- ② 일별 농가 출하자료
- ③ 일별 고객 구매자료
- ④ 일별 거래보고서
- ⑤ 공동계산 정산서

주요설치메뉴는 취급품목, 선과장사용료 등인데 이들은 연초에 설치되므로 거의 변경시킬 필요가 없다. 일일거래보고서는 생산자, 구매자, 선과장, 판매에 관한 4가지 내용을 제공하는데 사용자재에 대한 정보도 제공한다.

공동계산정산은 협동조합의 공동계산기간 동안의 가격정산이 주요사항이다. 이 프로그램을 사용한 85%의 협동조합이 본 프로그램을 매우 긍정적으로 평가한 것으로 조사되었다. 현재는 다른 협동조합과 프로그램 호환성을 높이는 것이 당면과제로 지적되고 있다. 미국 USDA/AMS의 시장정보 네트워크가 협동조합의 본 프로그램운영에 도움을 주고 있음.

<부록 1> 사과 선과방식

사과는 직접선과(direct pack)와 예비선과(presize system)의 두가지 방식으로 선과된다 직접선과방식은 산물산태로 빈(bin)상자에 담긴 사과를 선과라인에 부어 크기, 무게 등으로 선별하고 박스 포장하여 즉시 시장에 출하하는 형태이다.

예비선과방식은 직접선과방식과 같은 절차로 1차 선과하여 산물상태로 빈(bin)상자에 보관되었다가 나중에 포장하는 2단계로 나누어져 있다.

예비선과의 1차선과 절차는 다음과 같다.

- ① 사과는 빈상자로부터 꺼내 물 속에 담겨지고 물의 흐름을 이용하여 다음 단계로 이동된다.
- ② 작은 사과는 체인을 통과하면서 밀으로 걸러지고 수집상자에 담기며 남아 있는 사과는 씻겨지고 선과 테이블로 이동한다.
- ③ 선과테이블에서 숙달된 노동자가 결함 있는 사과를 골라낸 후 나머지 사과는 물 속으로 다시 투입된다.
- ④ 투입된 사과는 전자선별시스템(electronic eye)으로 이동되어 색, 크기, 중량에 따라 선과 된다. 같은 특성의 사과는 분류되어 컵이나 홀더에 놓인 후 홈통에 어느 정도 모이면 별도의 빈(bin)상자에 담긴다.
- ⑤ 빈상자에는 같은 크기, 같은 무게의 사과만이 담기게 되며 포장되기 전까지 저장고에 보관된다.
- ⑥ 포장을 위해서 저장고에 있는 사과는 다시 물속에 투입되어 컨베이어 벨트로 이동하여 세척과 헹굼 과정을 거친 다음 스폰지 롤러를 통과한 후 표면의 수분 건조를 위해 선풍기를 지나간다.
- ⑦ 사과에 왁스가 칠해지고 더운 공기건조기를 지나간다.
- ⑧ 사과에 상표 스티커 등의 부착되고 최종적으로 검사관이 결점 있는 사과가 있는지 체크한다.
- ⑨ 쟁반에 사과가 자동으로 놓인 후 박스에 들어가고 무게를 검사한다. 이과정에서 내부 품질관리기사나 주/연방품질관리관이 직접 사과를 검사한다.
- ⑩ 사과박스는 팔레트화되어 냉장고에 저장되어 출하될 때까지 보관된다.

직접선과시스템은 선과된 사과가 빈(bin)상자에 담겨 저장고(6번 이하 단계)를 거치지 않고 포장되어 출하하는 방식인데 다양한 형태의 크기와 등급으로 동시에 포장됨. 반면에 예비선과방식은 한가지 종류의 크기나 등급으로 포장된다.

제 5 절 모바일시스템의 개념 및 추진전략

1. 무선인터넷 및 유비쿼터스의 개념

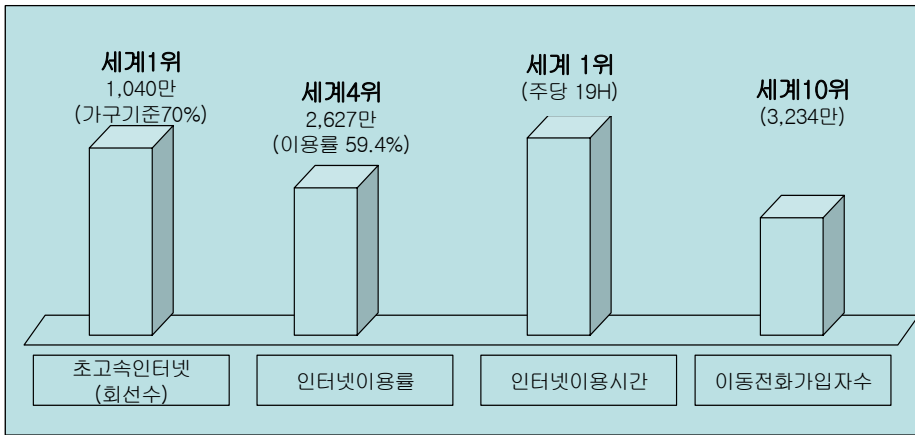
가. 무선인터넷의 개념 및 특징

1) 무선인터넷의 개념

무선인터넷서비스는 무선LAN 이나 B-WLL 등 무선을 통해 인터넷에 접속하는 것을 포함하지만, 일반적으로 휴대형 단말기 및 PDA 등 휴대형 기기를 통해 무선으로 인터넷에 접속하여 데이터 통신이나 인터넷 서비스를 이용하는 것을 말하며 사용자가 이동 중에 무선네트워크(Wireless Network)를 통해 인터넷 서비스를 제공받을 수 있는 환경과 기술을 말한다. 무선인터넷으로 이동전화나 PDA로 언제 어디서나 인터넷에 접속하여 다양한 정보검색과 전자상거래, 커뮤니케이션을 하는 것으로 기존의 인터넷 환경의 여러 제약을 극복할 수 있다.

'03년 8월을 기준으로 우리나라의 인터넷보급율과 사용율은 매우 높으며, 무선인터넷 및 모바일 분야에서 큰 기대를 갖고 있다.

<그림 2-4> 인터넷 및 이동전화가입자 현황



우리나라의 무선인터넷 서비스를 제공하는 업체는 SK텔레콤, KTF, LG텔레콤으로 '03년12말 약 3300만의 무선인터넷 가입자현황이 집계되었으며 점유율 면에서 SK가 50%, KTF가 35%, LG텔레콤이 15% 로 나타났다. 하지만 현재 번호이동성제도의 시행에 따른 변동이 일어나고 있으며 각 서비스사의 모바일 서비스 제공에 따른 업체의 점유율에 큰 영향을 미칠 것으로 전망된다.

e-CRM, e-biz, e-commerce 등의 인터넷 혁명에 이어 이동통신의 폭발적인 확산과 무선인터넷을 기반으로 한 상거래, 경영관리 시스템 등이 부상하면서 새로운 경쟁 패러다임 창출과 모바일을 기초로 하는 M-biz, M-commerce 등의 신조어 등이 생겨나고 있다.

M-biz는 인프라와 기기, 애플리케이션과 고객경험, 고객관계와 공급체인 등에 구조의 변화와 유통혁신, 신규 비즈니스 모델 창출을 이끌어 가고 있고 혁신적 플랫폼, 애플리케이션 인프라, 모바일 포털, 고객관계, 공급체인, 오퍼레이션, 유통 시스템 등의 분야에서 기회가 존재한다.

2) 무선인터넷 서비스의 특징

첫째, 위치기반 정보 서비스이다. 앞으로는 GPS(Global Positioning System)을 이용한 사용자의 현재 위치에 따른 지리 정보 서비스와 같은 사용자의 위치 정보와 연관된 서비스가 중요한 비중을 차지할 것이다.

둘째, 신속한 정보의 취득이다. 기존의 매체는 정보 공급자 위주의 정보 제공의 한계를 갖는 반면, 무선인터넷 환경에서는 이동 중에서도 능동적인 정보 접근이 가능하여 정보에 관한 푸쉬 방식으로 제공 받을 수 있어 주식 투자, 뉴스 속보 등 신속한 정보취득이 가능하다.

셋째, 모바일 오피스(Mobile office)이다. 전통적 의미의 사무실을 벗어나 언제 어디서나 회사 업무를 처리할 수 있는 환경으로 업무의 범위를 확대하여 모빌 오피스환경은 기업 경쟁력 제고를 위한 새로운 기업 문화로 확산되고 있다.

나. 유비쿼터스의 개념 및 구현

1) 유비쿼터스의 개념

유비쿼터스(Ubiquitous)란 ‘도처에 널려 있다’, ‘언제 어디서나 동시에 존재한다’라는 의미로 일반적으로 물과 공기처럼 도처에 편재한 자연자원 또는 종교적 의미로 신이 언제 어디서나 시공을 초월해 존재한다는 것을 상징할 때 사용하며, 유비쿼터스 컴퓨팅(Ubiquitous Computing)은 88년에 제록스(XEROX)사의 마크 와이저(Mark Weiser)가 주장하였다.

유비쿼터스 컴퓨팅이란 다양한 종류의 컴퓨터가 사람·사물·환경 속으로

스며들고(Embedded) 서로 연결되어, 언제 어디서나 쉽게 이용할 수 있는 최적의 컴퓨팅 환경을 의미한다. 아울러 유비쿼터스 컴퓨팅 및 네트워크 서비스는 논리적 인지가 가능한 스마트(smart) 공간을 기반으로 사물과 컴퓨터, 그리고 사람이 상호 연계되어 언제 어디서나 컴퓨터에 접속할 수 있는 환경을 의미한다.

유비쿼터스 컴퓨팅은 컴퓨터의 발전과정에서 나온 개념으로 인간중심을 지향하는 데서 출발하였으며, 사용자 보조수단으로서 기계중심(Computer-centric) 컴퓨팅의 문제점을 극복하기 위한 대안으로 인간중심(Human-centric)의 쉽고 편리한 도구로서 유비쿼터스 컴퓨팅이 제기되었다.

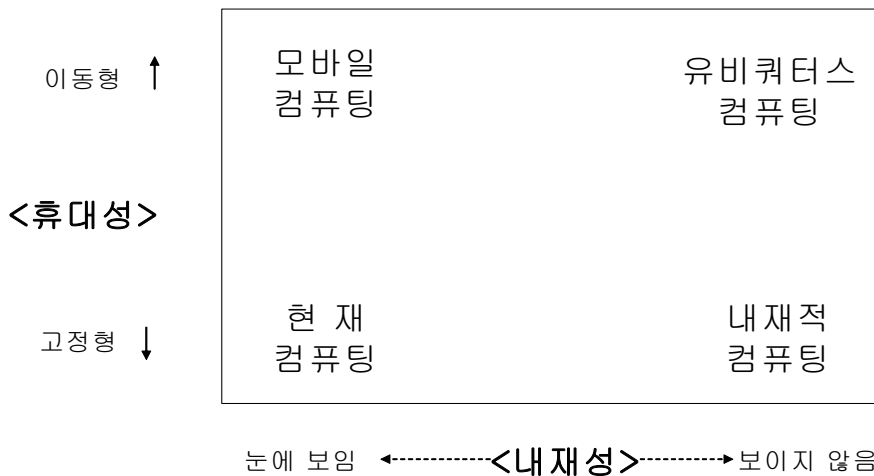
<표 2-28> 컴퓨팅 개념의 진화 단계

구 분	대형컴퓨터 시대	PC의 시대	유비쿼터스 시대
시 기	1980년대 이전	1990년대	2000년대 이후
컴퓨터	메인프레임 (클라이언트-서버)	+ PC	+ 지능화된 사물 (Intelligent Object)
활용 및 대응관계	대형의 고가 컴퓨터 여러 이용자가 공유 (Many Persons One Computer)	1인 1대 사용 (One Person One Computer)	다양한 컴퓨터가 도처에 편재, 사용자가 의식하지 않음 (One Person Many Computer)

유비쿼터스 컴퓨팅이 추구하는 것은 첫째, 장소에 구애받지 않아야 하고 (이동성) 둘째, 자연스럽게 존재해야 하며(내재성) 셋째, 스스로 판단할 수 있는 자율성을 가져야 함(지능성)으로 인간이 컴퓨터를 위해 센싱(인식, 감지) 및 인터페이스(상호작용) 기능을 제공하는 것이 아니라 컴퓨터가 스스로 정보를 센싱하고 사용자에게 맞게 인터페이스를 제공하는 것이다.

유비쿼터스 구현은 크게 컴퓨팅 기능의 내재성(Embedded) 강화와 휴대성(Mobility) 제고 방향으로 추진하고 있으며, 완전한 유비쿼터스 컴퓨팅은 내재성과 휴대성에 지능성까지 가미한 형태로 ‘언제 어디서나, 자연스러운 컴퓨팅’을 구현하는 것이다.

<그림 2-5> 유비쿼터스 컴퓨팅의 구현 방향



유비쿼터스 컴퓨팅의 구체적 구현을 위해서는 핵심 기술을 활용한 시스템, 네트워크, 애플리케이션(응용기술), 어플라이언스(응용제품), 플랫폼(기반환경) 등 다양한 기술개발이 요구되며, 유비쿼터스 사회는 현실(물리)공간과 가상(전자)공간의 경계가 사라지고 가상공간이 네트워크를 통해 생활공간으로 편입되어 새로운 제3의 가치공간을 창출하고 있다.

2) RFID 태그에 의한 유비쿼터스 컴퓨팅의 구현

신개념 바코드인 RFID 태그 기술이 유비쿼터스 컴퓨팅을 구체화하고 있으며, RFID 기술의 발전에 따라 컴퓨터의 사물에 대한 내재화가 급속히

진전되고 있다. 기본적인 RFID 시스템은 안테나, 리더기(판독장치), 태그, 주컴퓨터 등 기술적인 요소로 구성된다.

RFID 시스템은 '03년을 보급 원년으로 '04년부터 확산될 전망으로 '03년에 갭(GAP), 피앤지(P&G), 질레트(Gillete), 메트로(Metro) 등에서 도입하였고, 월마트(Wal-Mart)에서도 '05년 도입을 위해 테스트 중이며, 일본은 식품의 유통경로를 추적할 수 있는 RFID 시스템을 도입, 미국 국방성은 '05년까지 주요 조달품에 도입 예정이다. 국내에서도 CJ GLS, 삼성테스코 등이 시범 프로젝트를 진행 중이다.

3) RFID의 응용 분야

첫째, 옷감에 대한 정보를 가진 RFID칩을 내장한 세탁기가 최적 세탁을 제공하고, 냉장고가 음식물의 유통기한을 자동으로 체크하며, 가정 내 보안 센서와 연결하여 홈 네트워크를 구성하는 등 전자제품 분야에 응용되어 생활의 편의를 제공하고 있다.

둘째, 쇼핑 후 카트가 계산대를 통과하는 순간 일괄 계산, 효율적 재고관리와 자동적인 재고관리, 도서관의 장서관리 등 유통 및 서비스 분야에 응용되어 생산성 혁신 및 판매증진을 도모하고 있다.

셋째, 자동차 엔진에 칩을 내장하여 차량별 옵션 구분에 활용과 식료품 유통기한, 원산지, 생산자 확인과 타이어 안전상의 문제여부 체크(공기압, 마모정도 등)와 병원의 각종 의료장비 추적, 가축식별, 교통카드 등 기타 부가정보를 제공하고 있다.

4) 생활 속의 유비쿼터스 모바일 서비스

RFID와 더불어 휴대폰과 PDA(Personal Digital Assistants : 휴대용 개인정보 단말기)가 u컴퓨팅의 총아로 등장하여 무선통신 기술의 발전과 더불어 컴퓨터의 이동성을 극대화시키고 있으며, PDA는 자체 컴퓨팅이 가능하므로 차세대 PC로서 각광을 받고 있다.

휴대폰 또는 PDA에 의한 응용서비스에는 텔레매틱스(Telematiks), 원격 제어, 위치확인시스템(GPS), 모바일 방송, 영상전화 등 첨단 서비스와 차량 항법장치(Navigation)를 장착해 사용하는 모바일 교통안내 서비스, 휴대폰을 이용한 홈 네트워크의 무선원격제어 서비스, 위성디지털미디어방송(DMB)을 통한 영상, 음성, 데이터 등 다양한 콘텐츠와 무선인터넷 제공서비스, 휴대폰에 의한 쇼핑·결제·금융거래 등 모바일 커머스로 이동통신사와 금융기관의 제휴에 의한 모바일 인터넷 뱅킹, 통장번호 대신 휴대폰 번호를 이용한 송금, 공과금 납부, 물품대금 결제, 티켓예매, 현금카드 및 교통카드 기능 등이 있다.

2. 모바일 컴퓨팅의 개념 및 등장배경

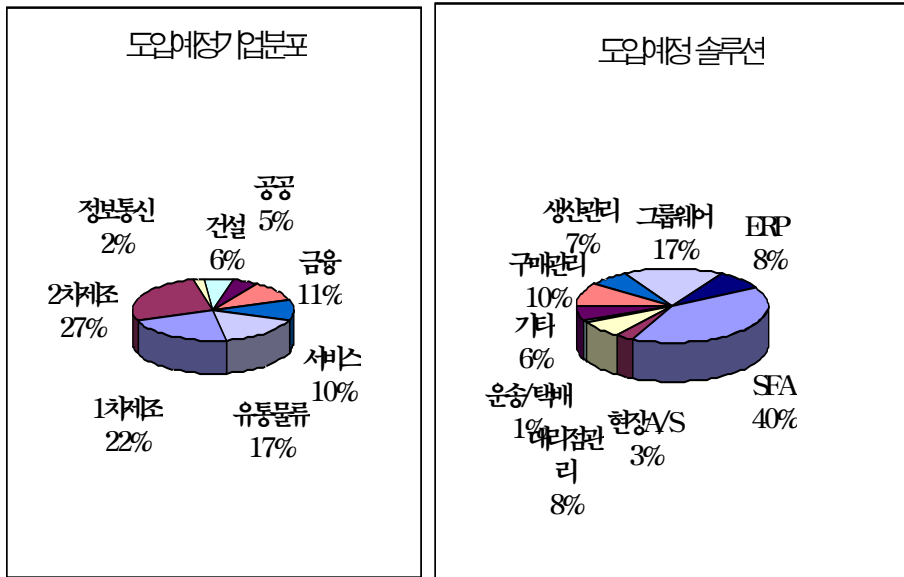
가. 모바일 컴퓨팅의 개념

모바일 컴퓨팅이란 기존 인터넷과 e비즈니스에 무선통신이 결합된 컴퓨팅 방법으로 이동형 단말기를 통한 업무 전산화를 통칭하며 기존 업무 프로세스 중 원격지의 업무가 자원 및 효율성 측면에서 비효율적으로 운영되는 것을 데이터 및 운영 인력의 효과적인 관리 체계를 구축하기 위한 기업 정보화 기술을 말한다.

모바일 컴퓨팅을 기업에 접목시키기 위해서 모바일 관련 프로젝트를 발주했거나 발주 예정인 기업이 국내 매출규모 1000대 기업 중 27%에 달하

며 실제적인 구현을 위한 움직임이 활발하며, 금융권의 모바일 banking 솔루션도 초보적인 모습에서 벗어나 유무선 통합 포털 시스템으로의 다양한 서비스 진화를 설계중이다.

<그림 2-6> 모바일 컴퓨팅의 도입 예정기업 분포 및 예정 솔루션



자료 : KRG ,2002

현재 일반 산업분야에서 업무생산성 향상과 비용절감을 효과적으로 이루어내기 위해서 모바일 인프라로 전환을 서두르고 있으며, 현장근로자가 직접회사 데이터베이스에 접근할 필요성이 증가하면서 모바일 컴퓨팅 사무환경에 대한 수요가 급증하고 있다.

모바일 컴퓨팅은 최첨단 통신기술과 자동차기술과의 결합인 텔레매틱스, 새로운 개념의 모바일 게임, PC근방 10m 내에서 활용할 수 있는 무선인터넷 페이스인 블루투스, 전자상거래 등의 분야를 통해 이용될 수 있으며, 현재 유비쿼터스 혁명과 더불어 유통업체등 많은 기업들이 투자를 준비하고 있

다.

M-Commerce는 상품의 구매와 대금의 지급 결제에 이동통신 단말과 망을 이용하는 것으로 유선 인터넷을 이용한 전자상거래를 무선 인터넷까지 확장한 형태로 이해할 수 있으며, 핸드폰, PDA 및 노트북 컴퓨터등과 같은 이동단말과 이동통신망상에서 이루어지는 모든 형태의 가치전달 활동을 의미하며, 현재까지의 추세를 보면 m -커머스 시장은 미국보다는 유럽시장을 중심으로 빠르게 확산되고 있고, 특히 핀란드, 노르웨이, 스웨덴 등 스칸디나비아 3국은 산악 국가들로 전국을 유선으로 연결하기 어려워 자연스럽게 m -커머스가 발전하게 된 대표적 국가이며 이동전화기로 자판기안의 커피나 청량음료를 사서 마실 정도로 무선인터넷 시장이 발전되어 가고 있다.

경제 환경의 변화가 급격히 진행됨에 따라 미국의 보다폰 에어터치, 독일의 만네스만, 일본의 NTT 도코모 등 세계적인 이동전화회사를 비롯 노키아 모토로라 등 이동전화 단말기업체, 마이크로소프트를 비롯한 소프트웨어 업체들이 무한 생존경쟁을 벌이고 있다. 하지만 우리나라 모바일시장은 세계1위 수준으로 모바일 비즈니스의 발전가능성이 큰 국가로 평가받고 있어 세계 모바일 컴퓨팅 산업의 선두국가로 도약할 수 있을 것으로 예상된다.

<표 2-29> M-commerce가 확산될 수 있는 동기 요인

M-commerce 확산
□ 보급된 무선 단말기수가 PC보다 많을 뿐 아니라 보다 빠르게 성장하고 있음
□ PC보다 무선단말기가 더 친숙하고 휴대하는 경우가 많음
□ PC나 TV의 한계에 비해 무선 단말기를 통한 M-commerce는 언제 어디서나 서비스가 가능

- 언제 어디서나 접속이 가능한 점, 빠른 성장성과 불확실하긴 하지만 잠재적으로는 큰 수익성 때문에 많은 사업자들이 골드러쉬를 찾아 몰려들고 있으며 이들간에 합종연횡을 통한 Win-Win 게임이 가능

나. 모바일 컴퓨팅의 등장 배경

정보화 사회에 들어선 후 가장 비약적인 발전을 이룬 부분은 바로 인터넷과 모바일 통신이다. 이 두 가지는 정보화 사회의 기반을 이루는 부분으로 점차 많은 정보의 교류가 요구되고, 정보의 이용이 제한된 공간에서만 국한되는 것이 아니라 언제 어디서나 정보의 이용이 필요해 졌기 때문이다. 이러한 광대한 정보의 보고인 인터넷과 이동성을 확보한 무선통신은 이러한 요구에 의하여 융합하게 되고, 따라서 현대인의 생활 패턴을 크게 바꾸어 놓았다.

아울러 1995년 마이크로소프트사의 빌게이츠는 ‘정보는 모든 사람의 손끝에’를 말하며 PC에서 전자지갑, 광대역 쌍방향 무선통신 등 정보사회가 도래할 것임을 예측. 앞으로는 PC가 아닌 휴대폰이 인터넷 단말기 시장을 이끌 것으로 전망하였다.

호출기로부터 시작하여 셀룰러폰, 개인휴대전화기(PCS)를 거쳐 CDMA2000, IMT2000 기반의 이동통신까지 최근 몇 년 사이에 이동통신 환경은 엄청난 속도로 발전하여 현재 휴대폰은 음성통화뿐만 아니라 문자, 사진, 동영상 등의 영역까지 포함하는 종합적인 커뮤니케이션 미디어로의 발전을 계속하고 있고, 단말기의 발전과 함께 단말기를 통해 가능한 부가서비스의 개발도 활발히 이뤄지고 있으며, 이 같은 무선 인터넷의 활성화는 mCRM, M-commerce, B2B 등의 경영 전반적인 분야로 확장 되고 있다.

아울러 모바일 컴퓨팅을 통해 인터넷에 접속되어 있는 가전제품 및 장비를 무선 단말기를 통해 언제든지 상태를 점검하고 조정할 수 있게 되었고, 네트워크, 단말 플랫폼, 컬러 핸드폰 출시가 맞물려 진정한 멀티미디어 서비스가 가능해지고 있다.

또한 이동 및 활동 중 정보수집 및 검색, 전자메일 송수신, 티켓예매 등의 필요성이 늘어나고 있으며, 커뮤니케이션의 신속성과 다양성을 요구하고 이동사업자간 경쟁 환경에서 새로운 수익모델을 창출하고자 하는 기업의 성장, 발전 논리 확산을 통해 모바일 컴퓨팅의 중요성의 대두되고 있다.

그러나 유선 인터넷에서 이용되는 기술들은 무선 단말기에 그대로 적용되기엔 많은 문제가 있다. 따라서 일반 데스크탑과 달리 저용량의 프로세서와 메모리를 사용, 무선통신 단말기에 맞는 프로토콜이나 기술들이 요구된다.

<표 2-30> 모바일 컴퓨팅 환경 도래

- | |
|---|
| <ol style="list-style-type: none"> 1. 시장의 확대 2. 경쟁의 심화 3. 시장의 양적 성장에 따른 구조적 변화 필요 4. 통신환경의 급속한 발달로 인한 경쟁력 강화의 필요성 5. 고객 욕구의 다양한 변화 6. CRM 시대의 도래(e-CRM탄생) 7. 처리과정의 정확, 신속, 투명화 필요(업무처리의 효율성) |
|---|

3. 모바일 컴퓨팅의 효과 및 문제점

가. 모바일 컴퓨팅의 효과

개방된 인터넷망을 이용하여 새로운 거래처 발굴은 물론 구매절차의 모바일화로 납기단축, 구매단가, 인건비절감 등 구매비용의 절감의 효과를 이룰 수 있다. 그리고 판매, 생산 계획의 사내 공유로 재고를 최소화하고 재고회전을 증가로 재고 관련 비용을 절감할 수 있다.

아울러 정보의 실시간 공유로 부품개발, 유통 등의 시간단축과 신제품 개발기간을 단축시켜 주며, 신속한 정보 전달과 제품 구매 소요시간을 단축함은 물론 적시, 적량의 공급시스템을 구축함으로써 효율적인 물류체계를 관리할 수 있다. 또한 모바일 마케팅을 통한 비용절감, 고객과의 열린마당을 제공함으로써 문의사항 및 서비스개선에 대해 만족도를 높이고 인터넷을 통해 새로운 시장을 개척함은 물론 고객에 개개인의 성향에 맞게 맞춤형 주문제작을 가능케 해준다.

<그림 2-7> 모바일시스템의 효과



나. 현재의 모바일 컴퓨팅의 과제(문제점)

비즈니스에 모바일을 도입한 기업들은 속도 및 안정성의 문제로 가장 큰

곤란을 겪고 있으며 구축비용, 보안, 어플리케이션 부족 등의 운영상 문제를 지적하고 있다. 모바일 컴퓨팅이 어떤 사업분야에서 주수단으로 사용되기 보다는 보조수단으로서의 무선 시스템으로 전략하는 경우도 있기 때문에 모바일 컴퓨팅 도입 시 비용부담, 유지보수 및 운영의 부담 등 다양한 문제점을 내포하고 있다. 따라서 모바일 컴퓨팅의 도입 전에 반드시 투자 대비 효과에 대한 검증이 필요하다.

<표 2-31> 모바일시스템 도입과정의 장애요인과 해결방안

구분	장애 요인	해결 방안
모바일 시스템 도입	<ul style="list-style-type: none"> - 새로운 변화에 대한 오해 - 거래상대와 관계 변화 우려 - 투자비용의 회수 및 효과 우려 	<ul style="list-style-type: none"> - 경영책임자의 정보화 마인드 표명과 교육 실시 - 폐쇄적환경의 정보공유 실행으로 상호신뢰 관계증진 설득 - 기존의 PC 활용비용에 추가적 비용 - 장기적인 비용 및 효과 강조
모바일 시스템 실행	<ul style="list-style-type: none"> - 부서간 및 유통주체간 비협조 - 모바일의 기술적 복잡성 우려 	<ul style="list-style-type: none"> - 초기단계부터 관련 담당자의 합동 교육 - 전문업체의 기술지원, 전문교육 과정 이수
모바일 시스템 통제	<ul style="list-style-type: none"> - 업무의 통제 우려 - 효과에 대한 의구심 - 데이터의 안정성 및 보완 	<ul style="list-style-type: none"> - 사용자 ID, 비밀번호 부여로 업무처리 통제가능법적 근거 마련 - 비밀번호 관리, 데이터의 암호화 인증 기능

아울러 콘텐츠를 주고받는 무선인터넷방식의 표준이 통일되어 있지 않아 보안 뿐 아니라 각종 솔루션 상에서 자유로운 콘텐츠 제공이 어렵다.

또한 단말기 공급업체와 어플리케이션 공급자들은 이용자의 기대를 충족시키기 위해 기술적 투자 등의 자원을 투자 중이지만 이용자의 요구기능과

가격수준을 충족시키기에 개발 및 상용화 기간이 소요되는 문제가 있다.

4. 모바일시스템의 적용사례

모바일시스템은 업무의 효율성, 종업원 생산성 및 사기, 고객 서비스 및 만족에 영향을 미치는 주요요인으로 효율적인 의사결정을 위한 정보 제공의 원천이며 시장에 경쟁 우위를 지니는 제품 및 서비스의 개발 및 제공능력을 뒷받침하는 주요소이다.

무선PDA는 CDMA(코드분할다중접속) 및 무선랜 모듈을 본체에 내장해 위치정보, 교통정보, 모바일뱅킹, 증권거래 등을 수행할 수 있는 것은 물론, 물류·유통·고객관리 등 다양한 분야로 적용이 가능하다.

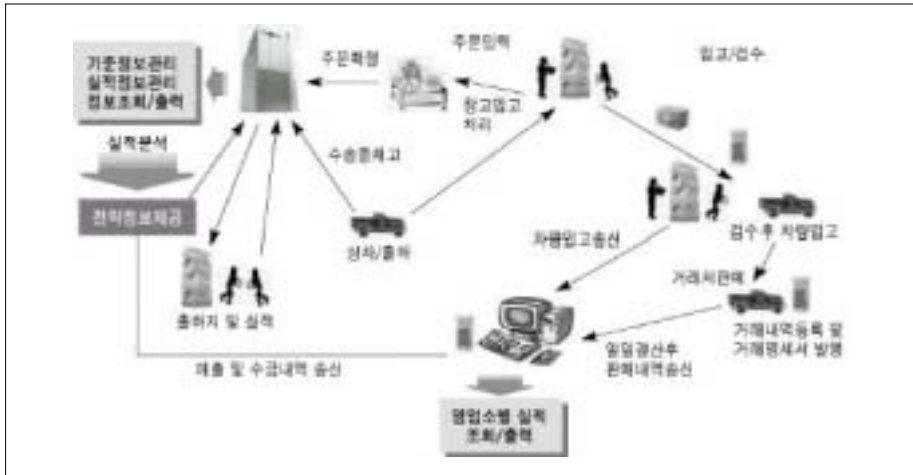
가. 모바일 영업지원시스템

모바일 영업지원시스템은 식음료 업체와 보험업계를 중심으로 활용이 증가하고 있으며, 무선 통신 모듈을 사용 입출고관리, 재고관리, 생산관리 등이 가능해 짐으로 물류시스템의 불합리성을 배제하고 합리적인 시스템을 구축하여 생산관리 및 판매조직으로의 합리적 운용이 이루어진다.

동양제과는 영업소, 공장, 물류창고를 하나의 시스템으로 연결한 물류창고를 구축 물류의 흐름에서 발생하는 입고, 재고, 출고, 판매 등 생성되는 데이터를 실시간 수집, 분석하고 있다.

이 밖에도 실시간 차량 추적을 이용한 물류차량 관제와 출·도착 확인 서비스 및 화물 이동경로의 고객서비스의 지원이 가능하고 화물 배송시간의 예측을 통한 설치 및 관련 서비스의 향상을 기대할 수 있다.

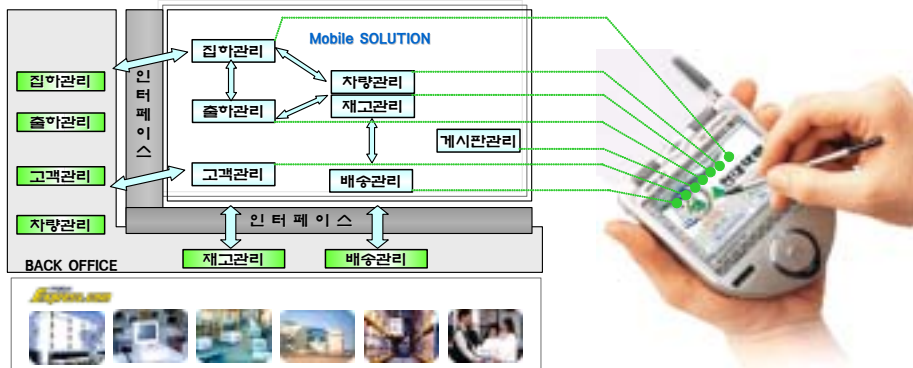
<그림 2-8> 모바일 영업지원시스템 구축사례



자료 : Mobile Business, 2001. 7

현재 대부분의 택배 운송 업체들을 비롯한 대부분의 사업에서 시스템을 도입하였거나 도입의 의지를 가지고 있으며 시스템을 구축한 많은 업체들의 생산 집계에서부터 제품 출하에 이르기까지 모바일 컴퓨팅을 구축하여 기존업무의 효율성을 제고하고 있다.

<그림 2-9> 택배사 모바일 정보 시스템 사용 사례



롯데제과 모바일 시스템은 식음료품의 입출고, 재고, 생산관리에 이르기

까지 하나의 시스템으로 연결하여 입고, 재고, 출고, 판매 등 생성 데이터를 실시간 수집, 분석하여 영업에 필요한 정보로 변경할 수 있다. 또한 차세대 무선 CDMA 서비스 및 위치기반 서비스를 위한 모바일 인프라를 구축하고, 바코드 처리 업무, 거래명세서 출력 업무 등에 활용하여 업무처리 간소화, ERP 자료 신뢰성 보장 등 업무 생산성측면에서 큰 향상을 보이고 있다.

<표 2-32> 롯데제과 모바일 시스템의 효과

구분	롯데제과 기존방식	신규방식(모바일 시스템)
동시에 여러 프로그램 구동기능	하나의 프로그 램수행도중 다른프로그램 동시수행불가	윈도우의 편리한 사용자 인터페이스(GUI) 방식으로 업무 메뉴간 이동이 편리
무선통신방법	불가능	향후 실시간 정보 분석가능 (CDMA, 무선LAN, 블루투스, 적외선 포트 활용 등 다양한 통신 방식 지원)
다양한 주변기기 활용업무 추가 기능	불가능	가능 -바코드(상품 및 물품박스) -향후 신용카드지불(매출 및 입금처리) -GPS, 메모리 카드 부착 및 프로그램 가능

<그림 2-10> 롯데제과 모바일 시스템 구축 사례



<그림 2-11> 롯데제과 PDA 활용사례

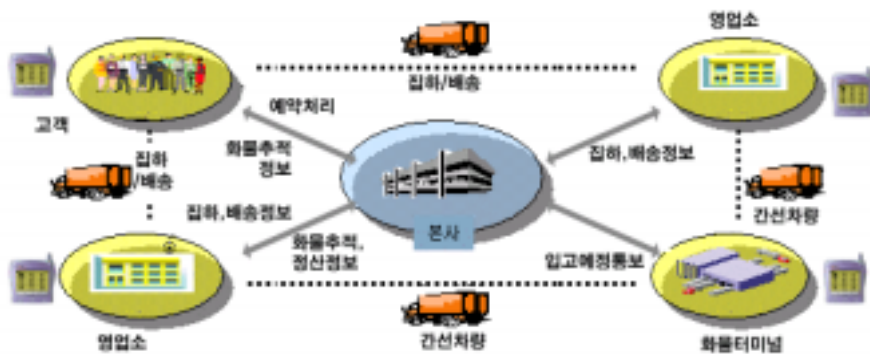


농심의 물류정보는 NICE(신정보시스템)라는 기간계 물류시스템을 통해 소통되며 MRP나 NAVL(차량위치추적시스템), 근거리수송시스템, 자동재고보충시스템, 배송계획시스템 등이 밀받침 되어 이를 통해 상세 스케줄링이나 공급망 최적화, 납기약속, 판매계획 수립 등을 추진할 수 있다. 농심은 더 나아가 물류센터관리를 비롯, 화물추적관리, 수배송현황, 물류비관리, 냉동물류관리, 등 각각의 물류정보시스템 기능을 통합화를 추진하고 있다.

모바일 물류관리시스템은 물류업자의 주요 업무인 거래처관리, 입출고 관리, 재고관리, 화물배송상태 등의 업무를 PDA 등의 모바일 시스템으로 데이터 입력, 결과 송수신을 지원하고 물류업체 및 물류센터등과의 시스템 연동을 통해 정보자원의 통합관리와 운영이 가능한 물류/유통통합 서비스이다. 모바일 물류관리시스템의 특징을 살펴보면 첫째 무선통신모듈과 스캐너 기능이 결합된 M Scan Phone을 사용하여 창고 재고의 적정화와 합리적 경영이 가능함, 둘째 물류 시스템의 불합리성을 배제하고 합리적인 시스템을 구축하여 생산관리 및 판매조직을 보다 합리적으로 운용이 가능함, 셋째 배달화물이 추적기능은 물론 화물배달지령처리, 진행 및 확인, 배송사원실적관리 등 무선단말 환경을 제공하는 특징이 있다.

한진택배는 2000년 상반기부터 무선통신과 PDA를 연계한 모바일 시스템의 도입으로 인터넷을 통해 화물을 예약, 접수하고 화물추적 서비스를 제공하고 있으며, 1일 1회 집배송서비스를 1일 2회로 전화하고, 접수로부터 30분 이내 집하하는 '30분 집하서비스'와 시간지정 서비스' 등 고품질 서비스를 통해 기업 고객의 SCM(공급망관리)에 효과적으로 사용하고 있다.

<그림 2-12> 모바일 물류관리 시스템의 구축사례



의료분야에서는 삼성서울병원·서울아산병원등 주요 대형 병원들이 무선 디지털 진료를 할 수 있는 전자의무기록(EMR)시스템과 무선 PDA 도입을 추진하고 있으며, EMR이란 환자의 정보와 진료기록, 과거병력, 처방, 통계 등을 디지털 처리해서 중앙컴퓨터에 저장해 놓은 뒤 의료진이 자신의 단말기로 확인하는 시스템을 말한다.

보험판매분야에서는 대한생명을 사례로 볼 수 있으며, 보험판매를 단계별로 지원하고 자료를 구성하며, 일관된 노트북과 지점서버로 고객활동 정보, 입력조회 등 고객관리가 기본적이며 고객의 요구를 반영하여 상품설계, 일정관리와 e메일, 무선모뎀 SMS 전송, 보험료계산, 계약심사 등 고객 서비스를 우선으로 시행하고 있다.

방문판매분야에서는 용진코웨이가 '02년 4월 업계 최대 규모인 8500여대의 무선 PDA를 도입하고 모바일 영업시스템을 구축하였다. 모바일 영업시스템을 통해 방문시간 약속과 작업 일정 작성, 고객 이력 관리 등 업무를 처리하는 것과 정기점검내용과 AS 내용 등록, 부품 재고 파악까지 가능하여 모바일 컴퓨팅으로 방문판매의 중점인 고객관리 측면에서 큰 호응을 불러일으켰다.

이마트는 최근 한국형 할인점시스템의 특성을 집대성한 원스톱 운용체계 'e투데이'시스템을 개발하였다. e투데이 시스템은 실시간으로 매출 및 고객정보를 파악하고 필요할 때 바이어가 산지에서 무선발주단말기를 통해 상품을 매입할 수 있으며, 상품이 매장까지 투입되는 전과정이 하루만에 처리된다. 기존에 고객정보 및 영업환경을 분석하는 DWH(Data Warehouse-Housing)시스템과 협력사와의 거래관계를 최적조건에서 가능케하는 GOT(Graphic Order Terminal)시스템을 기본으로 전자인증계약시스템과 모바일발주시스템 등을 통합하였다.

5. 농산물 유통에의 모바일시스템 응용 전망

가. 농산물 유통의 모바일 시스템의 문제점

농업 관련 분야에는 농협 유통이 하나로 클럽의 생산관리, 재고관리 등에 정보시스템의 일부를 활용하고 있으며 농산물 수배송 차량위치추적 등 농산물 유통부문에 모바일 시스템 일부가 활용되고 있으나 미진한 실정이다.

아울러 지금까지의 모바일 컴퓨팅을 활용한 정보기술은 대기업형 유통 및 물류기업을 위해 개발되어졌으며, 산업적 측면에서 특히 농업 관련 기술로 개발된 사례가 부족한 실정이다.

또한 산지유통에 대한 독자적인 분석이 부족하여 산지유통 전문조직은 브랜드관리 시스템 자체가 지향해야 될 방향 및 목표를 정하고, 생산 → 입고 → 예냉 → 저장 → 선별/포장 → 출하 등 표준적인 프로세스를 정립하여 적절한 시스템을 구축함으로써 기존 유통 및 물류관련 기술과는 다른 독창성과 혁신성이 반영되는데 어려움이 있다.

나. 앞으로 추가될 모바일 컴퓨팅을 이용한 농산물 유통의 전망

모바일 컴퓨팅 환경은 판매, 재고관리 등 기존 업무의 생산성 최적화는 물론 CRM 등 다양한 부문의 핵심도구로 활용될 것이며, 특히 농업의 특수성에 비추어 볼 때 그 활용은 매우 다양하다.

산지유통센터의 정보를 Networking하여 개별 기관, 조직별로 제공되던 농산물 유통 관련 정보를 집중시켜, 농산물 유통 관련기관과 조직이 정보

제공과 동시에 정보이용을 할 수 있게 될 것으로 예상되며, 국내뿐만 아니라 모바일 컴퓨팅을 통하여 해외에서 수출입되는 농산물의 물량과 가격을 통해 합리적인 농산물 유통정책 추진이 용이할 것으로 전망된다.

아울러 국내의 농산물 유통 관련정보를 실시간 주제별로 검색할 수 있도록 하여 시기별로 큰 차이를 보이는 계절과일의 출하시기를 조정하여 큰 폭락의 시장가격을 적정수준 유지할 수 있을 것이며, 앞으로 한국농림수산정보센터, 농협중앙회, 농산물 유통공사 등 소비지 정보제공기관과의 정보공유를 통해 모바일 콘텐츠를 제공하고 각 지역별 생산품의 품질, 가격 등을 한자리에서 비교하여 구매할 수 있도록 소비자의 구매정보가 확대될 것이다.

또한 농업 관측 전문가를 활용하여 예측단수, 예측생산량, 예측가격 등 전문가들의 분석을 모바일 정보화 하여 제공함으로써 농산물가공업체 및 대형급식업체 등에게 구매예측 정보를 제공할 수 있고, 농업 생산자에게 농협 등에서 제공되는 정보를 모바일 서비스를 통해 전달될 수 있으며, 농업에 재료인 비료, 농약 등의 값싼 공급추진방법에도 도입되어 원가 절감을 이룰 수 있을 것으로 예상된다.

차량위치추적서비스를 여러 기관과 통합적으로 사용함으로써 차량 배송의 적절성 및 공차율을 줄이는 것은 물론, 소비자에게 정확하게 배달, 운송할 수 있는 시스템을 도입할 수 있을 것으로 보이며, 고객이 주문한 농산물의 지불수단을 현금만이 아닌 모바일을 통해 카드 및 전자서명, 모바일뱅킹을 통해 지불하게 할 수 있으며, 포인트 적립을 통해 가입된 농산물 유통업체에서 마음대로 이용 가능하도록 하여, 고객의 이용을 확장시키는 방안도 검토할 수 있다.

특산품 위장과 상품의 질에 대해서 전문기관의 인증을 전산처리하여 소비자들이 직접 그 상품의 질을 눈으로 확인하지 않아도 안심하고 이용할 수 있도록 상품의 질적 평가를 단계별로 표기하여 제공하는 콘텐츠 개발도 가능 할 것으로 예상된다.

6. 산지유통전문조직의 모바일 컴퓨팅 추진전략

산지유통전문조직의 생산성 향상 및 효율성 제고를 위한 현실적인 대안을 제시할 수 있도록 모바일 시스템을 개발하는 것이 중요하다.

입고→예냉→저장→선별/포장→출하 등 산지유통전문조직의 농산물 유통에 대한 표준적인 프로세스를 정립하여 독창성과 혁신성이 반영되도록 시스템을 구축하는 것이 필요하며, 현장에서 발생하는 데이터관리 및 상품관리, 재고관리, 판매관리 등 효과적인 관리체계를 구축하고, 현장 정보가 수집, 보관, 분산되어 경영의 합리화, 과학화가 진전될 수 있도록 시스템을 구축하는 것도 필요하다.

산지유통전문조직의 종사자들에게 있어서 언제 어디서나 데이터베이스에 접근하여 원하는 정보를 등록, 수정, 삭제, 조회의 가능과 데이터의 실시간 처리 및 고객 대응력 제고 등 산지유통전문조직의 경쟁력과 통제력을 향상시킬 수 있는 시스템으로 구축하는 것이 필요하다.

생산 및 출하예측에 의한 자율수급조절이 가능토록 시스템을 확장하여 산지에서 농산물 출하의 중심적 역할을 담당할 수 있도록 유도하고, 고객관리(CRM)까지 가능한 가치사슬상의 모든 업무프로세스를 관리하기 위한 확장ERP(Extended Enterprise Resource Planning)로 발전시킬 수 있도록 시스템을 구축하는 것이 필요하다. 또한 산지유통전문조직의 내부정보화와

연계하여 합리적인 경영 및 의사결정이 가능한 시스템으로 확장하고, 향후 전자상거래(B to B, B to C) 등과 연동, 산지유통 조직의 핵심 전산화를 이끌어갈 수 있도록 시스템을 구축하는 것이 필요하다.

결국 산지유통전문조직 종사자들의 정보화 마인드를 높이고 다양한 정보를 제공함으로써 정보활용도를 제고시킬 수 있도록 시스템을 구축하는 것이 중요하다.

<제2장의 부록>

산지유통전문조직 정보화 실태 조사표

본 조사표는 산지유통전문조직의 정보화 전략을 수립하기 위한 연구의 일환으로 산지유통전문조직의 정보화 실태를 조사하기 위해서 작성되었습니다.

본 조사에 의해 밝혀진 내용은 오로지 연구 목적에만 이용될 것임을 약속드리며, 내실있는 조사가 될 수 있도록 많은 협조 부탁드립니다.

조 합 명 : _____

응 답 자 : 직위 _____ 성명 _____

조합주소 : _____

전화번호 : _____

(HP) _____

조사일시 : _____

1. 정보인프라 현황

1-1. 귀 조합에서 농산물 판매 업무를 위해 사용하고 있는 컴퓨터의 현황에 대해 적어주십시오. (신용 및 기타사업용은 제외)

구분	주요사양	대수
컴퓨터		
프린터		

(주요사양 예 : Pentium III 1.3 GH, Pentium IV 2.0 GH 등 해당 사양을 기입해주십시오.)

1-2. 귀 조합에서 사용하고 있는 네트워크 통신망의 현황에 대해 적어주십시오.

통신망	사용목적

(통신망 예 : 농협중앙회 통신망, 매가패스, 하나포스, 조합자체 LAN, 기타 통신네트워크 등 해당 통신망을 기입해주십시오.)

2 농협의 경제부문종합정보시스템 사용 현황

2-1. 귀 조합에서 사용하는 경제부문종합정보시스템의 분야는 무엇입니까?
(다수선택가능)

- | | |
|--------------|---------------|
| ① 판매관리 () | ② 구매관리 () |
| ③ 가공공장관리 () | ④ 미곡처리장관리 () |
| ⑤ 공판장관리 () | ⑥ 자금관리 () |
| ⑦ 회계관리 () | ⑧ 공통관리 () |
| ⑨ 기 타 () | |

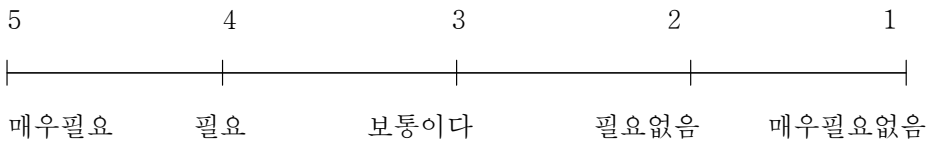
2-2. 귀 조합에서 경제부문종합정보시스템 사용으로 얻는 효과는 무엇입니까?

2-3. 귀 조합에서 경제부문종합정보시스템 사용 시 느끼는 문제점은 무엇입니까?

3. 종합관리시스템(ERP)도입의향

※ 종합관리시스템(ERP)란 기업내의 운영시스템인 생산, 판매, 인사, 회계, 유통 등을 하나의 시스템으로 통합하여 시스템을 재구축함으로써 기업운영의 효율성을 극대화 하려는 통합정보시스템입니다.

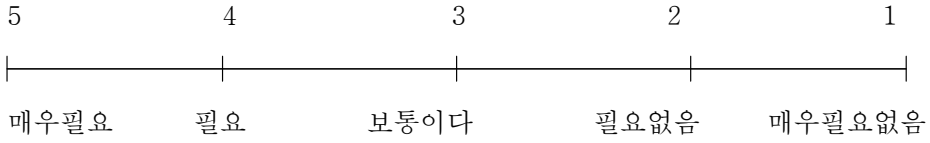
3-1. 귀 조합에서 유통업무의 효율화를 위해 종합관리시스템(ERP)의 도입이 필요하다고 생각하십니까? ()



3-2. 귀 조합에서 종합관리시스템(ERP)의 도입의향을 가지고 계시다면 그 이유는 무엇입니까?

3-3. 귀 조합에서 종합관리시스템(ERP)의 도입 시 기대효과는 무엇이라고

4-1. 귀 조합에서 모바일시스템의 도입이 필요하다고 생각하십니까? ()



4-2. 모바일시스템의 도입의향을 가지고 계시다면 도입하고자 하는 분야는 어떤 곳입니까?

(다수선택가능)

- | | |
|----------------|----------------|
| ① 입고관리시스템 () | ② 출하관리시스템 () |
| ③ 선별관리시스템 () | ④ 포장관리시스템 () |
| ⑤ 생산관리시스템 () | ⑥ 판매관리시스템 () |
| ⑦ 포장관리시스템 () | ⑧ 창고관리시스템 () |
| ⑨ 주문관리시스템 () | ⑩ 고객관리시스템 () |
| ⑪ 조합원관리시스템 () | ⑫ 거래처관리시스템 () |
| ⑬ 기타 () | |

4-3. 모바일시스템의 도입의향을 가지고 계시다면 그 이유는 무엇입니까?

4-4. 모바일시스템의 도입 시 기대효과는 무엇이라고 생각하십니까?

(중요하다고 생각하시는 것 2개를 선택하여 주십시오.)

- ① 현장에서 발생하는 정보 수집이 가능함 ()
- ② 언제 어디서나 정보의 취득이 용이함 ()
- ③ 시스템 연결의 신속함 ()
- ④ 농산물 수배송차량 위치추적이 가능함 ()
- ⑤ 재고파악이 용이함 ()
- ⑥ 대금결제가 용이함 ()
- ⑦ 기타

4-5. 모바일시스템의 도입의향을 가지고 계시지 않다면 그 이유는 무엇입니까?

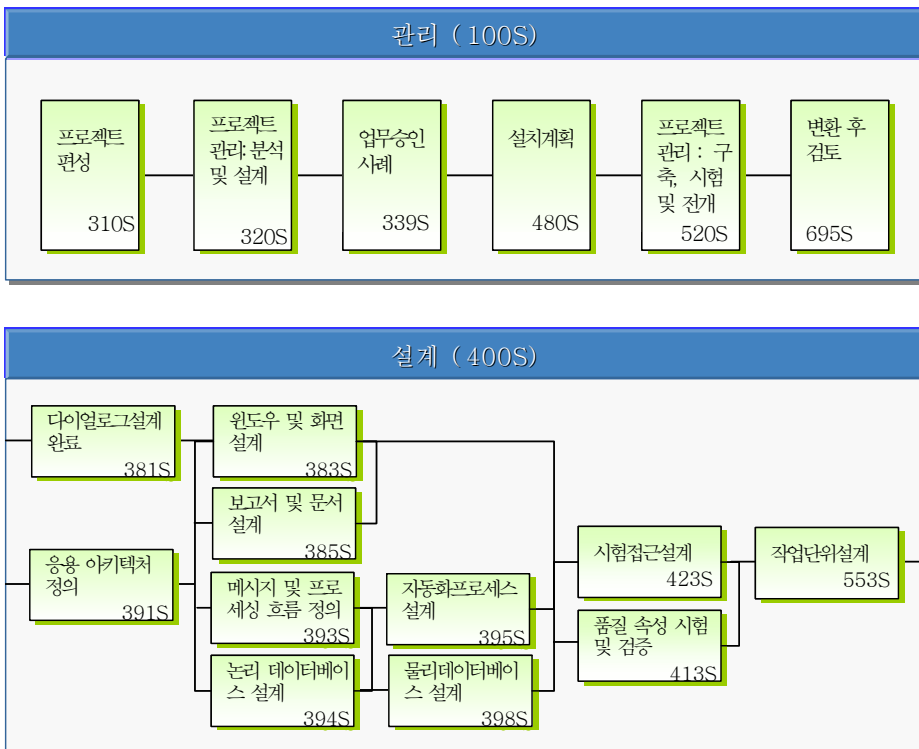
제 3 장 연구개발 및 평가 방법

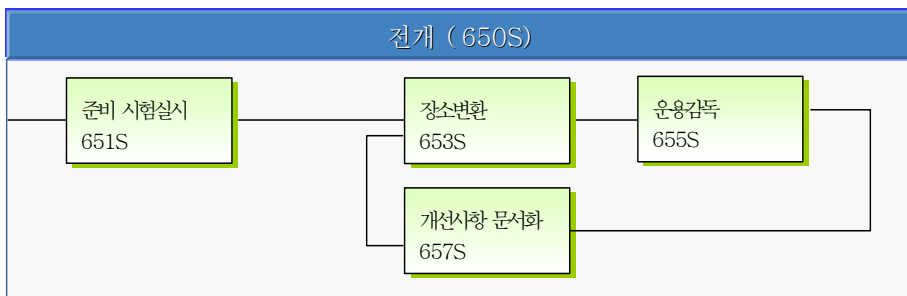
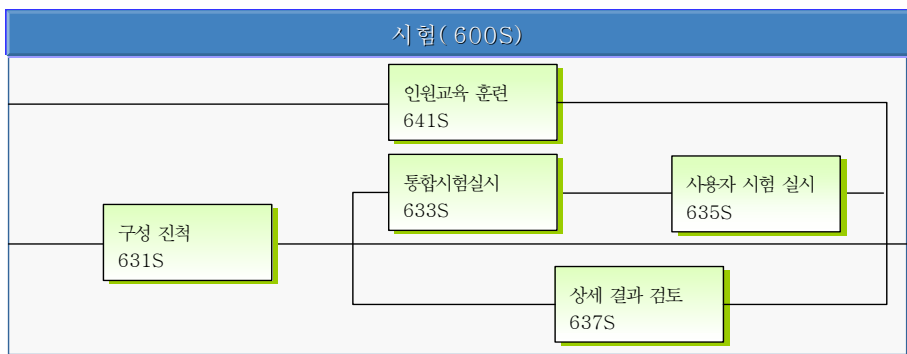
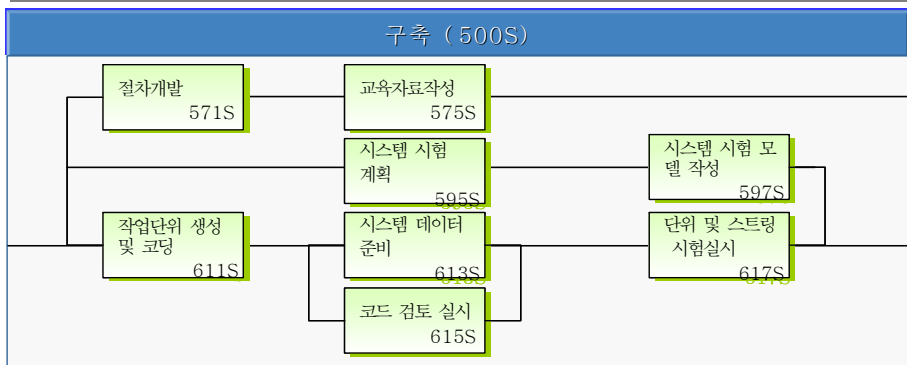
제 1 절 연구 개발 방법

본 연구는 유통전문가, 전산전문가, 산지유통센터 관계자 등이 참여하여 2년간 의 연구기간으로 추진되었으며, 산지유통센터의 업무를 분석하여 모바일을 활용한 정보시스템 설계 및 구현에 반영하였다. 시스템 개발 방법은 관리기법/1 개발방법을 채택하였다.

“관리기법/1”은 시스템 개발 전 단계 즉, 계획수립에서 분석 및 설계, 설치 및 운용을 지원하는 시스템 개발 방법론이며, 시스템 개발 및 운용관리, 품질보증활동 등을 지원 하는 프로젝트 관리 방법론이다.

<그림 3-1> 관리기법/1 적용 절차





제 2 절 개발 단계별 산출물

시스템 개발방법론에 따른 단계적 업무를 세분화하고, 각 단계별 명확한 산출물을 정의하며 산출물들을 표준화하여 참가자간의 상호이해와 의사소통을 원활히 하고, 기술사항들에 대한 정확한 이해 및 서술을 위한 자료가 될 수 있도록 한다. 산출물의 효율적인 표준화를 위하여 완벽한 문서화를 위한 사전기획, 공정 진행 중 수시 점검을 통한 표준화 통제, 최종 검토 및 승인 과정을 거침으로써 품질과 생산성을 극대화한다.

<표 3-1> 시스템 개발의 단계별 추진내역 및 산출물

시스템개발 생명주기	개발단계	단계에 따른 산출물
시스템 계획	1) 기초연구 2) 시스템 개발 착수 3) 업무 및 환경 분석 4) 개발 틀 설정 5) 정보 인프라 분석	연구보고서 업무 및 환경 분석서
시스템 분석	6) 사용자 요구 조사 7) 사용자 요구 분석 8) 기존 정보 시스템 분석 9) 업무 흐름 분석 10) BPR 11) 개발 목표 설정	사용자 요구 면담일지 사용자 요구 분석서 자료 흐름도(DFD) 업무흐름도
시스템 설계	12) 세부 업무 분석 13) 데이터 사전 작성 14) 데이터 베이스 스키마 작성 15) 메뉴 설계 16) 화면 설계 17) 보고서 설계	개념적 엔터티관계도(ERD) 데이터 사전 물리적 엔터티 관계도 메뉴 구성도 화면 설계서 보고서 설계서
구 현	18) 모듈 설계 19) 프로그래밍	프로그램 명세서

	20) 사용자 매뉴얼 작성 21) 인스톨 22) 베타 테스트 실시	사용자 매뉴얼 베타 테스트 결과서
유지·보수 및 평가	23) 디버깅 24) 개발 및 납품 완료 25) 사용자 만족도 조사 26) 유지 보수 27) 활용방안	시험·보완 보고서 UIS 측정 보고서 및 활용방안 보고서

제 3 절 세부개발 단계별 방법

1. 시스템 계획

1) 기초연구

시스템 개발 발주 후, 참여 인력 전원이 참석하는 워크샵 등을 통해 본 연구 사업의 의의 및 목적을 주지한다. 농산물 유통에서 산지유통센터가 차지하는 위치와 향후 발전방향에 대한 사전적 학습으로 참여 연구원의 인식의 공감대를 형성한 후, 실제 업무에 투입한다.

2) 시스템 개발 착수

시스템 개발 발주 후, 인력과 장비를 점검하고, 개발에 착수한다.

3) 업계 및 환경 분석

현장을 방문하여 현장의 특성 및 환경을 분석한다. 농산물 유통현황과 산지유통센터 업무 현장에 대한 충분한 사전이해는 개발 Tool 설정의 근거가 되

고, 사용자 요구 조사와 업무 흐름 분석에 있어서 올바른 방향을 제시할 수 있다.

4) 개발 톨 설정

현장의 환경과 특성을 고려하여 적합한 개발 톨을 설정한다. 개발 톨의 설정에 따라 시스템 분석 및 설계, 프로그래밍에 있어서의 방향이 결정된다.

5) 정보 인프라 분석

최종 사용자의 컴퓨터 및 정보 처리 능력을 분석하고, 현장의 물리적인 정보 인프라를 조사한다.

2. 시스템분석

6) 사용자 요구 조사

질문과 면담을 통하여 법규나 제한, 권한 등에 의해 규정되는 조직의 구조를 명확히 파악한다. 수행 방법으로는 닫힌 질문(Closed Questions), 열린 질문(Open Question), 브레인스토밍(Brainstorming), 그룹 토의(Group Consensus) 등이 있다. 사용자와의 면담은 일지 형태로 작성하여 보관, 관리한다.

7) 사용자 요구 분석

수집되고 분석된 정보 요구는 명세화하여 시스템 개발에 있어서 꾸준히 참조되어야 하며, 정보 요구들 간의 관계를 밝혀서 중복되거나 상치되는 일이 없

도록 하여야 한다. 이를 위해서 HTML등을 이용한 간단한 툴을 개발하여 관리하거나 CASE(Computer-Aided System Engineering)를 사용하는 것이 좋다.

8) 기존 정보 시스템 분석

해당 조직과 유사한 외국이나 국내의 정보 시스템, 또는 해당 조직에 현존하고 있는 정보 시스템에 대한 철저한 분석이 있어야 한다. 이러한 분석을 통해 해당 조직의 정보 요구와 문제점을 비교 파악할 수 있다. 기존 정보 시스템에 대한 사용자들의 평가는 향후 개발될 시스템의 방향을 설정할 수 있다.

9) 업무 흐름 분석

해당 조직의 현재 업무 프로세스와 데이터 흐름을 분석한다. 검증된 방법으로는 표준 분석(Normative Analysis), 전략 분석(Strategy Set Transformation), 주요 요인 분석(Critical Factors Analysis), 업무 분석(Process Analysis), 의사결정 분석(Decision Analysis), 사회-기술 분석(Social-Technical Analysis), 입력-기능-출력 분석(Input-Process-Output Analysis) 등이 있으며 이를 수행함으로써 현재 운용되고 있는 조직의 특징을 명확히 파악할 수 있다.

농업 시스템은 일반 기업이나 공업 시스템과는 확연히 구분되는 특성을 지니고 있다. 또한 농업 시스템에 있어서 업무 프로세스는 작목별로 큰 차이가 나므로 수시 현장 방문을 통해 업무 프로세스를 파악하여야 한다. 이 단계에서 산출되는 데이터 흐름도(Data Flow Diagram;DFD)는 조직의 전체적인 업무 흐름을 파악하고 업무의 중복성이나 효율성을 검증하는데 이용된다.

10) BPR(Business Process Reengineering)

사용자 요구 분석과 업무 흐름 분석을 통해서 조직 업무의 중복이나 비효율적인 업무 관행을 찾아내어 업무 프로세스를 수정한다. 정보 시스템의 개발은 단순한 업무 자동화뿐 만 아니라 조직의 업무의 효율성 자체를 재고하여 경쟁력을 가질 수 있도록 한다.

11) 개발 목표 설정

상기 작업들을 근거로 타당성 있는 정보 시스템 개발 목표를 설정한다.

3. 시스템 설계

12) 세부 업무 분석

업무 흐름도를 근거로 조직의 업무를 상세하게 조사, 분석한다. 업무 흐름도가 업무 프로세서와 데이터의 흐름에 초점을 맞추고 있는 반면, 개념적 엔터티 관계도는 데이터의 저장에 초점을 맞춘다. 설정된 정보 시스템 개발 목표에 의거하여 개념적 엔터티 관계도를 작성한다.

13) 데이터사전 작성

작성된 개념적 엔터티 관계도의 각 필드에 대한 데이터 사전을 작성한다. 데이터 사전을 근거로 물리적인 엔터티 관계도를 작성할 수 있다.

14) 데이터 베이스 스키마 작성

개념적 엔터티 관계도와 데이터 사전을 근거로 물리적인 엔터티 관계도를 작성한다. 효율적인 질의(query)를 고려하여 물리적인 엔터티 관계도를 완성

한다. 완성된 물리적 엔터티 관계도를 매핑(mapping)하여 실질적인 데이터 베이스 스키마를 작성한다.

15) 메뉴 설계

개발될 시스템의 메뉴를 구성한다. 메뉴 설계는 사용자와의 지속적인 인터뷰를 통해 프로토타이핑 기법으로 수행한다. 사용자가 가장 편하게 사용할 수 있는 메뉴를 설계한다.

16) 화면 설계

업무 흐름도와 물리적 엔터티 관계도를 근거로 개발될 시스템의 입력 화면을 설계한다. 프로토타이핑을 수시로 작성해 사용자에게 검증을 받고, 지속적인 사용자 요구사항을 받아들여 적용시킨다. 입력 화면에 있어서 지속적인 프로토타이핑은 개발 완료 후 사용자들의 시스템에 대한 만족도를 높인다.

17) 보고서 설계

업무 흐름도와 물리적인 엔터티 관계도를 근거로 개발될 시스템이 출력할 화면과 보고서를 설계한다. 정보 시스템의 가장 큰 목적은 사용자가 원하는 정보를 출력하는 것이므로, 철저한 프로토타이핑을 통해 사용자가 원하는 보고서가 출력될 수 있도록 한다.

4. 구현

18) 모듈 설계

구조화된 프로그래밍을 위해서 모듈을 설계한다. 모듈은 자료 흐름도 잘 설계된 모듈은 프로그램 내에서의 일관성을 유지할 수 있도록 해주고, 차후 비슷한 프로세서를 프로그래밍 할 때 유용하게 사용될 수 있다.

19) 프로그래밍

업무 흐름도와 데이터 사전, 물리적인 엔터티 관계도, 화면 설계서, 보고서 설계서, 설계된 모듈을 이용해서 프로그래밍을 한다. 프로그래밍을 과정에서 수시로 프로토타이핑 기법을 이용해서 사용자 요구를 지속적으로 반영시킨다.

20) 사용자 매뉴얼 작성

사용자가 시스템을 쉽게 사용할 수 있도록 사용자 매뉴얼을 작성한다.

21) 인스톨

구현된 시스템을 사용자가 사용할 수 있도록 인스톨한다.

22) 베타 테스트 실시

사용자가 구현된 시스템을 직접 사용하도록 함으로써 시스템이 가지고 있을 수 있는 버그나 문제점들을 지적하도록 한다.

5. 유지보수 및 평가

23) 디버깅

베타 테스트에서 지적된 버그나 문제점들을 수정한다.

24) 개발 및 납품 완료

버그 및 문제점들을 해결한 후 다시 인스톨하여 납품 완료한다.

25) 사용자 만족도 조사

개발 후 사용자들에 대한 만족도 조사를 실시한다.

26) 유지보수

시스템의 정상적 가동을 위해 점검하고, 데이터 수정 등 유지보수를 실시한다.

27) 활용방안

개발 시스템의 활용도 제고를 위한 방안을 마련한다.

제 4 절 정보시스템 평가 방법

1. 정보시스템 평가의 이론적 배경

IT투자로 발생하는 효과에는 유형적 요소와 무형적 요소가 혼재되어 있다.

특히 최근 들어 IT의 급속한 변화와 기술영향의 복잡성 증대로 특정기술 선택 및 시스템 도입에 대한 장기적인 영향을 파악하는 것이 쉽지 않으며, 특정시스템의 변화가 미치는 개별적인 영향을 파악하기도 힘들어졌다. 이러한 정보화 효과의 복잡성, 간접성, 장기성 증대에도 불구하고 정보화 투자비용은 지속적으로 증가하고 있으며 효율적인 IT 투자관리를 위한 정보화 경제성 분석에 대한 요구가 커지고 있다.

정보화는 본래 수단적, 지원적 측면이 강하여 인간계와 연계되지 않고서는 효과를 발휘할 수 없다는 사실, 기술혁신이 빨라서 장래의 코스트가 확정되기 어렵다는 점, 교육, 보급 활동 등 소프트적인 면도 포함한 단계적인 투자가 필요한 점 등의 이유로 도로, 철도, 공공시설 등과 달리 최종적인 편익 및 비용을 평가하는 것이 어려운 것도 확실하다. 또한 종래 하드 주도의 투자에 비해 소프트웨어의 강화에 의한 효과창출 대책으로서 정보화를 취한다는 생각도 존재한다. 그러나 지금까지 정량적인 이론으로 일관한 정보화 사회론에서 일보 전하여 무엇인가 정량적 평가수법을 확립할 시기에 왔음은 틀림없다. 덧붙여 말하면, 도로의 경우 그 정비효과로서 건설 후 발생하는 생산력 확대효과(스톡효과)와 건설중의 수요창출효과(플로우 효과) 및 보행경비편익, 시간편익 등의 직접효과와 생산력 확대효과, 물가절감, 산업진흥 등의 간접효과로 나누어 편익효과가 행해지고 있다.

2. 정보화 투자 효과 유형 및 효과 지표 도출 방법

모바일 컴퓨팅을 이용한 산지유통 전문조직의 유통관리시스템 개발로 인한 효과분석은 전산원의 정보화 경제성 평가 연구(2001)에서 보는 바와 같이 IT 투자의 효과를 운영적 효과와 전략적 효과로 구분할 수 있다. 운영적 효과는 주로 비용감소와 관련되며 전략적인 효과는 수익증가 효과와 관련된다.

<표 3-2> 투자효과의 유형 2)

효 과	내 용	세 부 항 목
운영적 효과	업무기능성 향상	·업무생산성 향상 ·업무처리시간 단축
	의사결정력 향상	·의사결정시간 단축 ·정보품질 향상
전략적 효과	경쟁력 향상	·제품 차별화 ·절대적 비용 우위
	고객관계 강화	·거래고객 확대 ·서비스 강화

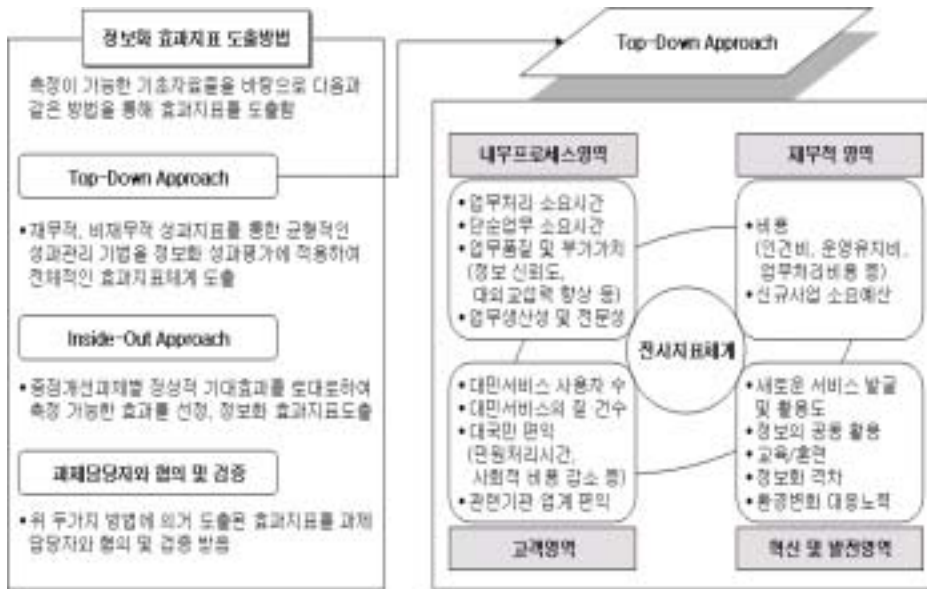
이와 같은 투자효과 유형은 크게 유형적 요소와 무형적 요소들이 혼재되어 있으며, 유형적 요소에 대한 예로는 ERP시스템 도입에 의한 인력 감소, 생산성 향상 등에 따른 실질적 비용 감소가 해당되며, 무형적 요소는 새로운 시스템이 가져다주는 고객만족의 증가, 프로세스 효율성 향상, 지식공유 활성화 등을 예로 들 수 있다.

이러한 유형에 비추어 본 연구에 맞고 취득 가능한 기초 자료들을 바탕으로 다음 세 가지 방법을 통해 효과지표를 도출하고자 한다.

Top-Down Approach 방법은 일반적인 균형 성과표(BSC : Balanced Score Card) 기법을 정보화 성과평가에 적용, 전체적인 효과지표체계를 도출하는 방법이며, Inside-Out Approach는 정성적 기대효과를 기반으로 가시화 가능한 효과를 선전하여 이로부터 정보화 효과지표를 도출하는 방식이다.

<그림 3-2> 정보화 효과지표 도출방법

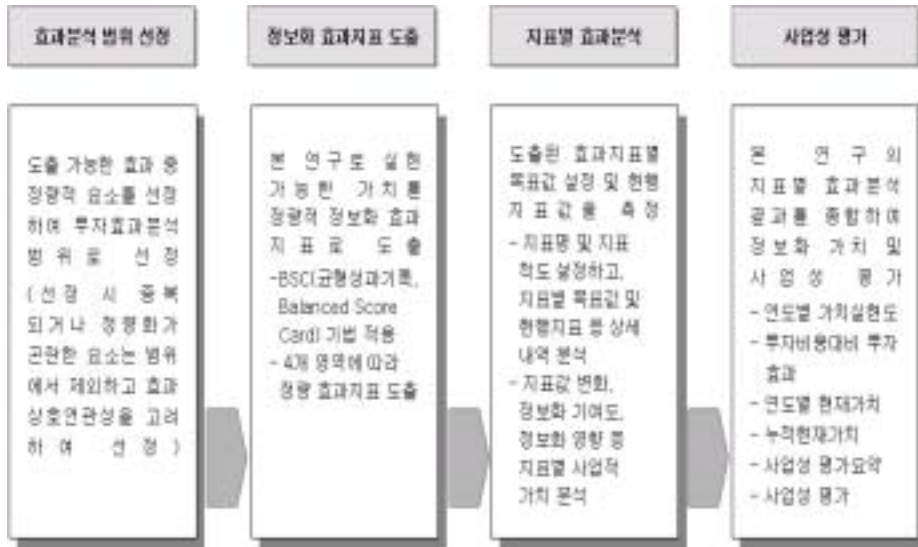
2) 한국전산원, 정보화 경제성 평가, 2001



3. 정보화 투자효과 분석절차의 주요 이슈

IT 투자관리 체계의 구현과정은 세단계로 구분할 수 있다. 먼저 분석범위 선정단계에서는 도출 가능한 효과 중 정량적 요소를 선전하여 투자효과분석범위로 선정한다. 선정 시 중복되거나 정량화가 곤란한 요소는 범위에서 제외하고 효과 상호간 연관성을 고려하여 선정한다. 본 연구로 실현 가능한 가치를 정량적 정보화 효과 지표로 도출하는 단계에서는 IT 투자관리의 방향성 및 현업의 요구사항을 분석하여 작업이 이루어진다. 지표별 효과분석은 도출된 효과지표별 목표값 설정 및 현행 지표값을 측정하는 단계이며, 사업성평가는 지표별 효과분석 결과를 종합하여 정보화 가치 및 사업성 평가를 하는 단계이다.

<그림 3-3> 정보화 투자효과 분석절차



상기 각각의 단계에서 무형적 효과의 정량화 및 IT투자의 영향도 파악, 평가의 타당성 및 신뢰성 확보, 투자관리의 실행력 강화 등 다양한 이슈들이 제기될 수 있다. IT 투자효과와의 경제성 분석상의 이슈는 정보화를 추진하기 위해서는 정보화를 통해 얻을 수 있는 경제적 효과를 파악할 수 있는 지표의 선정 및 측정이 필요하다. 그런데, 정량화된 효과중심으로 지표를 선정하다보면 효과와 비용측면만을 중점적으로 다루게 되고, 기회비용을 포함한 위험 요인이나 이외 영향요인들에 대해서는 충분히 다루지 못할 수 있다. IT 투자평가의 정교화를 위해서는, IT 시스템 및 투자 유형에 따라 평가를 차별화할 필요가 있다. 그러나 대개의 평가지표가 투자 유형을 고려하지 않거나 고려한다고 하더라도 정성적인 효과와 전략적인 영향을 충분히 반영하지 못하고 있다. 또한, 대부분의 IT 투자평가는 개별시스템의 투자효과에 대한 가이드만을 제공하고 전략적 가치와 타시스템과의 연계효과에 대한 분석이 미흡하여 IT 투자대안의 선택기준을 제공하는 데는 한계가 있다.

IT 투자관리체계 관리상의 이슈는 현재까지의 정보화 투자평가는 평가체계와 평가기술의 부족으로 인해 평가자체가 형식적이고 그 타당성과 신뢰성 확보가 어려운 것이 현실이다. 공공기관의 정보화 평가제도 현황을 파악한 한국전산원 자료(이석재, 2000)에 의하면, 대부분의 조직의 정보화 목표와 개별 정보화 사업 간의 연계가 미흡한 것으로 나타났다. IT 투자평가체계의 도입으로 인해 비용 중심의 의사결정이 강화될 우려가 있으며 이로 인해 많은 경영기회를 잃게 될 위험도 있다. 따라서 단기적이고 정량적인 가치 이외에 장기적이고 정성적인 가치에 대한 고려와 전략적 측면에서 IT 투자를 고려하는 제도의 보완이 필요하다.

제 4 장 시스템 설계 및 개발

제 1 절 시스템 구축 환경 정의

1. 구축 대상의 개요

그동안 산지유통은 개별 농업인을 비롯하여 20,000여개의 작목반, 5,000여개의 영농법인, 1,200여개의 협동조합 등 영세규모의 조직에 의해 주도되어 왔으나, 이제 이와 같은 영세규모 조직으로는 상품의 균일성, 공급의 지속성, 브랜드 상품 선호 등 시장의 요구를 도저히 수용하기 어려운 실정이다.

특히 소비지에 대형유통업체가 속속 등장하고, 전자상거래가 활성화되는 등 농산물 판매여건이 크게 달라짐에 따라 산지의 농산물 마케팅조직의 규모화를 집중 지원하기로 하였다. 2003년도까지 288개 협동조합 및 영농조합법인이 산지유통전문조직으로 선정되었으며, 총 10,516억원의 자금을 집중 지원하여 산지유통전문조직으로 육성하였다.

<표 4-1> 산지유통전문조직 선정현황

(단위 : 억원, 개소)

구분	2000	2001	2002	합계
조직수	99	62	127	288
총사업비	2,641	2,875	5,000	10,516

선정된 전문조직은 공동출하, 품목별 전문화, 해당 품목 전국협의회 가입 및 지역협의회 결성, 공동계산, 파렛트 출하 등이 높은 조직으로 ① 공동선별, 공동수송, 공동판매를 통한 유통효율 증진 ② 조합당 유통사업 취급품목을 3개

품목 이내로 제한, 주산지화·전문화 촉진 ③ 지역협의회 운영활성화를 통해 자율적인 수급조절 능력 및 시장교섭력을 강화하는데 목적이 있다.

따라서 주산지화·전문화와 공동선별, 공동수송, 공동판매 및 자율적인 수급조절과 시장교섭력을 증진시키기 위한 하나의 대안으로써 정보통신기술(ICT)인 모바일 컴퓨팅을 도입하여 산지유통전문조직의 유통 효율성과 수급조절 및 시장교섭력을 제고할 필요가 있다.

가. 구축 대상의 선정

본 연구는 산지유통센터의 정보화를 모바일컴퓨팅 환경을 통하여 구현하는 것으로서, 산지유통센터 중에서 운영이 활성화되어 있으면서, 운영자들이 정보화마인드가 있어서, 모바일컴퓨팅이 실질적으로 적용 가능한 사업장을 선정할 필요가 있다.

본 연구에서는 경제사업의 활성화를 위해 읍면 단위의 조합을 시 단위 광역조합으로 통합하고, 산지유통센터를 설립 운영하면서, 파머스마켓 유통사업단을 조직하여 마케팅 활동을 적극적으로 펼치고 있는 순천농협을 시스템 구축 대상으로 했다. 순천농협은 산지유통센터와 자체 판매장인 파머스마켓을 설립하여 산지 농산물의 상품화와 판매제고를 위해 각별한 노력을 기울여 왔다. 특히 마케팅에 있어서도 초기 도매시장 중심의 단순 출하방식에서 다양한 소포장 상품의 개발 등 상품성 제고를 통해 자체 판매장은 물론, 대도시 백화점, 물류센터, 대형할인점 등으로 판로를 확대하고 있다.

순천농협은 산지유통센터가 담당하여야 할 거의 대부분의 기능들을 효율적으로 수행하고 있으며, 시설이나 정보화 장비 등도 충분히 확보하고 있다. 특히 농협의 판로가 도매시장 등 전통적인 판매처에서 벗어나 자체 파머스마켓,

대형 할인점, 백화점 등을 다변화되어 있다는 점이 본 사업의 대상지로서 가장 적합하다는 판단이다.

나. 순천농협 농산물 산지유통센터 일반 현황

1) 현황

<표 4-2> 순천농협 농산물 산지유통센터 일반현황

명 칭	순천농업협동조합	대표자 성명	채 대 홍
주 소	전남 순천시 조곡동 159-9	전 화 번 호	061)743-3581
조직결성일	1961년 8월 15일	조 직 원 수	16,677명

2) 2003년 사업실적

<표 4-3> 순천농협 농산물 산지유통센터 사업실적

품목	출하량(톤)			상표명 순천OO드림원	주요 출하처 공영도매시장, 유통센터
	계	규격출하	일반출하		
오 이	169,223	152,300	16,923	"	"
가 지	155	139	16	"	"
딸 기	3,321	996	2,325	"	위탁상, 유통센터
참다래	522	522	0	"	공영도매시장, 유통센터
매 실	2,920	2,336	584	"	"
복숭아	329	197	132	"	"
기 타	3,101	930	2,171	"	"
계	179,571	157,420	22,151		

3) 공동유통시설 보유현황

<표 4-4> 순천농협 농산물 산지유통센터 공동유통시설 보유현황

집하시설	선별(과) 시설	저온 저장시설	일반창고	예냉시설	선별(과) 기	기타공동 시설및장비
1동 250 평 (30)	1동 250 평 (30톤)	2동 100 평 (200톤)	51동 5,315 평 (29,814)	2동 50평 (50톤)	2조 (15톤)	냉동실 (34평)

2. 구축 대상 시스템의 기술적 동향

순천농협 농산물 산지유통센터를 대상으로 하는 산지유통센터 종합관리시스템은 기술적인 측면에서는 모바일 시스템으로 정의할 수 있다.

가. PDA의 개요

1) PDA의 특징

PDA(Personal Digital Assistant)는 개인 휴대정보단말기를 지칭하는데, 주로 주소록, 일정관리, 계산기 등 전자수첩에서 제공하는 기능 외에 별도의 운영체제를 통해 노트북에서 수행하는 대부분의 기능을 가지고 있는 이동 컴퓨팅을 목적으로 한 기기이다. 이러한 PDA는 CPU, 메모리, OS 등으로 구성되고, 소형 키보드, 터치 스크린, 펜 입력 방식 등 다양한 데이터 입력 장치를 가지고 있다. 또한 휴대를 목적으로 하는 기기이므로 디스플레이의 사이즈에 제한을 받는 단점을 지니고 있다. 초기에 단순한 오거나이저 기능을 가지고 있던 PDA는 간단한 문서 작성과 더불어 단순하지만 다양한 PC의 기능을 구현하게

되었으며, 최근에는 무선 인터넷 접속 기능으로 사용범위가 점점 확대되고 있다.

2) PDA의 발전과정

1984년 Psion사의 “Psion Organizer”가 최초의 PDA라 할 수 있는데, 이 제품은 데이터베이스를 이용한 PIM(Personal Information Management)기능을 가지고 있었다. 공식적으로 PDA란 용어가 사용되기 시작한 것은 1993년 Apple사가 개발한 “Newton Message PAD 100”이 출시되면서부터이고, 이후 본격적인 PDA 제품이 등장하기 시작하였다. 초기 PDA 시장은 높은 가격과 어플리케이션의 부족으로 대중화에는 실패하였고, 1996년 Palm OS기반의 “Palm Pilot”의 출시로 PDA 제품의 대중화가 시작되었다고 할 수 있다.

최근 들어, PDA는 무선 인터넷 접속 기능의 추가로 다양한 통신 모듈을 탑재한 제품이 출시되고 있다.

3) PDA의 분류

보통 개인정보단말, 휴대용 정보단말 등으로 불리는 PDA는 노트북이나 Handheld PC 등과 함께 휴대를 목적으로 하는 기기를 일컫는다.

Tablet PC 주로 컨텐츠에 접속하거나 PIM을 위해 사용, 데이터 수집, 지도 보기, Tablet PDA의 산업용 버전은 바코드 스캐너, 재고 컨트롤, 특히 관련 자료 및 타산 업과의 통합에 대한 어플리케이션 등의 기능 3~4인치 소형 디스플레이, 6온스 이하의 무게, 스타일러스 입력 방식 Palm OS, Windows CE를 기반으로 하는 PDA(일부 제외)

Clamshell PDA 상대적으로 큰 스크린을 통해 컨텐츠 제작, 워드 프로세싱, 엑셀, 파워포인트 등과 같은 모바일 오피스 어플리케이션을 포함, 판매 자동

화, 의료 부문에 주로 사용 1파운드 이하의 무게, 디스플레이, 소형키 보드, 스타일러스를 통한 입력방식 HPC 기종- 가트너에서 분류한 PDA 분류

일반적으로 PDA는 형태에 따라 Tablet 또는 Clamshell PDA로 불리기도 하고, 시장 구분에 따라 SHD(Smart Handheld Device)로 불리고 있다.

SHD는 다음 에서 보는 것처럼 핸드헬드 컴패니언과 버티컬 어플리케이션 디바이스로 구분하여 정의할 수 있는데, 버티컬 어플리케이션 디바이스는 데이터 접속, 관리, 입력, 수집 기능을 핵심으로 하는 기기로 펜 또는 키패드를 주 입력기반으로 가지고 있고, 바코드와 같은 보조장치를 사용하기도하며 주로 산업용 및 업무용으로 쓰이고 있다. 핸드헬드 컴패니언은 데이터 접속 및 관리 기능을 핵심으로 하는 기기로 데스크탑 또는 노트북 PC 보조용으로 설계된 기기이고, 주로 PIM이나 데이터 창출과 같은 개인 용도로 사용되고 있다.

나. PDA 시장 현황 및 전망

1) PDA 시장의 산업 구조

PDA시장의 산업구조를 살펴보면, 무선데이터 서비스 제공업체와 OS 및 ASP제공업체, 데이터 서비스를 제공하는 셀룰러 폰, 스마트 폰 등과 같은 경쟁제품 등으로 구성되어 있다. 이러한 산업구조 내에서 OS, ASP 제공업체들은 새로운 버전의 어플리케이션 및 킬러 어플리케이션의 제공으로 PDA의 부가적인 기능을 실현시키고 있고, 무선데이터 서비스 제공업체들의 참여로 PDA의 적용 범위가 점점 확대되고 있다. 따라서 PDA 시장은 기업 및 일반 소비자 고객들로 하여금 수요를 창출하는 기반을 마련하게 된다.

2) 시장 성장 요인 및 장애 요인

현재 세계 PDA 시장은 전세계적인 경기 침체로 인한 수요 감소와 업체간 과

잉 경쟁으로 인한 가격 하락 등 시장이 위축되어 있고, 성장률이 둔화되고 있는 실정이지만, 앞으로 기업 고객과 일반 소비자의 수요 확대로 시장이 점차 성장할 것으로 전망하고 있다. PDA 시장은 주로 기업시장을 중심으로 성장해 왔으나, PDA 제품의 기능 확대에 따라 많은 잠재력을 가지고 있는 일반 소비자 시장에도 점차 영역을 확대하고 있으며, 새로운 형태의 PDA 제품, 업데이트된 운영체제 및 부품의 개발 등이 수요를 촉진하는 요소가 될 것으로 보고 있다. 하지만, 짧은 배터리 수명, PDA를 일반 상품화시키기 위한 과도한 가격 경쟁으로 인한 부정적인 영향, 서비스와 콘텐츠의 부족 등 수요를 방해하는 요인도 존재하고 있다.

- 성장 요인 PDA 기능 확대에 따라 시장 잠재력 부각
- 서비스와 콘텐츠 결합으로 무선 인터넷 솔루션 통합
- 기업의 PDA 채택으로 인한 시장 확대의 기회 마련
- 로컬 벤더의 마케팅 강화에 의한 수용 증가
- 장애 요인 하이엔드 기기의 짧은 배터리 수명으로 인한 수요 확산의 저해
- 기기의 일반 상품화에 따른 부정적인 영향
- 서비스와 콘텐츠의 부족으로 인한 다기능 제품의 매력 감소
- 로컬 벤더의 난립에 따른 표준화의 부재와 상호 운영성의 결핍

3) PDA 벤더 현황 및 전망

2000년과 2001년 출하량에 따른 전세계 PDA 벤더의 시장점유율 추이를 보면, 다음과 같은데, 2001년 한 해 동안 Palm이 약 500만 대의 출하량으로 전체 시장의 38.6%를 차지하여 2000년에 이어 가장 높은 점유율을 차지하였고, 뒤를 이어, Handspring과 Compaq이 12.6%, 9.8%의 점유율로 각각 2위와 3위를 차지하였다. 성장률 기준으로 볼 때, Sony가 700%를 상회하는 성장률을 기록하여 가장 빠르게 성장하는 벤더임을 주목해 볼 수 있다.

IDC의 최근 자료에 의하면, 2002년 1/4분기 세계 PDA 시장 규모가 전년 동기 대비 12.1% 감소하였다. Palm이 전년에 이어 선두를 지키고 있고, 지난해 3위를 기록한 Compaq이 2위, Handspring이 3위를 차지하여 지난해와 순위가 뒤바뀌었다. 높은 성장률을 보이고 있는 Sony는 2002년 1/4분기 세계 PDA 시장에서 전년 동기 대비 250%의 성장률을 기록하여 4위를 차지한 것으로 나타났다. 한편, 아시아/태평양 지역 PDA 시장에서 미국 벤더가 차지하는 비중은 그리 높은 편이 아니라고 볼 수 있는데, Palm과 Compaq만이 Top 5 순위에 들어가 있는 실정이다. 이 지역 선두를 차지하고 있는 벤더는 Minren으로 2001년 아시아/태평양 시장의 총 19%를 점하고 있는 것으로 나타났고, 2000년 대비 95.5% 성장률을 보여주었다. 뒤를 이어, Hi Tech Wealth는 2001년 총 18%의 점유율로 2위를 차지하였다. 중국 시장은 저렴한 가격과 지역적 특성을 고려한 어플리케이션으로 인하여 지역 벤더가 시장의 대부분을 점하고 있지만, 앞으로 전세계 많은 벤더들이 치열한 경쟁을 벌일 것으로 전망된다.

4) PDA 시장 현황 및 전망

가) 전세계 PDA 시장

IDC에서 최근 발표한 자료에 의하면, 2001년 전세계 PDA 시장은 2000년 대비 출하량 측면에서 약 24%, 매출액 측면에서 33% 성장한 것으로 추정된다. 이는 2000년 134%의 성장에는 크게 미치지 못하는 수치인데, 이러한 하락의 원인으로 전세계적인 경제 침체를 들 수 있다. 하지만, 오는 2003년부터는 가정 및 기업 시장의 수요 회복이 예상되는 가운데, 전세계 시장 규모가 1,600만대를 넘어설 것으로 예측되고, 2005년에는 출하량이 약 4,400만대에 이를 것으로 전망된다.

나) 지역별 PDA 시장

대부분의 주요 PDA 벤더와 PDA 소프트웨어 개발자들은 북미에 기반을 두고 있어서, 이 지역이 PDA 출하량의 주류 시장이 되고 있는 실정이다. 2001년 전세계 출하량의 약 50%를 차지하고 있는 미국은 2001년 약 10.4% 정도 성장한 740만 대의 출하량을 기록하였고, 일본의 경우 약 32% 정도 성장한 것으로 나타났다. 지난 2000년 200%가 넘는 고성장을 기록했던 아시아/태평양 지역의 PDA 시장은 2001년 약 50% 정도 성장하였고, 20%의 점유율을 기록하였다. 아시아/태평양 지역의 점유율은 점점 증가하여 오는 2005년에는 전세계 시장의 26.3%를 점할 것으로 예상된다.

다) 국내 PDA 시장

아직까지 국내 PDA 시장은 규모가 크다고 볼 수는 없지만, 광범위한 통신 인프라와 적극적인 정부의 지원으로 인해 높은 잠재성을 보여주고 있다. 또한, PDA 가격의 하락으로 인한 전문가 시장의 수요확대와 우편, 금융, 유통 분야의 PDA 사용의 확대에 의하여 국내 PDA 시장은 앞으로 계속 성장할 것으로 전망된다.

라) PDA용 OS 시장

현재 대표적인 핸드헬드 기기 탑재 OS로는 MS의 Windows CE, Palm의 Palm OS, Symbian의 EPOC32(Symbian Platform) 등이 있는데, Palm의 Palm OS가 시장에서 가장 높은 점유율을 차지하고 있다. 2002년 하반기 PDA OS 시장은 Palm의 Palm OS 5와 Microsoft의 Windows CE 4.0 등 과 같은 새로운 플랫폼의 출시로 양사의 경쟁구도가 성립될 것으로 보이지만, Windows CE 4.0의 경우, Microsoft가 Pocket PC 기기용 레이어의 다음 버전을 개발하고 있기 때문에 판매량에 큰 영향을 미치지 못할 것으로 예측된다.

국내 OS 시장은 셀빅 OS와 Windows CE가 주류를 이루고 있는 가운데, Palm OS가 점유율을 확보하기 위해 적극적인 양상을 보이고 있다. 셀빅 OS는 우수한 한글지원 기능, 국내 사용환경에 적합한 구조 등으로 인하여 아직까지 많은 인기를 얻고 있는 플랫폼으로 50%가 넘는 시장 점유율을 보여주었으나, 2001년 국내 PDA 시장에서 Compaq iPAQ 시리즈 등 Windows CE를 탑재한 기기의 고성장으로 인하여 상대적으로 점유율이 많이 하락하고 있는 실정이다.

3. 시스템 개발시 고려사항

가. 목표에 적합한 범위 설정

1) 사용의 편의성

- 가) 높은 이동성과 휴대의 용이성이 가능한 시스템 구축
- 나) 비상시적 접속 등 시공간의 한계성 극복이 가능한 시스템 구축
- 다) 내외부 환경변화에 유연한 시스템 구조 구현

2) 대고객 서비스 제고

- 가) 실시간 접수 등 대고객 대응력이 향상된 시스템 구축
- 나) Real Time성을 추구하여 고객 신뢰도 확보가 가능한 시스템 구축
- 다) 현장 밀착형 대고객 전개가 가능한 시스템 구축

3) 시스템 구축, 확장, 연계 용이

- 가) 대용량의 데이터 처리 및 정보의 실시간 제공 시스템 구축
- 나) 기존시스템 존재시 자동 연계에 의한 시스템 통합 구축
- 다) 유지보수의 용이성 및 사용하기 쉬운 사용자 위주의 컴퓨팅 환경 구축

나. 연성과 확장성의 고려

전문적인 전산인력과 조직을 갖춘 일반기업과 달리 비교적 규모가 큰 순천농협농산물산지유통센터를 포함한 거의 모든 산지유통센터가 별도의 전산인력을 확보하고 있지 않으며 이것은 개발업체로부터의 기술이전을 통한 자체적인 유지보수가 현실적으로 불가능하다는 것을 의미한다. 따라서 단위업무의 변경이나 대내외적인 환경변화에 시스템적인 대응이 극히 취약할 수밖에 없다.

결국 개발되는 모바일시스템은 내적으로 고도의 '유연성'을 확보하여야 하며 일반사용자 수준에서 시스템에 대한 일정한 기준에 의하여 업무변동이 심하지 않은 분야를 구축대상으로 삼아야 한다.

다. 적용기술

1) 개요

Client/Server 방식의 모바일 시스템으로 구축하였다. 전자 수발주 기능을 제외한 대부분의 업무가 사무실내의 LAN환경 내에서 이루어지고 네트워크 환경이 만족스럽지 못한 점을 고려하였다.

2) 하드웨어(H/W) 플랫폼

<그림 4-1> 모바일 하드웨어(H/W) 플랫폼



HP 5450

CPU : 400MHz 인텔 엑스스케일 PXA 250 프로세서
메모리 : 64MB RAM, 48MB Flash ROM
무선랜 내장 : 802.11b
OS : Microsoft Windows CE 3.0 Pocket PC 2002
CDMA2000 1x, 핸드폰 기능 지원

- 추가옵션인 CDMA2000 1x Module 장착시 무선 인터넷 및 음성통화가능
- 향후 EV-DO/DV, IMT 2000 지원예정

Wireless LAN 지원

- Wireless LAN Card 장착내장 최대 11Mbps로 Data통신가능

하드웨어 플랫폼은 모바일 응용 프로그램과 데이터베이스 등을 수용하는 구조이므로 현재 가장 범용성 있게 사용하는 HP제품 중 5450 시리즈를 표준으로 정하였다.

본 하드웨어의 표준적용에 대한 부분은 WIN CE 기반의 PPC 기반을 지원하는 표준기기를 기본적인 시스템으로 적용한다는 내용을 가지고 있다. Windows CE 는 Microsoft의 임베디드 시스템을 위한 플랫폼의 이름이다. 여기서 말하는 임베디드라는 것은 컴퓨터에 장착된 CPU처럼 컴퓨팅을 하기 위해 만들어 진 것이 아니라 특정한 기능을 지원하기 위해 제작 사용되는 전용 칩을 의미한다. 즉 PDA의 CPU뿐 아니라, 전자식 밥솥에서 기능을 관리하는 칩이라든지, 디지털 냉장고에서 그 기능을 관리하는 칩등이 여기에 해당된다.

반면 PocketPC 는 MS에서 정하고 하는 하드웨어 규격을 MS에서 테스트받은 PDA 하드웨어에 거기에 최적화된 Windows CE 을 운영체제 두가지를 가리키는 말이다.

PDA 제작자 입장에서 MS의 PDA용 운영체제는 크게 세 가지로 생각해 볼 수 있다. 그냥 OS만 구입해서 설치하여 사용하면 되는 Windows CE, 제품 개발 때부터 Microsoft의 가이드라인을 지켜 만든 후 MS의 인증을 거쳐 사용할 수 있는 PocketPC 운영체제, 또 이 PocketPC에 휴대폰에 적합한 기능이 추가 되는 PocketPC Phone Edition 으로 나뉘어 진다.

WindowsCE 와 PocketPC는 소프트웨어 사용면에서 달라진다. 흔히 프로그램 개발에 사용되는 API라는 명령어 집합이 PocketPC가 더 많기 때문이다. 따라서 Windows CE를 위해 만들어진 프로그램은 PocketPC운영체제에서 사용할 수 있지만, PocketPC를 위해 만든 프로그램은 Windows CE에서 사용할 수 없는 경우가 발생한다. 다시 말해서 PocketPC는 Windows CE에 호환되지만, Windows CE는 PocketPC에 완전히 호환되지는 않는다.

중소규모의 제작업체에서 만드는 PDA의 경우, PocketPC 인증을 받고 PocketPC 운영체제를 사용하기 위해서는 생산비용의 불가피한 증가로 인해 가격자체가 상승하기 때문에 Windows CE 운영체제를 사용하는 경우가 많다.

그러나 현재 상황에서는 PDA 프로그램이 PocketPC용으로 출시되는 경우가 많기 때문에 Windows CE를 채용한 PDA보다 PocketPC를 채용한 PDA를 구입하는 것이 합리적이라 할 수 있다.

이는 향후의 기기적인 발전의 양상이 PPC계열로의 발전 방향의 가능성이 높다는 개발의 예측이 되어 업무에 필요한 각종 어플리케이션의 지원 및 확장에 필요한 시스템적인 기반 호환성을 염두에 둔다면 이의 채용은 공통적이고 범용적인 업무를 다루는 현장업무에서 관과 해서는 안 될 부분이다(고재관, 실전 PDA 프로그래밍, 삼각형프레스).

3) 오퍼레이팅 시스템(O/S)

모바일의 운영체제는 Windows CE(이하 Win CE) 3.0 Pocket PC(이하 PP) 2003 한글버전이다.

Win CE가 처음 발표될 때의 버전1.0부터 Win CE.net인 4.0 버전까지 많은 변화를 겪어왔다. Win CE는 원래 HPC라 불리는 노트북과 유사한 형태의 운영체제였다. 이것을 지금의 PP와 같은 유형의 PDA운영체제로 만들었는데 그것이 PPC이다. 현재 본 모바일산지유통시스템의 OS기반인 PP는 Win CE 3.0을 기반으로 만들어진 확장판이다.

4) 데이터베이스(DBMS)

모바일시스템에서 필요로 하는 각종 데이터나 시스템 안에서 발생하는 각종 정보를 보관하고 종합적으로 관리하는 저장소이다. 모바일에는 모바일 전용 Access를 사용하였고 이는 NT 정보로의 이전 시 바로 재사용이 가능하다.

5) 네트워크(N/W)

모바일과 Server간을 연결하는, 즉 동일 지역 또는 원격지끼리 동일기종이나 이 기종을 막론하고 물리적인 연결을 하기 위해 사용되는 것으로 각 ERP서버의 t는 TCP/IP 프로토콜을 사용하여 전송 프로토콜을 사용하며 모바일 기기의 무선랜 방식을 권장한다.

6) 사용자 인터페이스(UI)

사용자의 응용프로그램의 접근을 원할 하게 해주는 인터페이스로 쉽게 사용자 모바일의 운영체제를 지칭한다. Windows 95/98/2000/XP/CE 등을 모두 지원한다.

7) 개발지원도구(Tool)

응용 프로그램을 구현하는데 사용되는 프로그램 도구로서 주로 데이터 모델링이나, 데이터베이스 관리, 화면 및 메뉴 개발 등의 업무에 사용된다.

가) Program Language : 임베디드 Visual C++ 3.0

8) 응용프로그램(Application) : 산지유통센터 종합관리시스템을 지칭한다.

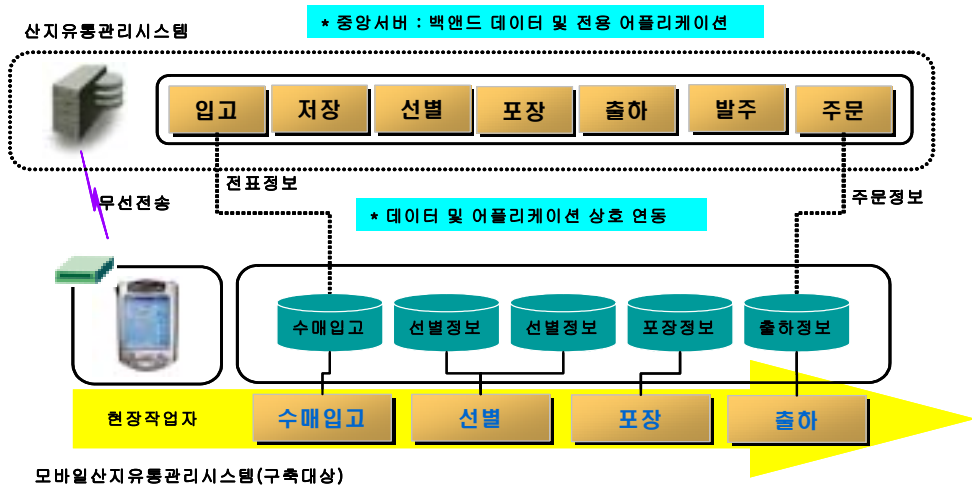
제 2 절 시스템 개요

1. 시스템 구성도

순천농협농산물산지유통센터 모바일 시스템은 사용자의 모바일 PC와 종합관리시스템 Server가 인터넷환경으로 연결되어 운용된다. Server는 향후 확장성과 안정성을 고려하여 HP5450 모델을 도입하였다. 네트워크는 서버를 축으로 하여 순천농협농산물산지유통센터의 각 부서와 사용자들은 LAN(100Mbps)으로 연결되어 있으며 서버는 인터넷전용회선이 설치되어 있다.

한편, 보안을 위하여 사용자가 접속 시에는 이중의 확인절차를 거치도록 설계되어 있다. 사용자는 우선 서버에 탑재되어 있는 DBMS(MS-SQL)의 Data Source Name(IP)와 DB에 접근할 수 있는 공인 ID 와 Password를 환경설정에 등록하여야 한다. 다음으로 시스템관리자가 개인별로 지급한 ID를 사용하여 접속한다. 개별 ID는 시스템 사용권한이 상이하므로 다른 사람의 ID로 접속할 경우 자료입력이나 보고서조회 등에 제약을 받게 된다.

<그림 4-2> 시스템 구성도



2. 구현 시스템의 기술적 특성

가. 모바일 C/S방식시스템

과거 중앙 집중식 환경 하에서는 업무담당자가 사용하고 있는 터미널은 중앙의 주전산기로부터 얻어온 정보를 뿌려주는 단순한 단말기 역할에 그쳤다. 모바일 또한 다량의 정보를 이전하는 그러한 시스템이 아닌 지원시스템의 기기적인 특성이 있으므로, 서버나 모바일 기기의 부하가 크게 영향 받는 구조는 아니다. 그러나 향후 모바일 기기의 성능 향상 및 정보 부하량의 상향에 따른 서버부하의 산출은 고려하지 않는다.

나. 4세대 언어(4GL : 4Generation Language)의 채택

개발기간의 단축, 그래픽 환경의 사용자 인터페이스 지원 등을 위하여 Visual Basic, C++, Power builder, Delphi, Java 등의 4세대 개발언어가 일반

적으로 사용되는데 순천농협농산물산지유통센터 모바일 시스템은 모바일 전용 개발 틀이면서 확장성과 재사용성이 높은 임베디드 Visual C++_으로 개발되었다.

다. 모바일전용 데이터 베이스

과거 거의 모든 시스템은 원장형 데이터베이스 구조 - 파일시스템 구조를 채택하고 있었다. 모바일 기기는 그 시스템적인 특징과 현재 모바일 시스템의 적용이 업무 지원이라는 특수성을 가지므로 기기 내부에 데이터베이스를 내장하여 자료를 보관하고 서버 데이터베이스와의 정보교환을 목적인다.

라. 객체지향기술(OOT : Object Oriented Technology)

순천농협농산물산지유통센터 모바일 시스템내의 각 모듈(프로세스)은 제각각 독립된 개체(Object)로서의 역할을 한다. BPR을 통하여 단위 업무들을 정의하고 이를 모듈화 함으로써 모바일 시스템은 수많은 모듈들의 집합체로 기능한다.

시스템이 업그레이드 되거나 기능이 추가 또는 삭제되는 경우에 전체를 건드릴 필요 없이 해당 모듈에 대한 교체만으로 시스템의 변경이 가능하다. 구매, 입고, 선별, 포장, 출하 등 각 모듈들을 서로 짜 맞추는 식으로 전체를 최적화시켜 나가면 되고, 모바일 시스템이 구축된 이후에도 언제나 단위 모듈의 변경이 가능하다.

이 같은 기술적 특성은 순천농협농산물산지유통센터와 같은 산지유통센터에 특히 중요한 요소이다. 산지유통센터가 별도의 전산 인력을 확보하지 못한 상태에서 프로그램 개발업체는 대부분 서울 등 수도권에 있어 시스템에 대한 일상적인 유지보수가 현실적으로 대단히 곤란하다. 또한, 농산물 유통환경의

변화와 이에 따른 조직과 업무의 변경 등에 따라 시스템의 기능이나 내용이 지속적으로 업그레이드되어야 하나 이 또한 여의치 않은 것이 현실이다. 실제로 많은 수의 산지유통센터가 과거 업무관리프로그램 개발에 적지 않은 비용을 투자하였으나 이후 지속적인 유지보수가 이루어지지 않고 급변하는 시장과 소비자의 요구변화에 시스템이 적절히 대응하지 못하게 되어 ‘용도폐기’한 사례가 많다.

객체지향기술을 적용한 시스템의 모듈화를 통하여 유지보수성을 향상시키고 개발(변경)기간을 단축시킬 수 있게 되어 시스템의 효용성을 제고할 수 있게 된다.

3. S/W의 설치 내역

모바일 시스템은 기본적으로 사용자 PDA에 실행프로그램 등이 존재하고 Server에는 산지유통관리시스템이 탑재되어 실행프로그램과 DB간의 연결에 의하여 작동하게 된다. - 현재 연동적용 모듈은 미개발 대상이다.

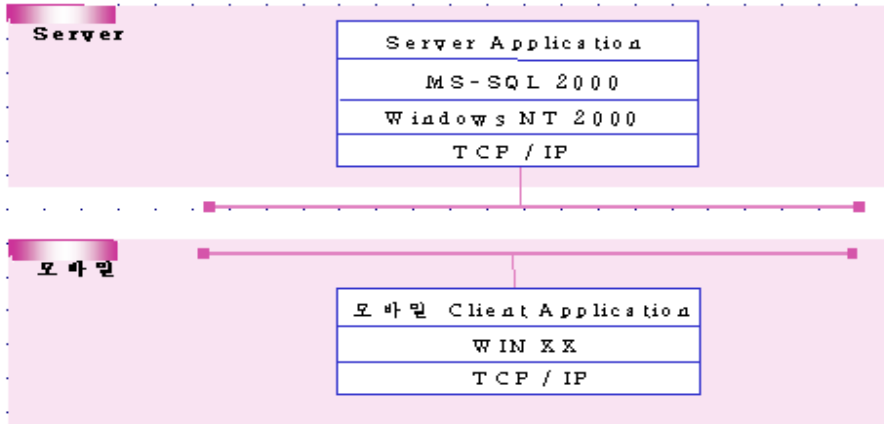
적용된 S/W를 모바일 시스템의 기술적 구조계층으로 파악하면 다음과 같다.

<표 4-5> 모바일 기술적 구조계층

모바일 구조계층	적용기술내역
응용프로그램	모바일 Application(프로그램)
개발도구	임베디드 Visual C++
사용자 인터페이스	Window CE3.0 Pocket PC 2003(한글)
네트워크	TCP/IP
데이터베이스	임베디드 Access DB
운영체제	Window CE3.0
H/W 플랫폼	Window CE3.0

실제 Client(사용자 PC)와 Server에 설치된 S/W는 다음과 같다.

<그림 4-3> 소프트웨어 구성도



가. Server - ERP 시스템

- 1) 운영체제(O/S) : MS Windows 2000 Server
- 2) DBMS : MS SQL 2000 Server
- 3) Language : Visual Basic 6.0
- 4) 개발Tool : Crystal Report 7.0(Report), Spread 3.0(Interface)
- 5) 응용프로그램 : 자동업데이트 모듈, 코드/메뉴관리모듈, 재고수불 모듈

나. 모바일시스템

- 1) 운영체제(O/S) : Windows CE 3.0 Pocket PC 2003
- 2) 응용프로그램 : 모바일 산지유통시스템

4. 시스템 운용 개념도

모바일 시스템은 다음과 같이 작동한다.

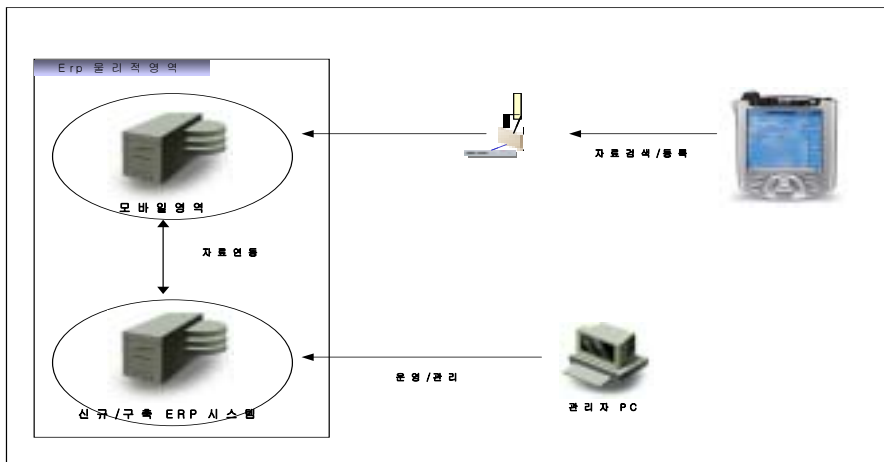
가. 사용자 모바일 PDA에 설치된 모바일 실행프로그램이 선택되면 자동으로 ADOCE3.1 데이터베이스 연결 프로토콜이 작동한다. 이후 로그인 화면이 작동하고 프로그램이 실행된다.

나. 사용자 모바일 PDA에 설치된 메인프로그램이 구동된다.

다. TCP/IP 프로토콜을 통하여 서버의 DB에 접속한다. 이때 DB를 사용할 수 있는 ID와 비밀번호를 확인한다.

라. 사용자 Log-in 과정을 거쳐 모바일 프로그램이 구동되는데, 이때 서버의 코드/메뉴관리 모듈이 사용자 ID를 체크하여 사용자별로 허용된 메뉴와 기능에 따라 제한된 화면을 제공하게 된다.

<그림 4-4> 시스템 운용개념도



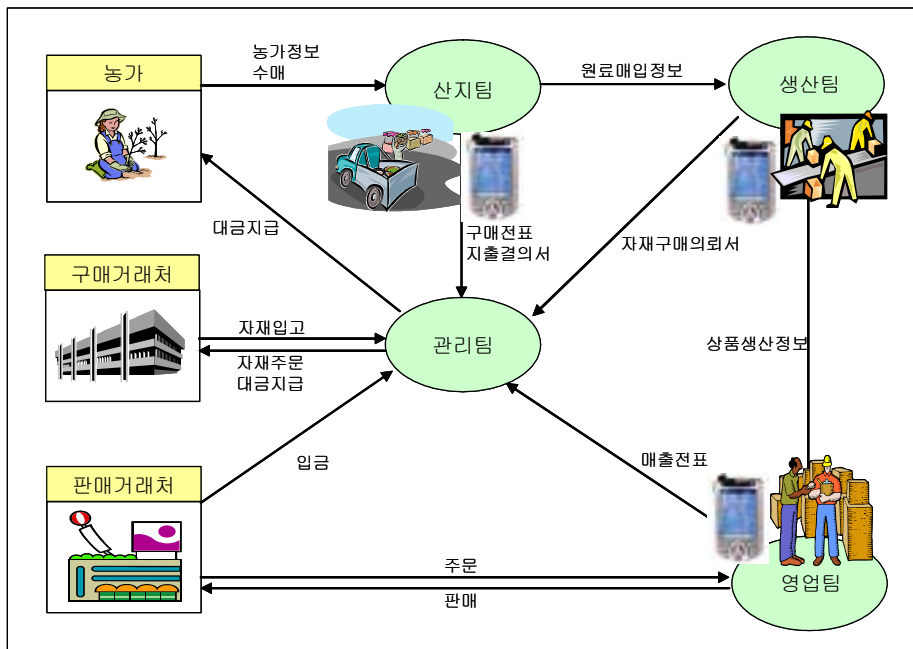
제 3 절 업무 및 시스템 분석

1. 시스템 구축을 위한 업무분석

가. 전체 업무 흐름도

기본적으로 순천농협농산물산지유통센터의 업무는 ‘농가로부터 농산물을 수매하여 소비자에게 판매하는 것’이며 이 과정에서 내부적으로 선별-포장의 작업을 수행하는 것이다. 농가와 접촉하여 농산물을 수매하는 산지팀과 수매된 농산물을 선별-포장하는 생산팀, 생산된 제품을 판매하는 영업팀, 그리고 자재구매와 관리업무를 위하여 관리팀을 두었다.

<그림 4-5> 전체 업무 흐름도

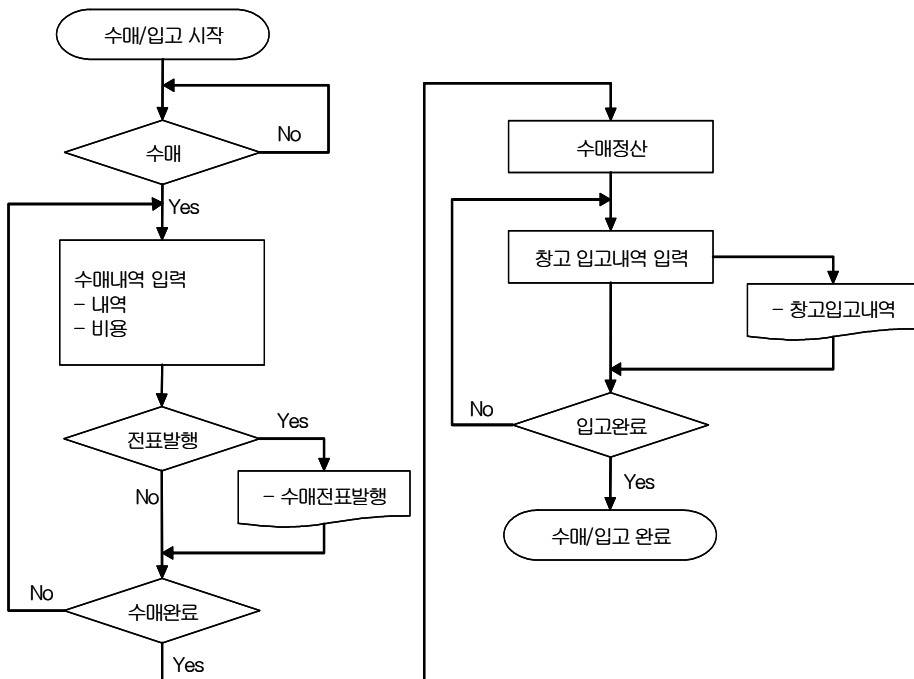


나. 절차별 업무흐름도

1) 구매/입고 업무

산지팀은 농가에서 재배하는 농산물에 대한 구매업무를 담당하고, 구매된 농산물은 창고에 입고 처리된다.

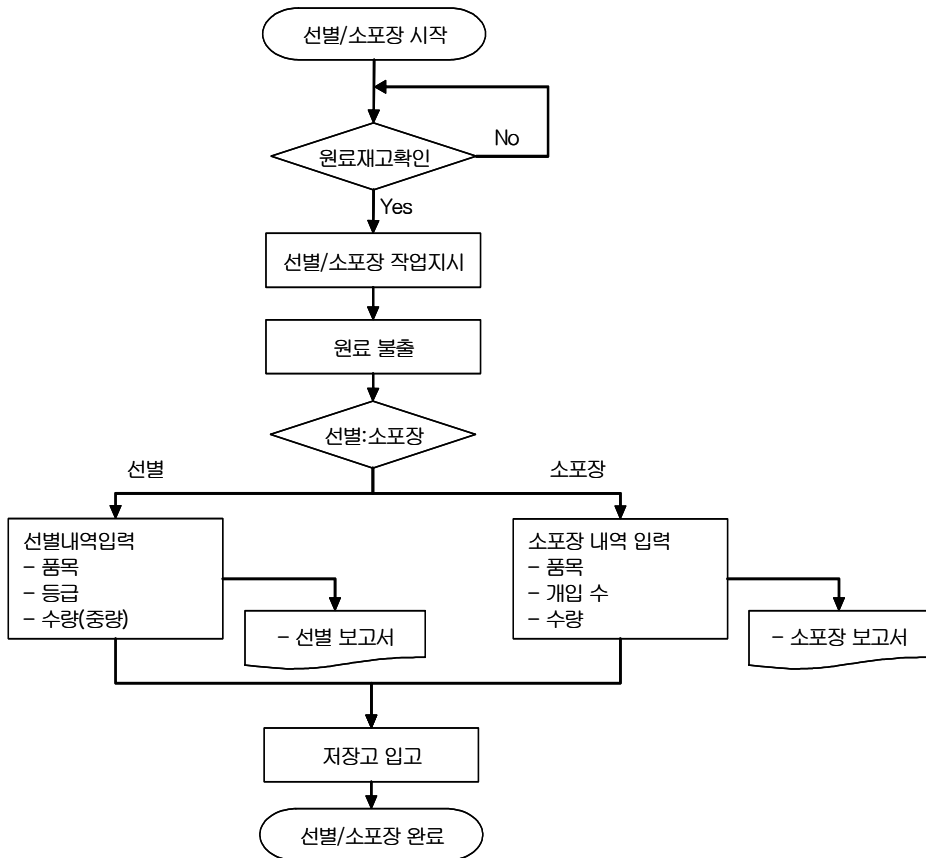
<그림 4-6> 구매/입고 업무 흐름도



2) 선별/소포장 업무

수매 입고가 종료된 데이터에 대하여 출하업무를 위해 선별/소포장 업무로 이전하여 관리한다. 선별 완료된 상품은 소포장과정을 거쳐 출하하게 된다. 소포장을 하여 주문내역에 따라 출고하게 된다.

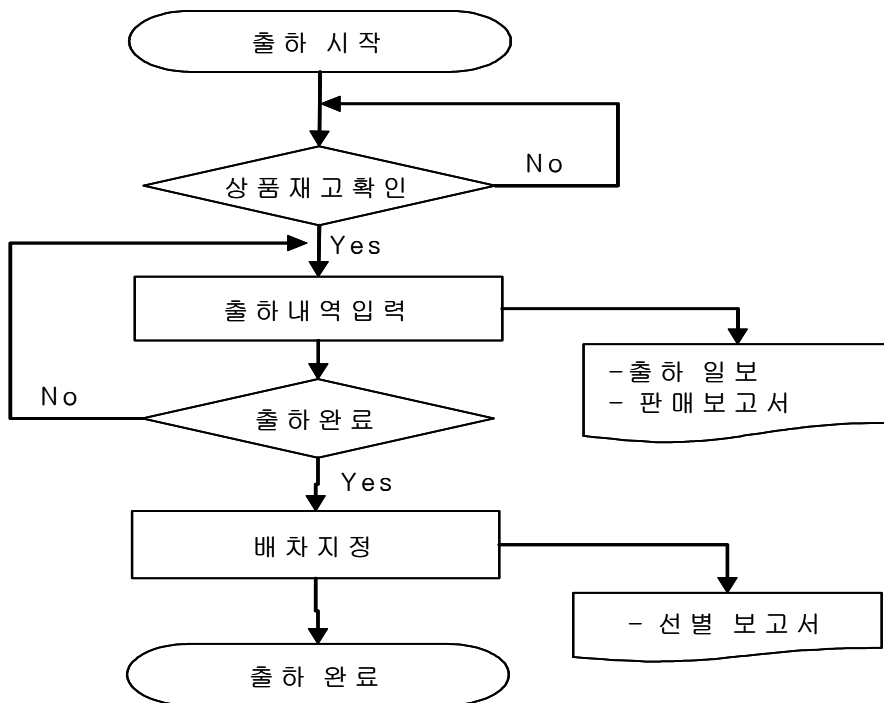
<그림 4-7> 선별/소포장 업무 흐름도



3) 출하업무

판매거래처의 주문에 따라 포장 완료된 상품이 배송 차량을 지정받아 출하한다. 이때 업무상의 편의를 제공 하는 모바일의 이용이 두드러지는 부분으로 포장 후의 정보 조회와 출하시점에서의 출하 정보 확정, 출하 취소 등을 검색하여 출하지원 업무를 병행할 수 있다.

<그림 4-8> 출하업무 흐름도



다. 현행 업무흐름의 문제점

현행 업무관리의 문제점을 살펴보면, 구매관리의 비효율성, 재고관리의 불명확성, 마케팅의 비효율성, 생산관리의 비효율성을 내포하고 있다.

1) 구매관리의 비효율성

산지거래처관리와 산지 구매에 대한 전문성이 결여된 상태에서 산지거래처 정보, 가격정보의 비공유로 인한 구매비용의 증가와 이중구매 가능성이 증대한다.

2) 재고관리의 불명확성

통일되지 않은 작업으로 인한 재고확보가 어렵고, 이에 따라 가용재고 파악의 어려움이 도출되며, 각 팀별 재고사용으로 인한 재고 Loss가 증가한다.

3) 마케팅의 비효율성

마케팅 본연의 업무에 전념하지 못하기에 신규거래처 확보 및 신규 거래품목 개발업무를 저해하며, 다품목 취급 시 업무가중이 품질저하로 이어진다.

4) 생산관리의 비효율성

정형화되지 못한 작업지시로 인해 작업 능률이 떨어지고, 긴급작업이 증가하며, 이에 따라 출하 신뢰성이 감소되고, 생산단가증대와 재작업율의 증가를 초래한다.

2. 현행 업무 흐름의 문제점 개선 방향

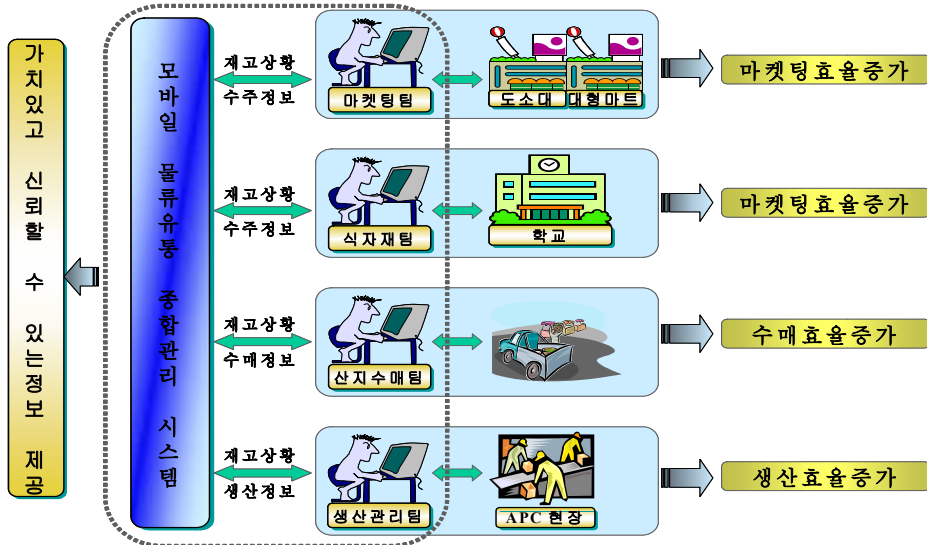
현행 업무처리상의 문제점을 개선하기 위해서 모바일 시스템을 적용하여, 위와 같은 문제점의 해결방안을 제시한다.

가. 모바일 입력을 통한 현장 구매관리를 함으로써, 현장에서의 산지거래처 정보 조회를 통해 보다 전문적이고 체계적으로 거래처를 관리하고, 실시간으로 농산물 가격정보를 조회함으로써 구매비용의 감소의 효과를 누릴 수 있다. 또한 현지에서 직접 구매정보를 입력하므로 이중구매 가능성을 줄일 수 있다.

나. 모바일을 통해 재고 조회를 실시간으로 함으로써, 가용재고 파악이 용이하고, 이에 따라 과다재고로 인한 Loss를 줄일 수 있다.

다. 모바일을 통한 체계적인 구매, 선별/포장, 재고관리, 출하관리를 함으로써, 상품에 대한 전문성을 증가시킬 수 있고, 이를 마케팅팀의 업무에 적용함으로써, 우수품질의 농산물을 높은 가격에 판매할 수 있다.

<그림 4-9> 현행 업무 개선방향



3. 요구분석정의서

가. 모바일 산지유통 시스템 요구정의

요구 사항명	모바일 시스템에 대한 적용
현재상황	<ul style="list-style-type: none"> - 모바일에 관한 업무 적용은 미비한 상황임 - 현행 타 시스템의 경우를 벤치마킹하여 업무적용 범위와 개발 범위를 산정해야함. - 초기 사용에 관한 진통적인 모습을 상당히 우려함 - 실제 현장에 대한 적용시 업무방해요인으로 작용할 우려를 보임 - 시스템과 서버간의 데이터 교류에서 나타날 혼선이 우려됨

<p>요구사항 정의</p>	<ul style="list-style-type: none"> - 입력화면은 최대한 단순화하여 처리 요망 - 자료 입력 : 수치입력은 모바일의 키패드 말고 사용자가 쉽게 입력할 키패드를 사용하여 입력 할 수 있도록 처리 요망 - 모바일 시스템의 특성상 현장에서 사용하다보면 번거롭고 사용하다 힘들 수도 있으니 가급적 현장상황을 고려한 자료처리 방안을 적용 요망 - 자료의 전송이 안 될 경우 모바일 안쪽에 저장하여 자료를 재전송할 수 있는 시스템으로 적용요망 - 버튼의 사이즈를 크게 하여 사용자가 쉽게 누르고 입력 할 수 있도록 처리 요망 - 많은 자료를 표현하지 말고 간단간단하게 현재의 상황을 알 수 있도록 처리 요망 - 무선랜에 대한 규격을 표준화하여 향후 확장시에 이상이 없도록 표준화 제시 요망 - 타 시스템과의 연계시 자료를 분리하여 구축하고 모바일로 인하여 자료의 소손을 받지 않도록 설계시 이원화 및 자료구축의 완벽한 분리 요망
<p>요구사항 반영시 제약조건</p>	<ul style="list-style-type: none"> - 개발 구조의 제약이 일부 따름.
<p>반영여부</p>	<p>반영</p>
<p>미반영 사유</p>	
<p>반영방법</p>	<ul style="list-style-type: none"> - 기능을 최소화하여 자료의 이동이 소량이 되도록 유도한다. - 모바일의 특성을 반영한 DB구조를 가지게 설계한다. - 시스템은 가급적 본 서버 ERP정보에 영향이 없도록 설계 구조를 가지고 전송 기능의 향후 추가에 따른 데이터 교환 방법을 사용한다.

나. 시스템 요구정의

요구 사항명	시스템에 대한 업무 적용
요구사항 정의	<ul style="list-style-type: none"> - 거래처정보를 관리할 수 있도록 한다. - 약정, 수매계약 내용과 수매내역의 차이발생분이나 계약이 성사가 안 되었을 때 파악할 수 있도록 구현 - 약정계약시 총중량/등급별 계약인지를 파악할 수 있도록 구현 - 감모율, 감모량, 감모종류를 파악할 수 있도록 구현 - 포장관리는 관리하기 용이한 품목에 대해서 선 수행 후 발전 방향을 모색 할 수 있도록 한다. 포장작업은 Packing List를 작업자에게 전달해서 작업을 할 수 있도록 한 후, 전달 된 Packing List로 자동으로 전산 입력 될 수 있도록 하고 포장 내역을 통한 자재 Loss 관리, 자재소요량을 관리할 수 있도록 한다. - 각 공정단위별로 원가 계산을 할 수 있도록 한다. - 창고관리를 할 수 있게 한다.
요구사항 반영시 제약조건	<ul style="list-style-type: none"> - 개발 구조의 제약이 일부 따름.
반영여부	반영
미반영 사유	
반영방법	<ul style="list-style-type: none"> - 요구사항을 반영한 DB구조를 가지게 설계한다.

4. 자료 흐름도(DFD)

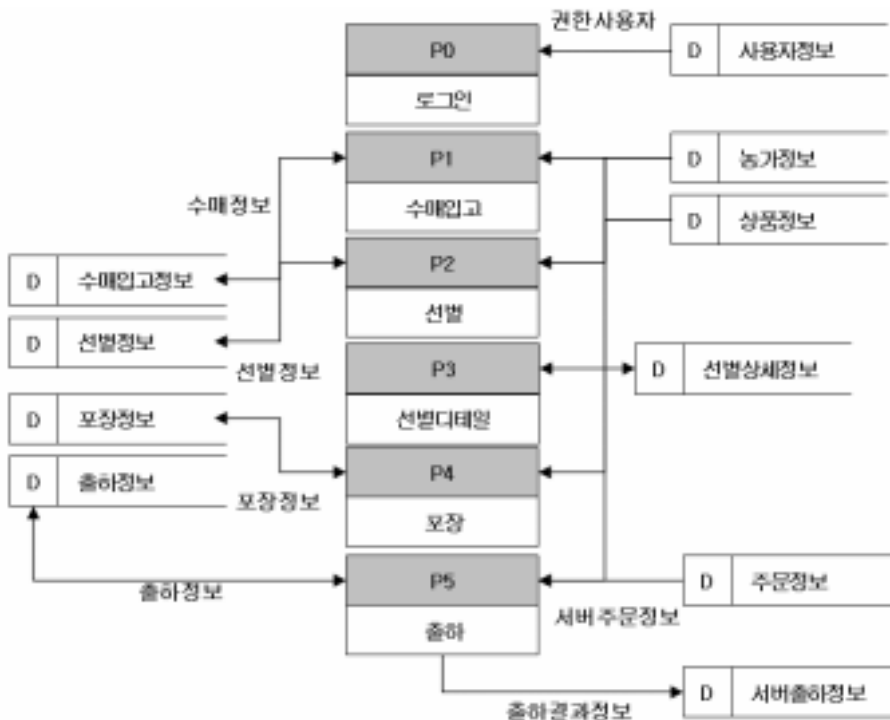
모바일 산지유통시스템은 산지유통센터에 범용적용 가능한 모델을 목표로 개발되었다. 이에 따라 코드체계, 사용자 인터페이스, 메뉴구성, 출력 및 조회 방법 등을 설계함에 있어서 편의성과 탄력성을 가장 중요하게 고려하였다.

가. 데이터 흐름도

데이터의 흐름은 공통부분과 업무흐름을 중심으로 작성 되었다.

현행 모바일 사용 업무 중 가장 업무에 적용이 가능한 부분을 중심으로 데이터의 흐름을 정의하였다.

<그림 4-10> 데이터흐름도 모바일부분 자료 흐름



제 4 절 시스템 설계

1. 논리적 엔터티 관계도(Logical ERD)

논리적 엔터티 관계도는 시스템을 구성하는 논리적인 실체(Entity)의 상호연관성을 표현한다. 본시스템의 핵심은 ERP 시스템과의 데이터적인 연결 구조를 가지면서 구매, 입고, 선별, 포장, 출하의 과정을 구현 하는 것이다.

그러나 취급하는 농산물의 종류와 특성에 따라서 실제적인 업무처리과정은 상이하게 나타나며 그것은 엔터티의 상호 연관관계를 통하여 보다 분명히 파악된다. 산지유통 시스템의 경우 단위 업무당의 운용은 크게 변하는 것은 없으나 현장의 업무 여건에 따라 적용하는 방식이 다르기 때문에 고정된 업무 방향을 제시하기에는 어렵다. 이러한 변형된 업무 적용에 다른 분야는 고려되어야 할 부분이며 향후 보다 실증적인 연구와 개발을 통하여 보완되어야 할 사항이다.

아래 그림은 모바일산지유통시스템 전체의 논리적 ERD를 보여주는 것으로 <그림 4-11> 논리적 엔터티 관계도, <그림 4-12> 물리적 엔터티 관계도를 나타낸다. 모바일 산지유통에서 가장 핵심적인 엔터티는 농가와 상품인데, 이는 본서버 ERP 시스템의 정보를 전송 받아 업데이트를 지속적으로 해야 한다. 구매, 입고, 선별, 포장, 출하까지의 과정을 진행하면서 농가와 상품의 정보를 중심으로 자료정보가 흐르며, 이후 정보의 전송에서도 이러한 기준과 각 단위의 주기를 중심으로 자료를 전송하게 된다.

2. 물리DB 설명

가. 시스템 DB table

테이블명 (한글)	테이블명 (영문)	설 명
농가정보	TM_Store	서버ERP 시스템에서 등록된 농가정보 및 사용자 정보를 가져온다
품목정보	Tbl_item	품목에 대한 공통기초 코드를 가져온다. 서버 ERP 전송
수매입고관리	Tbl_store	수매 입고에 대한 모바일 전용 정보를 관리한다
선별관리	Tbl_sel	선별에 대한 정보를 저장 관리한다
선별상세관리	Tbl_sel_detail	선별 상세 관리의 정보를 히스토리 별로 저장한다
포장관리	Tbl_pack	포장관리의 정보를 저장한다
출하관리	Tbl_ship	출하관리 정보를 저장한다

3. 물리 DB 정의서

물리 DB 정의는 본 연구과제의 개발범위에 포함된 모바일 산지유통관리시스템에 대하여 기술하였다. 논리 및 물리 ERD는 실제 개발 부분에 엔터티를 작성하였으며 물리 DB 정의는 실제 개발된 내역에 따라 기술하였다.

개별 시스템을 구성하는 DB 테이블에 대하여 실제 프로그램에서 적용된 Table ID(영문으로 표시된 테이블 명칭)와 해당 Table ID의 한글명칭을 머리

글로 하여 해당 테이블을 구성하는 각각의 속성(Attribute)에 대하여 명칭(영문/한글), Primary Key 혹은 Foreign Key로서의 역할 여부, 속성(Attribute)의 형식(Type)과 물리적인 크기를 나타내었다.

가. 모바일 산지유통관리시스템

엔티티 ID	TM_store	엔티티명	농가정보				
속성 ID	속성명	TYPE	SIZE	PK	Null	Default	설명
Farm_cd	농가코드	STR	7	PK	N		
Farm_name	농가명	STR	50		N		
Farm_type	농가구분	STR	12		Y		
Member_yn	조합원여부	STR	1		Y		
Business_no	사업자번호	STR	12		Y		
Td_no	주민번호	STR	13		Y		
Owner_name	대표자명	STR	50		Y		
Region_cd	지역코드	STR	4		Y		
Zip_no	우편번호	STR	7		Y		
Addr1	주소	STR	100		Y		
Addr2	상세주소	STR	50		Y		
Tel_no	전화번호	STR	20		Y		
Hp_no	핸드폰	STR	20				
Fax_no	팩스번호	STR	20				
Bank_cd	은행코드	STR	121				
Back_name	은행명	STR	50				
Account_no	계좌번호	STR	20				
Account_name	예금주명	STR	20				
Email	이메일	STR	50				
Joining_dt	조합원 가입일	STR	10				

Leave_dt	조합원 탈퇴일	STR	10				
Memory_cd1	기념일코드1	STR	12				
Memory_day1	기념일자1	STR	10				
Memory_cd2	기념일코드2	STR	12				
Memory_day2	기념일자2	STR	10				
Memory_cd3	기념일코드3	STR	12				
Memory_day3	기념일자3	STR	10				
User_yn	사용여부	STR	1				
RegID	등록자ID	STR	10				
Create_dt	최종생성 일자	STR	16				
Update_dt	최종수정 일자	STR	16				

엔티티 ID	Tbl_item	엔티티 명	품목정보				
속성 ID	속성명	TYPE	SIZE	PK	Null	Default	설명
Item_cd	품목코드	STR	6	PK	N		
item_nm	품목명	STR	100		Y		
item_desc	품목설명	STR	200		Y		
basic_unit	기준단위	STR	12		Y		
relation_cd	연결코드	STR	20		Y		
user_yn	사용유무	STR	1		Y		
RegID	등록자ID	STR	10		Y		
Create_dt	최종생성일자	STR	16		Y		
Update_dt	최종수정일자	STR	16		Y		

엔티티 ID	Tbl_store	엔티티 명	수매입고관리				
속성 ID	속성명	TYPE	SIZE	PK	Null	Default	설명
Buy_ser	수매일련번호	STR	4	PK	N		
Mat_opt	원료상품구분 (원료/상품)	STR	4		Y		
Buy_dt	수매일자	STR	10		Y		
Item_cd	품목코드	STR	6		Y		
Item_name	품목명	STR	100		Y		
Fram_cd	농가코드	STR	7		Y		
Fram_name	농가명	STR	50		Y		
Store_dt	입고일자	STR	10		Y		
Grade_cd	등급코드 (사용안함)	STR	4		Y		
Cont_dt	약정일자	STR	10				
Areacd	지역코드	STR	20				
Quanti	수량	INT	0				
Amt	금액	Money	0				
Kg	중량	Float	0				
RegID	등록자ID	STR	10				
Create_dt	최종생성일자	STR	16				
Update_dt	최종수정일자	STR	16				
Next_ok	Y,N	STR	1				

엔티티 ID	Tbl_sel	엔티티 명	선별관리				
속성 ID	속성명	TYPE	SIZE	PK	Null	Default	설명
Sel_ser	선별일련번호	STR	4	PK	N		
Buy_ser	수매일련번호	STR	4		Y		
Store_ser	입고일련번호	STR	4		Y		

Sort_dt	선별일자	STR	6		Y		
Fram_cd	농가코드	STR	7		Y		
Fram_name	농가명	STR	50		Y		
Item_cd	품목코드	STR	6		Y		
Item_name	품목명	STR	100		Y		
Cont_dt	약정일자	STR	10		Y		
RegID	등록자ID	STR	10		Y		
Create_dt	최초생성일자	STR	16		Y		
Update_dt	최종수정일자	STR	16		Y		
Next_ok	Y,N	STR	1		Y		

엔티티 ID	Tbl_sel_Detail	엔티티명	선별상세관리				
속성 ID	속성명	TYPE	SIZE	PK	Null	Default	설명
Sel_ser	선별일련번호	STR	4	PK	N		
Seq_detail	선별일련번호 상세번호	STR	4		Y		
Ck_cd	실적분류	STR	1		Y		0:실적,1: LOSE,2: 재입고)
Ck_name	실적분류명	STR	10		Y		(실적L OSE,재 입고)
Grade_cd	등급	STR	4		Y		
Size_cd	크기	STR	10		Y		
Kg	중량	INT	0		Y		
Store_cd	입고처	STR	6		Y		
RegID	등록자ID	STR	10		Y		
Create_dt	최초생성일자	STR	16		Y		
Update_dt	최종수정일자	STR	16		Y		

엔티티 ID	Tbl_pack	엔티티 명	포장관리				
속성 ID	속성명	TYPE	SIZE	PK	Null	Default	설명
Sel_ser	선별일련번호	STR	4	PK	N		
Pack_ser	포장일련번호	STR	4	PK	N		
Pack_dt	포장일자	STR	10		Y		
Goodcd	상품코드	STR	4		Y		
Storecd	입고처코드	STR	6		Y		
Amt	금액	int	0		Y		
Grade_cd	등급	STR	7		Y		
Quanti	수량	Int	0		Y		
Unit_qty	박스	Int	0		Y		
RegID	등록자ID	STR	6		Y		
Create_dt	최초생성일자	STR	16		Y		
Update_dt	최종수정일자	STR	16		Y		

엔티티 ID	Tbl_ship	엔티티 명	출하관리				
속성 ID	속성명	TYPE	SIZE	PK	Null	Default	설명
Order_ser	주문번호	STR	4	PK	N		
Ord_dt	주문일자	STR	10		Y		
Goodcd	상품코드	STR	6		Y		
Ord_qty	주문수량	int	0		Y		
Chul_end_opt	주문확정구분	STR	1		Y		
Pda_confirm	PDA출하확정 승인	STR	1		Y		
Chulha_dt	출하일자	STR	10		Y		
Chulha_qty	출하수량	Int	0		Y		
Chulha_type	출하형태	STR	11		Y		

Chulha_mod_dt	출하정정요청 일자	STR	10		Y		
Chulha_mod_opt	출하정정구분	STR	20		Y		
Mod_opt	정정사유	STR	20		Y		
RegID	등록자ID	STR	6		Y		
Create_dt	최초생성일자	STR	16		Y		
Update_dt	최종수정일자	STR	16		Y		

4. 물리적 엔터티 관계도(Physical ERD)

상기 ‘1. 논리적 엔터티관계도(Logical ERD)’에 표시된 엔터티에 대하여 ‘3. 물리DB 정의서’에 정의된 실제의 물리적인 상관관계를 표시하고 있다.

<그림 4-12> 물리적 엔터티 관계도

모바일산지유통시스템은 산지유통 현장에서 일어나는 ERP 시스템의 일부분을 모바일이라는 시스템으로 연동하는 구조를 가진다. 효율적인 모바일 시스템의 접근구조는 단순하면서도 현장에서 꼭 필요로 하는 업무데이터를 조회하고 빠르게 처리할 수 있는 단위 시스템의 자료를 전송하는 구조로 만들어져야 한다.

본 모바일 산지유통 시스템은 모바일시스템으로는 다소 많은 입력자료를 가지고 있는 편이지만, 현장의 업무 특성상 최소한의 정보를 입력해야하기 때문에, 정보 입력량이 한정되어 있는 모바일의 구조에서 보기에는 다소 많다고 볼 수 있다.

6. 코드 구성

가. 코드체계 및 구성내역

모바일에서 사용되는 각종 코드 정보는 본 서버의 ERP에서 적용되는 코드 체계를 따른다. 이에 본 시스템에서는 일정한 코드체계를 제시하지 않고 코드의 적용에 대한 부분만 언급한다. 아래의 내용은 산지유통의 코드 체계를 표시한 것이다.

<표 4-6> 코드구성내역

코드분류		코드명	레벨	사용유무
10	0	없음	1	Y
10	10000000000	중량단위	0	Y
10	1	g	1	Y
10	2	kg	1	Y
10	3	ton	1	Y
10	4	ml	1	Y

10	51		1	Y
20	0	SuperAdmin	1	Y
20	20000000000	사용자그룹	0	Y
20	20001	경영자	1	Y
20	20002	관리자	1	Y
20	20003	운영자	1	Y
20	20004	사용자	1	Y
20	20005	Quest	1	Y
30	30000000000	자재구분	0	Y
30	30001	저장품	1	Y
30	30002	공통	1	Y
30	30003	기타	1	Y
40	40000000000	작업자등급	0	Y
40	40001	시급	1	Y
40	40002	일당	1	Y
40	40003	조장	1	Y
40	40004	반장	1	Y
50	50000000000	기타비용	0	N
60	60000000000	출하주구분	0	Y
60	60001	일반	1	Y
60	60002	우수	1	Y
60	60003	작목반	1	Y
70	70000000000	은행	0	Y
70	70001	국민은행	1	Y
70	70002	한미은행	1	Y
70	70003	농협	1	Y
70	70004	수협	1	Y
70	70005	우체국	1	Y
70	70006	시티은행	1	Y
80	80000000000	직급코드	0	Y
80	80001	사원	1	Y
80	80002	대리	1	Y
80	80003	팀장	1	Y
80	80004	과장	1	Y
80	80005	차장	1	Y
80	80006	부장	1	Y
80	80007	부단장	1	Y
80	80008	단장	1	Y
80	80009	상임이사	1	Y

90	90000000000	부서코드	0	Y
90	90001	관리팀	1	Y
90	90002	마케팅(영업)팀	1	Y
90	90003	산지팀	1	Y
90	90004	생산팀	1	Y
90	90005	농산팀	1	Y
90	90006	지원팀	1	Y
90	90007	Ezfarm	1	Y
91	91000000000	저장고Type	0	Y
91	91001	예냉	1	Y
91	91002	저온	1	Y
91	91003	급냉	1	Y
91	91004	냉동	1	Y
91	91005	일반	1	Y
91	91006	자재	1	Y
92	0	없음	1	Y
92	92000000000	등급	0	Y
92	1	특	1	Y
92	2	상	1	Y
92	3	보통	1	Y
92	4	4등급	1	Y
92	5	5등급	1	Y
92	6	6등급	1	Y
92	7	7등급	1	Y
92	8	8등급	1	Y
92	9	등외	1	Y
93	0	없음	1	Y
93	1	1개(내, 뿌리)	1	Y
93	2	2개	1	Y
93	3	3개	1	Y
93	4	4개	1	Y
93	5	5개	1	Y
93	6	6개	1	Y
93	7	7개	1	Y
93	8	8개	1	Y
93	9	9개	1	Y
93	93000000000	크기	0	Y
93	10	10개	1	Y
93	11	11개	1	Y

93	12	12개	1	Y
93	13	13개	1	Y
93	14	14개	1	Y
93	15	15개	1	Y
93	16	16개	1	Y
93	17	17개	1	Y
93	18	18개	1	Y
93	19	19개	1	Y
93	20	20개	1	Y
93	21	21개	1	Y
93	22	22개	1	Y
93	23	30내	1	Y
93	24	40내	1	Y
93	25	50내	1	Y
93	26	60내	1	Y
93	27	70내	1	Y
93	28	80내	1	Y
93	29	90내	1	Y
93	30	100내	1	Y
93	31	110내	1	Y
93	32	120내	1	Y
93	33	130내	1	Y
93	34	140내	1	Y
93	35	150내	1	Y
93	36	160내	1	Y
93	37	170내	1	Y
93	38	180내	1	Y
93	39	190내	1	Y
93	40	200내	1	Y
93	41	210내	1	Y
93	42	220내	1	Y
93	43	250내	1	Y
93	44	300내	1	Y
93	45	350내	1	Y
93	46	400내	1	Y
93	47	450내	1	Y
93	48	500내	1	Y
93	49	500개 이상	1	Y
93	51	15내	1	Y

93	52	25내	1	Y
93	53	35내	1	Y
93	54	45내	1	Y
93	61	10내(5단위)	1	Y
93	62	20내(5단위)	1	Y
93	63	30내(5단위)	1	Y
93	64	40내(5단위)	1	Y
93	65	50내(5단위)	1	Y
93	70	113개	1	Y
93	71	88개	1	Y
93	72	72개	1	Y
93	81	특대	1	Y
93	82	대	1	Y
93	83	중	1	Y
93	84	소	1	Y
94	94000000000	수량단위	0	Y
94	94001	장	1	Y
94	Ea	개	1	Y
94	m	미터	1	Y
95	95000000000	매출처종류(판매선)	0	Y
95	95001	본지점	1	Y
95	95002	타조합	1	Y
95	95003	유통업체	1	Y
95	95004	농협유통	1	Y
95	95005	공판장	1	Y
95	95006	파머스	1	Y
95	95007	학교	1	Y
95	95008	기타	1	Y
96	96000000000	거래처 구분	0	Y
96	96001	법인	1	Y
96	96002	개인	1	Y
96	96003	회원농협	1	Y
96	96004	중앙회	1	Y
97	97000000000	기념일	0	Y
97	97001	생일	1	Y
97	97002	결혼기념일	1	Y
98	98000000000	작업코드	0	Y
98	98001	선별작업	1	Y
98	98002	전처리작업	1	N

98	98003	(소)포장작업	1	Y
98	98004	출하작업	1	Y
98	98005	수매작업	1	Y
99	99000000000	면적단위	0	Y
99	99001	ha	1	Y
99	99002	평	1	Y
100	100000000000	재배방법	0	Y
100	100001	시설	1	Y
100	100002	노지	1	Y
100	100003	기타	1	Y
101	101000000000	조합공제항목	0	Y
101	101001	출하수수료	1	Y
101	101002	수집운송비	1	Y
101	101003	선별포장비	1	Y
101	101004	포장재비용	1	Y
101	101005	보관(저장)료	1	Y
101	101006	작목안기금	1	Y
101	101007	선지급액	1	Y
101	101008	선급금상환	1	Y
101	101009	연합판매수수료	1	Y
101	101010	산지작업비	1	Y
101	101011	용기대여료	1	Y
101	101012	예냉비	1	Y
101	101013	상하차비	1	Y
101	101014	인건비	1	Y
101	101015	지점수수료	1	Y
101	101016	기타	1	Y
102	102000000000	출하처 공제항목	0	Y
102	102001	상장수수료	1	Y
102	102002	운송비	1	Y
102	102003	하역비	1	Y
102	102004	용기대여료	1	Y
102	102005	쓰레기유발금	1	Y
102	102006	기타	1	Y
103	103000000000	공제적용방법	0	Y
103	103001	kg당	1	Y
103	103002	판매금액비율	1	Y
104	0	없음	1	Y
104	1	랩	1	Y

104		2	봉지	1	Y
104		3	팩	1	Y
104		4	상자	1	Y
104		5	그물망	1	Y
104		6	벌크	1	Y
104		7	파렛트	1	Y
104		8	비닐봉지	1	Y
104		9	봉지	1	Y
104		10	접시용기 트레이	1	Y
104	104000000000		포장상태	0	Y
104		11	단	1	Y
104		12	개	1	Y
104		13	접	1	Y
104		14	채	1	Y
104		15	속	1	Y
104		16	축	1	Y
104		17	분	1	Y
104		18	캐	1	Y
104		19	포	1	Y
104		20	필름	1	Y
104		21	팩2	1	Y
104		22	팩4	1	Y
104		23	팩8	1	Y
104		24	팩10	1	Y
104		25	P-Box	1	Y
104		26	PE대	1	Y
104		27	PP대	1	Y
104		28	트럭	1	Y
105	105000000000		출하형태	0	Y
105		105001	공동	1	Y
105		105002	개별	1	Y
105		105003	시식	1	Y
105		105004	기타	1	Y
106	106000000000		출하구분	0	Y
106		106001	내부거래	1	Y
106		106002	외부거래	1	Y
106		106003	파머스거래	1	Y
106		106004	홈플러스거래	1	Y
107	107000000000		매입용도	0	Y

107	107001	파머스용	1	Y
107	107002	식자재용	1	Y
107	107003	외부거래용	1	Y
108	0	없음	1	Y
108	1	1개입	1	Y
108	2	2개입	1	Y
108	3	3개입	1	Y
108	4	4개입	1	Y
108	5	5개입	1	Y
108	6	6개입	1	Y
108	7	7개입	1	Y
108	8	8개입	1	Y
108	9	9개입	1	Y
108	10	10개입	1	Y
108	108000000000	포장개입수	0	Y
108	11	11개입	1	Y
108	12	12개입	1	Y
108	13	13개입	1	Y
108	14	14개입	1	Y
108	15	15개입	1	Y
108	16	16개입	1	Y
108	17	17개입	1	Y
108	18	18개입	1	Y
108	19	19개입	1	Y
108	20	20개입	1	Y
108	21	21개입	1	Y
108	22	22개입	1	Y
108	23	23개입	1	Y
108	24	24개입	1	Y
108	25	25개입	1	Y
108	26	26개입	1	Y
108	27	27개입	1	Y
108	28	28개입	1	Y
108	29	29개입	1	Y
108	30	30개입	1	Y
108	31	31개입	1	Y
108	32	32개입	1	Y
108	33	33개입	1	Y
108	34	34개입	1	Y

108	35	35개입	1	Y
108	36	36개입	1	Y
108	37	37개입	1	Y
108	38	38개입	1	Y
108	39	39개입	1	Y
108	40	40개입	1	Y
108	41	41개입	1	Y
108	42	42개입	1	Y
108	43	43개입	1	Y
108	44	44개입	1	Y
108	45	45개입	1	Y
108	46	46개입	1	Y
108	47	47개입	1	Y
108	48	48개입	1	Y
108	49	49개입	1	Y
108	50	50개입	1	Y
108	51	51개입	1	Y
108	52	52개입	1	Y
108	53	53개입	1	Y
108	54	54개입	1	Y
108	55	55개입	1	Y
108	56	56개입	1	Y
108	57	57개입	1	Y
108	58	58개입	1	Y
108	59	59개입	1	Y
108	60	60개입	1	Y
108	61	61개입	1	Y
108	62	62개입	1	Y
108	63	63개입	1	Y
108	64	64개입	1	Y
108	65	65개입	1	Y
108	66	66개입	1	Y
108	67	67개입	1	Y
108	68	68개입	1	Y
108	69	69개입	1	Y
108	70	70개입	1	Y
108	71	71개입	1	Y
108	72	72개입	1	Y
108	73	73개입	1	Y

108	74	74개입	1	Y
108	75	75개입	1	Y
108	76	76개입	1	Y
108	77	77개입	1	Y
108	78	78개입	1	Y
108	79	79개입	1	Y
108	80	80개입	1	Y
108	81	81개입	1	Y
108	82	82개입	1	Y
108	83	83개입	1	Y
108	84	84개입	1	Y
108	85	85개입	1	Y
108	86	86개입	1	Y
108	87	87개입	1	Y
108	88	88개입	1	Y
108	89	89개입	1	Y
108	90	90개입	1	Y
108	91	91개입	1	Y
108	92	92개입	1	Y
108	93	93개입	1	Y
108	94	94개입	1	Y
108	95	95개입	1	Y
108	96	96개입	1	Y
108	97	97개입	1	Y
108	98	98개입	1	Y
108	99	99개입	1	Y
109	109000000000	공급방법	0	Y
109	109001	문전상차도	1	Y
109	109002	도착도	1	Y
110	110000000000	공제계산방식	0	Y
110	110001	공제1(조합원)	1	Y
110	110002	공제2(비조합원)	1	Y
110	110003	공제3(공동정산)	1	Y

제 5 절 입출력 화면 개발

1. 입출력 화면정의서

업무분석과 논리 및 물리 DB설계를 기초로 모바일산지유통시스템의 입출력 화면을 설계했다. 전체 입출력화면 설계내역은 나타날 수 있는 모든 유형과 종류의 실제내역(약24분)과 동일하다. 화면설계에서 입력화면의 경우는 각각의 입력 관련 메뉴와 동일한 내역으로 작성되었으나, 출력화면(보고서)의 경우에는 기본출력화면과 출력조건과 방식을 달리 하였을 경우의 화면까지 작성하였다. 본 연구보고서에서는 입출력화면에서 서로 다른 유형과 방식 및 출력내용을 가지고 있는 주요화면에 대하여만 수록한다.

화면정의서의 구성은 다음과 같다.

- Sub-System : 화면이 해당되는 메뉴명
- Name : 화면의 명칭
- Prototype : 화면의 구성과 기능의 배치 및 입출력항목의 내역을 정의
- 관련자료정보 : 화면과 관련되는 DB 테이블의 내역을 정의한다. 또한, 해당 테이블에서 이 화면으로 특정정보를 가져오거나(Select(S)), 이 화면의 입출력 정보가 해당 테이블에 추가되거나(Insert(I)), 이 화면의 처리결과로 해당 테이블의 특정정보가 갱신되거나(Update(U)), 삭제되는(Delete(D)) 것으로 구분하여 연관관계를 정의한다.
- 화면설명 : 화면의 각종 기능과 입출력항목의 내역과 작동 Logic을 정의한다. 처리되는 데이터들의 예외처리규칙 등도 함께 기술한다.

<표 4-7> 입출력화면정의서 설명

윈도우ID		윈도우 명		관계형 테이블	
프로세스ID		다이얼로그흐름도 ID		윈도우정의	
설 명					
[처리사항] [처리Menu] [처리절차] [항목 설명]					

아래의 목록은 윈도우 화면 리스트이다. 모바일의 기본적인 화면은 싱글화면으로 구성되며 화면의 열기와 닫기는 일반적인 웹 개념의 화면 형식과 동일하다.

업무 영역	업무기능	윈도우 ID	윈도우 명
모바일 산지 유통 시스템	로그인화면	LOG_001	로그인화면 관리
	수매입고조회	SPM_001	수매 입고에 대한 화면을 지정하고 삭제를 관리
	수매입고등록/수정	SPM_002	수매입고의 등록 및 수정을 관리 한다
	선별마스터조회	SPM_003	선별마스터에 대한 화면을 지정하고 삭제를 관리
	선별마스터등록/수정	SPM_004	선별마스터에 대한 등록 / 수정을 관리 한다
	선별상세조회	SPM_005	선별상세 정보를 조회 하고 삭제 관리
	선별상세등록/수정	SPM_006	선별 상세에 대한 자료를 등록 / 수정 한다.
	포장관리조회	SPM_007	포장관리에 대한 화면을 지정하고 삭제를 관리
	포장관리등록.수정	SPM_008	포장관리에 대한 자료를 등록 / 수정 한다.
	출하관리조회	SPM_009	출하관리에 대한 조회를 하고 삭제 관리를 한다
	출하관리등록/수정	SPM_010	출하관리에 따른 자료의 등록 수정을 할 수 있다
입력기관리	SPM_100	각 단위 화면의 등록에 사용되는 키입력 사용자 화면	

--	--	--	--	--	--

윈도우ID	LOG_001	윈도우 명	PDA 로그인 화면	관계형 테이블	TM_store
프로세스ID	P0	다이얼로그 흐름도 ID	P0	윈도우정의	사용자 접근 권한을 이행한다.



설 명

[처리사항]

- 1 등록사용자를 불러와서 자신의 정보를 지정한 후 비밀번호를 입력하고 로그인한다.

[처리Menu]

- 1 로그인, 정보수정, 아이디, 비밀번호

[처리절차]

- 1 사용자 정보 및 비밀번호를 입력하고 로그인 처리한다..

[항목 설명]

- 1 아이디 : 등록 아이디를 입력 내부처리(서버와의
원패스 개념 때문에 적용 하지 않음)
- 1 비밀번호 : 패스워드 입력 - 내부처리

윈도우ID	SPM_001	윈도우 명	수매입고조회	관계형 테이블	tbl_store
프로세스ID	P1	다이얼로그 흐름도 ID	P1	윈도우정의	수매입고 조 회 화면을 정의한다



설 명

[처리사항]

- 1 등록사용자를 불러와서 자신의 정보를 지정한 후 비밀
번호를 입력하고 로그인한다.

[처리Menu]

- 1 로그인, 정보수정, 아이디,, 비밀번호

[처리절차]

1 사용자 정보 및 비밀번호를 입력하고 로그인 처리한다..

[항목 설명]

1 아이디 : 등록 아이디를 입력 내부처리(서버와의 원 패스 개념 때문에 적용 하지 않음)

1 비밀번호 : 패스워드 입력 - 내부처리

윈도우ID	SPM_002	윈도우 명	수매등록/수정	관계형 테이블	tbl_store
프로세스ID	P1	다이아로그 흐름도 ID	P1	윈도우정의	수매입고 등록/수정화면을 정의한다

APPC 모바일 유통관리시스템

수매입고번호	0001		
원료/상품구분	<input checked="" type="radio"/> 원료 <input type="radio"/> 상품		
수매일자	2004-08-15		
품목	수박 ▼		
농가	밀성농장 ▼		
입고일	2004-08-19	약정일	2004-08-19
지역	34	수량	34 개
금액	60,000 원	중량	60 Kg

등록
뒤로

설 명

[처리사항]

1 등록/수정하고자 하는 자료가 정확한지를 검토한다.

[처리Menu]

1 등록

[처리절차]

1 각 항목의 정보를 기입한다.

1 조건 입력에 따르는 정보는 콤보 박스를 클릭하고 입력 자료는 키패드의 화면을 이용하여 입력한다.

1 입력된 정보를 등록버튼 클릭하면 자료는 저장된다.

윈도우ID	SPM_003	윈도우 명	선별조회	관계형 테이블	tbl_sel
프로세스ID	P2	다이얼로그 흐름도 ID	P2	윈도우정의	선별관리화면을 정의한다

ARPC 모바일 유통관리시스템

선별일련번호	0008
--------	------

수매번호	구분	일자	품목	농가명
0001	상품	08.15	수박	호두농장
0002	0002	08.26	다래	밀성농장

입고일	2004-08-19
약정일	2004-08-19

등록	뒤로
----	----

설 명

[처리사항]
 1 선별관리에 대한 자료를 조회한다.
 [처리Menu]
 1 조회
 [처리절차]
 1 조회 하고자 하는 조건을 기입한다.
 1 조회 버튼을 클릭 하여 현황 리스트를 그리드에 나타낸다.
 1 선별현황 현황리스트를 클릭 하고 등록/수정 버튼을 클릭하면 해당 화면으로 이동한다.

윈도우ID	SPM_004	윈도우 명	선별등록/수정	관계형 테이블	tbl_sel
프로세스ID	P2	다이얼로그 흐름도 ID	P2	윈도우정의	선별등록/수정화면을 정의한다

ARPC 모바일 유통관리시스템

선별일련번호	0008
--------	------

수매번호	구분	일자	품목	농가명
0001	상품	08.15	수박	호두농장
0002	0002	08.26	다래	밀성농장

입고일	2004-08-19
약정일	2004-08-19

등록

뒤로

설 명	
[처리사항]	1 등록/수정하고자 하는 자료가 정확한지를 검토한다.
[처리Menu]	1 등록
[처리절차]	1 각 항목의 정보를 기입한다. 1 조건 입력에 따르는 정보는 콤보 박스를 클릭하고 입력 자료는 키패드의 화면을 이용하여 입력 한다. 1 입력된 정보를 등록버튼 클릭하면 자료는 저장 된다

2. 유통관리시스템 화면 개발

가. 시스템 접속 및 사용자확인

모바일산지유통관리시스템은 사용자의 모바일 시스템과 ERP Server가 인터넷 환경으로 연결된 상태에서 종합사용이 가능하며 로컬 모바일용으로 자료의 운용은 가능한 시스템이다. 그러나 자료코드의 전송 및 정보확인 등의 흐름업무에 대한 부분은 단독으로 사용은 불가능하다. 사용자 모바일에는 시스템에 접속할 수 있는 실행파일이 설치되는데 여기에는 화면 및 보고서 출력과 관련된 각종 드라이버와 파일 등이 함께 설치된다. 사용자번호(ID)는 사용자가 임의로 설정하지 않고 시스템관리자가 부서와 업무에 따라 지정하며 각 사용자에게는 자신의 권한에 따라 입력과 조회가 가능한 화면이 제한적으로 제공된다. 본 구축시스템에서는 이러한 제한적인 시스템 권한은 구축하지 않았다.

먼저, 시스템에 접속하여 사용자확인을 수행하는 과정은 다음과 같다.

- 1) 설치된 실행파일(Client.exe)을 구동시킨다.
- 2) 시스템은 서버에 접속하여 사용자 모바일과 서버 프로그램의 버전

윈도우ID	SPM_005	윈도우 명	선별상세조회	관계형 테이블	tbl_sel_detail
프로세스ID	P3	다이얼로그 흐름도 ID	P3	윈도우정의	선별상세관리화면을 정의 한다

ARPC 모바일 유통관리시스템				
선별상세마스터관리		장진출		
농가	길농장 ▼	품목	수박 ▼	
선별 코드	수매 번호	선별일	품목	농가명
0001	0001	08.15	수박	호두농장
0002	0002	08.26	다래	밀성농장
0003	0006	08.10	배추	가물마을

조회 추가 수정 삭제 메뉴 ?

설 명

[처리사항]

1 수매입고에 대한 자료를 조회한다.

[처리Menu]

1 조회

[처리절차]

1 조회 하고자 하는 조건을 기입한다.

1 조회 버튼을 클릭 하여 현황 리스트를 그리드에 나타낸다.

1 수매입고현황 현황리스트를 클릭 하고 등록/수정 버튼을 클릭하면 해당 화면으로 이동한다

윈도우ID	SPM_006	윈도우 명	선별상세등록/수정	관계형 테이블	tbl_sel_detail
프로세스ID	P3	다이어로그 흐름도 ID	P3	윈도우정의	선별상세등록/수정화면을 정의한다

ARPC 모바일 유통관리시스템

선별일련번호: 0001

선별	상세	실적	중량	비고

등급: 34

크기: 34 LOSS: 34

실적: 34 입고처: 34

설 명

[처리사항]
 1 등록/수정하고자 하는 자료가 정확한지를 검토한다.

[처리Menu]
 1 등록

[처리절차]
 1 각 항목의 정보를 기입한다.
 1 조건 입력에 따르는 정보는 콤보 박스를 클릭하고 입력
 자료는 키패드의 화면을 이용하여 입력한다.
 1 입력된 정보를 등록버튼 클릭하면 자료는 저장 된다

(version)을 점검하여 프로그램의 업그레이드 또는 변경사항을 확인한 후 전송여부를 실행 하게 되는데 현재의 시스템에서는 이러한 전송 기능은 모 서버의 미구축으로 인하여 전송규약에 관한 모듈은 정의하지 않는다.

- 3) 로그인화면에서는 로컬사용자 정보를 기준으로 데이터베이스로의 접근을 확인한다.
- 4) 관리자로부터 부여받은 사용자 아이디와 비밀번호를 입력하여 시스템에 로그인 한다.
- 5) 로그인 정보는 모바일시스템의 사용에 대한 정보로 활용되면서 정보의 등록 및 수정에 대한 로그정보로 저장되어 이후의 정보에 대한 자료 추적에 활용된다.

<그림 4-14> 모바일 로그인 화면



이때 사용자 아이디는 모바일 사용자가 직접 지정하는데 현재는 모바일의 내부 정보로 고정된 상태이다.

윈도우ID	SPM_007	윈도우 명	포장관리 조회	관계형 테이블	tbl_pack
프로세스ID	P4	다이얼로그 흐름도 ID	P4	윈도우정의	포장관리화면을 정의한다

ARPC 모바일 유통관리시스템				
포장관리			장진용	
포장	선별	포장 일자	상품 코드	입고처
0001	0012	08.15	1002	100002

조회 추가 수정 삭제 메뉴 ?

설 명

[처리사항]

1 수매입고에 대한 자료를 조회한다.

[처리Menu]

1 조회

[처리절차]

1 조회 하고자 하는 조건을 기입한다.

1 조회 버튼을 클릭 하여 현황 리스트를 그리드에 나타낸다.

1 수매입고현황 현황리스트를 클릭 하고 등록/수정 버튼을 클릭하면 해당 화면으로 이동한다.

나. 코드 및 기준정보

코드 등록은 현재 모바일에서 이루어지지 않는다. 모바일의 시스템 운용상 로컬로 관리할 수 있는 정보의 양이 한정적이므로 코드는 본 서버 ERP의 코드정보와 규약을 따르도록 한다. 이에 정보코드는 ERP 시스템 별로 이루어지는 규격을 현구축 시스템에 수정 적용 하여야 한다.

이같이 코드체계를 모바일 기기에서는 관리하지 않고 서버ERP에서 관리할 경우 이 시스템이 단순히 한 소재의 ERP시스템이 아닌 다른 산지유통센터에 적용되더라도 기본적인 코드체계를 변경하지 않고 적용될 수 있다는 점과 유통환경의 변화 등으로 빈번하게 발생할 수 있는 코드체계의 변경요구에 탄력적으로 대응할 수 있는 장점이 있어 유지보수나 재개발 등 시스템 사용을 어렵게 만드는 제약요인을 제거하는 효과도 발생한다.

다. 수매입고정보

<그림 4-15> 수매입고 조회화면

모바일유통관리시스템의 가장 큰 특징 중의 하나는 모바일 기능을 극대화하여 업무간 연계성이 최대한 확보되는 입력/조회화면에 있다. 위의 화면은 농가별과 품목별로 바로 수매내역을 조회하고 입력하는 화면으로 상단 콤보창에서 수매입고관리가 필요한 농가와 품목을 선택하면 출력되는 기본화면이다.

여기서는 우선 모바일이라는 특수성을 인식하고 불필요한 정보의 표현을 배제한 다음, 수매구분, 수매일자, 품목, 농가명을 바로 인식하고 수매가 일어나

윈도우ID	SPM_00	형분	tbl_pack
프로세스ID	P4	정의	포장등록/수정 화면을 정의한다
설 명			
<p>[처리사항] 1 등록/수정하고자 하는 자료가 정확한지를 검토한다.</p> <p>[처리Menu] 1 등록</p> <p>[처리절차] 1 각 항목의 정보를 기입한다. 1 조건 입력에 따르는 정보는 콤보 박스를 클릭하고 입력 자료는 키패드의 화면을 이용하여 입력한다. 1 입력된 정보를 등록버튼 클릭하면 자료는 저장된다.</p>			

윈도우ID	SPM_009	윈도우 명	출하관리조회	관계형 테이블	tbl_ship
프로세스ID	P5	다이얼로그 흐름도 ID	P5	윈도우정의	출하관리화면을 정의한다



설 명

[처리사항]

1 수매입고에 대한 자료를 조회한다.

[처리Menu]

1 조회

[처리절차]

1 조회 하고자 하는 조건을 기입한다.

1 조회 버튼을 클릭 하여 현황 리스트를 그리드에 나타낸다.

1 수매입고현황 현황리스트를 클릭 하고 등록/수정 버튼을 클릭하면 해당 화면으로 이동한다

윈도우ID	SPM_010	윈도우 명	출하등록/ 수정	관계형 테이블	tbl_ship
프로세스ID	P5	다이얼로그 흐름도 ID	P5	윈도우정의	출하등록/수정화면 을 정의 한다

ARPC 모바일 유통관리시스템

주문번호	0001
주문일자	
상품코드	
주문수량	
주문확정구분	<input checked="" type="radio"/> 확정 <input type="radio"/> 미확정
POA출하확정	<input checked="" type="radio"/> 확정 <input type="radio"/> 미확정
출하일자	
출하수량	<input type="text"/>
출하형태	<input type="text"/>
출하정경요청일자	

등록
뒤로

설 명

[처리사항]

1 등록/수정하고자 하는 자료가 정확한지를 검토한다.

[처리Menu]

1 등록

[처리절차]

1 각 항목의 정보를 기입한다.

1 조건 입력에 따르는 정보는 콤보 박스를 클릭하고 입력
자료는 키패드의 화면을 이용하여 입력한다.

1 입력된 정보를 등록버튼 클릭하면 자료는 저장 된다

는 즉시 바로 모바일로 정보를 저장할 수 있는 구조를 가지고 있고 정보를 즉시 조회 할 수 있다.

신규 작업입력은 화면 하단 우측에 추가버튼을 통하여 입력 모드로 전환되며 자료를 입력한다. 조회창은 엑셀(Excel)과 같은 스프레드 시트 방식을 사용하여 사용자가 한눈에 정보를 요약된 모습으로 보이도록 편의성을 높였다. 등록, 수정, 삭제를 바로 한자리에서 이동하게 하여 좁은 화면에서 다른 정보와 혼선이 없도록 시스템을 구현하였다.

라. 주문 및 출하내역 입력

모바일산지유통시스템의 가장 핵심적인 부분이 '주문'과 '출하'이다. 포장에서 출하로 전환되어지는 경우에 출하정보는 언제나 주문전표를 확인하고 이행하는 작업이 상당히 까다로운 작업 중의 하나였다.

모바일의 공간제약성의 극복은 출하시에 주문정보를 즉시 조회하고 주문에 따른 출하승인 및 출하에 따른 정보를 저장하고 서버로 전송하여 한자리에서 모든 출하 공정을 마감할 수 있다는 것이다. 출하시에 나타나는 주문의 취소 및 주문 확인 등을 빠르게 처리할 수 있다.

현재 모바일 시스템의 이용 측면에서 보면 이러한 출하시스템과 확인시스템이 시스템 구축의 주류를 이룬다.

주문과 출하업무시 비효율적인 처리과정을 개선함으로써 인력과 시간 및 비용을 절감하고 업무 생산성을 제고하는데 초점을 두었다.

주문출하시스템에서의 확장은 재고관리에 있다고 볼 수 있다. 재고 조사를 통한 주문과 출하에 대한 현장 판단을 도울 수 있다. 이러한 현장 의사결정에서의 모바일시스템의 활용은 상당한 효과를 기대할 수 있다고 본다.

<그림 4-16> 모바일 산지유통 출하관리 화면

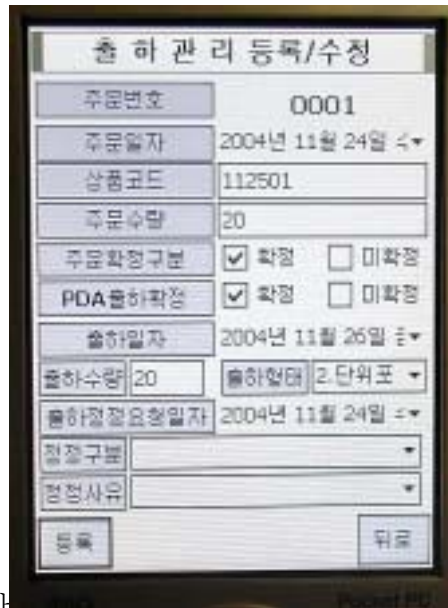
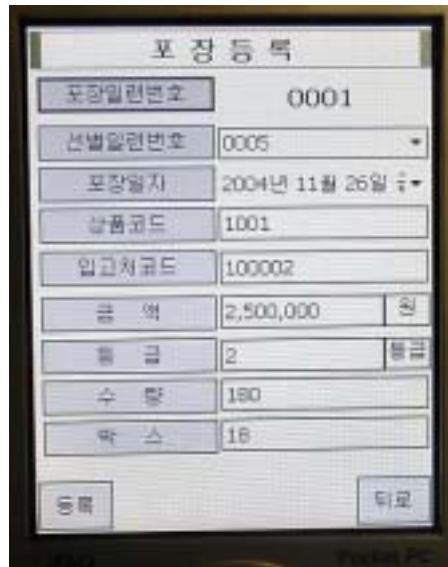


그림 1-17 모바일 산지유통시스템의 출하관리 화면



모바일 산지유통시스템의 포장관리는 선별정보를 포장 쪽에서 어떤 방식으로 관리할 것인가에 상당한 어려움이 있었다.

포장업무의 흐름을 방해하는 요인을 제거하고 빠른 정보 접근을 위한 입력 부분의 단순화 및 입력 항목의 최소화를 목표로 하여 상기와 같은 화면구조를 가지게 되었다. 포장업무 면에서 본다면 다소 항목이 부족할 수 있는 입력 항목수 이지만, 반드시 필요한 정보만을 화면에 배치하였다.

바. 선별 및 선별 상세관리

선별 및 선별 상세 관리 화면구조는 상기의 포장, 출하의 화면구조와 거의 흡사한 정보화면 구조를 가지고 있다. 정보 화면의 일관성을 유지하고 화면 운용에 따른 사용자의 혼란을 방지하면서, 모바일 운용이라는 기기의 사용에 대한 위험요소를 최소화 하도록 구조 모형을 단순화하였다.

선별과 선별 상세는 상당히 많은 량의 정보 입력 구조를 가지고 있던 모형으로, 선별에 대한 항목을 분리하기 어렵기 때문에 선별 마스터와 선별 상세 시스템이라는 구조로 분리하여 구축 하였다.

<그림 4-18> 모바일 산지유통 선별마스터관리 화면

선별	수량	선별일	품목명	농가명
0001	0001	09.10	포도	호두농장
0002	0002	09.10	포도	한대농장
0003	0006	09.10	포도	호두농장
0004	0005	11.24	복수수	별지농장
0006	0004	11.27	자두	한국농...
0007	0003	11.26	참외	호두농장

<그림 4-19> 선별마스터, 선별상세 등록/수정 화면



사. 자료 입력방식

모바일의 자료입력 방식은 모바일 기기의 표준 입력화면으로는 입력에 대한 편의성을 제공하기 어렵다. 빠른 입력 및 기기의 운용에 따른 입력화면의 구축은 필수적인 요소이다. 모바일의 특성상 다량의 정보 입력은 불가능한 아키텍처 구조를 가지므로 하단의 그림과 같은 입력 창을 구축하여 모듈화하였다.

각종 정보의 입력을 빠르게 실행 하고, 입력을 최소화 할 수 있는 모듈로써 응용에 따라서는 여러 가지의 방식으로 구축을 변형할 수 있다.

<그림 4-20> 자료 입력용 키패드 구축화면



모바일시스템의 운용과 적용 방식은 다양한 입력기기 및 입력 장치에 의해서 기능이 현재 업그레이드되고 있는 상황이나 입력화면 (300x200)의 제약성으로 인하여 정보운용은 그다지 크다고 볼 수 없다. 그러나 이와 같은 표현시스템 및 입력 시스템의 개발로 자료 표현의 한계를 극복하는 것이 모바일 정보의 운용에 성공 요인이다.

제 6 절 시스템 파일의 구성

모바일 산지유통시스템의 파일구성은 모바일 부분과 관리자 시스템 부분으로 나누어진다. 모바일 부분은 모바일에 탑재된 사용자 시스템의 운용 및 자료 입력, 조회부분으로 이루어지고 관리자시스템은 모바일 코드 전송 및 정보추출 후 ERP 모듈로의 전송 등이 주요 기능이다. 본 과제에서는 모바일 부분을 개발하였고, 농가정보, 품목정보, 수매입고, 선별, 포장, 출하관리의 테이블로 구성했다.

1. 시스템 실행화일

모바일 산지유통시스템은 시스템 실행 시, 우선 모바일에 설치된 실행파일을 실행하여 Log-in 과정을 거쳐 실행되는 과정으로 운용된다.

모바일의 실행 프로그램은 기기의 다크먼트 폴더에 저장하여 시스템의 배터리 방전에 따른 정보의 손실을 방지하였다.

2. 모바일 파일(Module)

가. 업무관리

<표 4-8> 모바일 산지유통 프로그램 목록

프로그램명	기능	내용
MOBILE_KEYBOARD	모바일 키보드 구축 모듈	전체 모바일시스템 운용에 필요한 자료 입력기로 객체화시킨 모듈을 공통으로 사용하도록 하였다.
siw_Q(T)0001	수매입고 관리	수매입고에 대한 정보를 입력하고 수정, 삭제, 조회하는 화면으로 수매입고 항목 중

프로그램명	기능	내용
SEM_Q(T) 0001	선별 관리	선별관리는 수매입고에서 넘어온 자료를 대상으로 선별 일자와 약정일자를 입력하고 대상 자료만 선별로 등록 하고 자료의 형식은 수매입고의 정보를 선별 테이블로 등록한다.
SED_Q(T) 0001	선별 상세 관리	선별상세관리는 현재 선별로 생성된 마스터를 기준으로 실적분류를 입력하는 형식으로 대상 정보에는 등급, 크기, 실적, LOSS, 입고처 등을 주요 대상으로 한다.
PAC_Q(T) 0001	포장 관리	포장관리 정보는 포장업무 진행시 단위업무를 기준으로 정보를 생성하는 형식으로 주요 입력 항목으로는 선별일련번호,포장일자,상품코드,입고처번호,금액,등급,수량,박스 등이 주요 항목이다
SHP_Q(T) 0001	출하 관리	출하관리는 서버와의 주문내역을 매치하여 정보를 입력하여야 한다. 출하 등록시 본 서버 ERP의 주문내역을 확인하는 정보를 가지고 오거나 일괄적으로 출하주문정보를 우선 저장하는 방식으로 해야 하나 현 시스템에서는 대상 ERP가 현재 명확한 주문구조를 가지고 있는 상황이 아니므로 출하정보관리의 단위 출하만 등록 하는 방식으로 시스템을 구축하였다. 아울러 출하에 대한 PDA 확정이라는 모바일 출하 개념을 두어 ERP로 정보를 전송할 경우 발생하는 출하정보의 혼선을 미연에 방지 하였다. 주요 입력 대상으로 주문일자,상품코드,주문수량,주문확정구분, PDA 출하확정, 출하일자, 출하수량, 출하형태, 출하정정요청일자, 정정구분, 정정사유 등을 들 수 있다.

3. Module 소스

모바일의 구축된 소스는 현 업무 외에도 타 업무 기반에 재사용되어 질 수 있도록 모듈화 하여 작업 하였다. 모듈은 그 구조가 확장성을 고려하여 개발 되어 졌기에 모듈의 분석에 의거하여 산비유통의 다른 모듈을 구축하는데 용이 하도록 구성 하였다.

가) 모바일 키보드 모듈

```
// MOBILE_KEYBOARD.cpp : implementation file
//
#include "stdafx.h"
#include "resource.h"
#include "MOBILE_KEYBOARD.h"
#include "exfarmDlg.h"

#ifdef _DEBUG
#define new DEBUG_NEW
#undef THIS_FILE
static char THIS_FILE[] = __FILE__;
#endif

extern int      KeyBoardOK ; //(0:패스워드,1:금액,수량)
extern HBRUSH brush ;
extern CFont font;

extern CString  strKEY      ; // 키보드 입력값
```

```

extern CString    strKEYTITLE ; // 키보드입력 타이틀
extern int        keyYS      ; // 키보드입력후 1확인/0취소
extern int        keyLength  ; // 키보드입력길이

CString oldKEY1          ; // 이전포커스값1
CString ChkPoint        ; // 키보드 입력포인트
int    imoney1          ; // 키입력값1
CString smoney1,oldKEY   ; // 입력값 금액을 스트링을 변환1
////////////////////////////////////
// CMOBILE_KEYBOARD dialog
CString KEYComma(int dNum)
{
    int i, j;
    CString strTmp;
    CString strNum;
    strTmp.Format(_T("%d"), dNum);
    for(i = strTmp.GetLength() - 1, j = 1; i >= 0; i--, j++)
    {
        if( ((j % 3) == 1) && (j > 3) )
        {
            strNum.Insert(0, ',');
            strNum.Insert(0, strTmp.GetAt(i));
        }
        else
        {
            strNum.Insert(0, strTmp.GetAt(i));
        }
    }
}

```

```

    }
    return strNum;
}

CMOBILE_KEYBOARD::CMOBILE_KEYBOARD(CWnd* pParent /*=NULL*/)
    : CDialog(CMOBILE_KEYBOARD::IDD, pParent)
{
   //{{AFX_DATA_INIT(CMOBILE_KEYBOARD)
    m_oldkey = _T("");
    m_keytitle = _T("");
    //}}AFX_DATA_INIT
}

// 화면을 풀스크린으로 만든다.
void CMOBILE_KEYBOARD::Full_ScreenOK()
{
    // START 화면을 전체화면으로 키운다.
    SHINTDLGINFO shidi;
    shidi.dwMask = SHIDIM_FLAGS;
    shidi.dwFlags = SHIDIF_FULLSCREENNOMENUBAR | SHIDIF_SIPDOWN;
    shidi.hDlg = m_hWnd;
    SHInitDialog(&shidi);
    ::CommandBar_Show(m_pWndEmptyCB->m_hWnd, FALSE);
    SetForegroundWindow();
    ::SHFullScreen(m_hWnd, SHFS_HIDETASKBAR | SHFS_HIDESIPBUTTON |
SHFS_HIDESTARTICON);
    MoveWindow(0, 0, 240, 360, TRUE);
}

```

```

// END 화면을 전체화면으로 키운다.

}

void CMOBILE_KEYBOARD::DoDataExchange(CDataExchange* pDX)
{
    CDialog::DoDataExchange(pDX);
   //{{AFX_DATA_MAP(CMOBILE_KEYBOARD)
    DDX_Control(pDX, IDC_OK_BTN, c_ok_btn);
    DDX_Control(pDX, IDC_CLEAR_BTN, c_clear_btn);
    DDX_Control(pDX, IDC_CANCLE_BTN, c_close_btn);
    DDX_Control(pDX, IDC_BACKBTN, c_backbtn);
    DDX_Text(pDX, IDC_KEYNO, m_oldkey);
    DDX_Text(pDX, IDC_KEYTITLE, m_keytitle);
   //}}AFX_DATA_MAP
}

BEGIN_MESSAGE_MAP(CMOBILE_KEYBOARD, CDialog)
   //{{AFX_MSG_MAP(CMOBILE_KEYBOARD)
    ON_BN_CLICKED(IDC_KEY7, OnKey7)
    ON_BN_CLICKED(IDC_KEY8, OnKey8)
    ON_BN_CLICKED(IDC_KEY9, OnKey9)
    ON_BN_CLICKED(IDC_KEY4, OnKey4)
    ON_BN_CLICKED(IDC_KEY5, OnKey5)
    ON_BN_CLICKED(IDC_KEY6, OnKey6)
    ON_BN_CLICKED(IDC_KEY1, OnKey1)
    ON_BN_CLICKED(IDC_KEY2, OnKey2)

```



```

ON_BN_CLICKED(IDC_KEY3, OnKey3)
ON_BN_CLICKED(IDC_KEY0, OnKey0)
ON_BN_CLICKED(IDC_KEY00, OnKey00)
ON_BN_CLICKED(IDC_KEY, OnKey)
ON_BN_CLICKED(IDC_BACKBTN, OnBackbtn)
ON_BN_CLICKED(IDC_CANCLE_BTN, OnCancleBtn)
ON_BN_CLICKED(IDC_CLEAR_BTN, OnClearBtn)
ON_BN_CLICKED(IDC_OK_BTN, OnOkBtn)
//}}AFX_MSG_MAP
END_MESSAGE_MAP()

////////////////////////////////////
// CMOBILE_KEYBOARD message handlers
BOOL CMOBILE_KEYBOARD::OnInitDialog()
{
    CDialog::OnInitDialog();
    CenterWindow(GetDesktopWindow());    // center to the hpc screen

MoveWindow(0,0,::GetSystemMetrics(SM_CXSCREEN),::GetSystemMetrics(SM_CYSCREEN));

    // START 화면을 전체화면으로 키운다..
    SHINTDLGINFO shidi;
    shidi.dwMask = SHIDIM_FLAGS;
    shidi.dwFlags = SHIDIF_FULLSCREENNOMENUBAR | SHIDIF_SIPDOWN;
    shidi.hDlg = m_hWnd;
    SHInitDialog(&shidi);
    ::CommandBar_Show(m_pWndEmptyCB->m_hWnd, FALSE);
    SetForegroundWindow();

```

```

::SHFullScreen(m_hWnd, SHFS_HIDETASKBAR | SHFS_HIDESIPBUTTON |
SHFS_HIDESTARTICON);

MoveWindow(0, 0, 240, 360, TRUE);

// END 화면을 전체화면으로 키운다..

UpdateData(true);

oldKEY = strKEY ;
strKEY = strKEY ;
m_keytitle = strKEYTITLE ;

// 이전입력값 //
smoney1 = oldKEY ;
imoney1 = _wtoi(smoney1) ;
if(KeyBoardOK == 0)
{
    m_oldkey = smoney1 ;
}
else
{
    m_oldkey = KEYComma(imoney1) ;
}

// 이전입력값 //
// 화면상의 글자크기 확대 //
font.CreateFont(24, 0, 0, 0, FW_BOLD, FALSE, FALSE, 0, 0, 0, 0, 0, 0, _T("굴림"));
((CStatic*)this->GetDlgItem(IDC_KEYNO))->SetFont(&font);

GetDlgItem(IDC_KEY )->SetFont(&font, TRUE);
GetDlgItem(IDC_KEY0 )->SetFont(&font, TRUE);
GetDlgItem(IDC_KEY00)->SetFont(&font, TRUE);

```

```

GetDlgItem(IDC_KEY1 )->SetFont(&font, TRUE);
GetDlgItem(IDC_KEY2 )->SetFont(&font, TRUE);
GetDlgItem(IDC_KEY3 )->SetFont(&font, TRUE);
GetDlgItem(IDC_KEY4 )->SetFont(&font, TRUE);
GetDlgItem(IDC_KEY5 )->SetFont(&font, TRUE);
GetDlgItem(IDC_KEY6 )->SetFont(&font, TRUE);
GetDlgItem(IDC_KEY7 )->SetFont(&font, TRUE);
GetDlgItem(IDC_KEY8 )->SetFont(&font, TRUE);
GetDlgItem(IDC_KEY9 )->SetFont(&font, TRUE);

font.CreateFont(16, 0, 0, 0, FW_BOLD, FALSE, FALSE, 0, 0, 0, 0, 0, 0, _T("굴림"));
((CStatic*)this->GetDlgItem(IDC_KEYNO))->SetFont(&font);
// 화면상의 글자크기 확대 //

// START 해당버튼에 이미지를 덮는다.
c_backbtn.LoadBitmaps(IDB_KEYBOARD_BACK_OFF
,
IDB_KEYBOARD_BACK_ON , NULL, NULL );
c_backbtn.SizeToContent();
c_close_btn.LoadBitmaps(
IDB_KEYBOARD_CANCEL_OFF,
IDB_KEYBOARD_CANCEL_ON, NULL, NULL );
c_close_btn.SizeToContent();
c_clear_btn.LoadBitmaps(
IDB_KEYBOARD_CLEAR_OFF
,
IDB_KEYBOARD_CLEAR_ON , NULL, NULL );
c_clear_btn.SizeToContent();
c_ok_btn.LoadBitmaps
( IDB_KEYBOARD_OK_OFF
,
IDB_KEYBOARD_OK_ON , NULL, NULL );
c_ok_btn.SizeToContent();

```

```
// END 해당버튼에 이미지를 덮는다.
```

```
Invalidate();
```

```
UpdateWindow();
```

```
UpdateData(false);
```

```
return TRUE; // return TRUE unless you set the focus to a control
```

```
// EXCEPTION: OCX Property Pages should return FALSE
```

```
}
```

```
void CMOBILE_KEYBOARD::OnKey7()
```

```
{
```

```
UpdateData(true);
```

```
int keyl = smoney1.GetLength() ;
```

```
if(keyl < keyLength)
```

```
{
```

```
smoney1 += L"7" ;
```

```
if(KeyBoardOK == 0 )
```

```
{
```

```
m_oldkey = smoney1 ;
```

```
}
```

```
else
```

```
{
```

```
imoney1 = _wtoi(smoney1) ;
```

```
m_oldkey = KEYComma(imoney1) ;
```

```
}
```

```
}
```

```
else
```

```

    {
        MessageBox(_T("설절된 필드길이보다 길어서 더이상 등록하실수 없습니
다..!"));
        Full_ScreenOK() ;
    }
    UpdateData(false);
}

```

```

void CMOBILE_KEYBOARD::OnKey8()

```

```

{
    UpdateData(true);
    int keyl = smoney1.GetLength() ;
    if(keyl < keyLength)
    {
        smoney1 += L"8" ;
        if(KeyBoardOK == 0 )
        {
            m_oldkey = smoney1 ;
        }
        else
        {
            imoney1 = _wtoi(smoney1) ;
            m_oldkey = KEYComma(imoney1) ;
        }
    }
    else
    {

```

```

        MessageBox(_T("설정된 필드길이보다 길어서 더 이상 등록 하실 수 없습
니다..!"));

        Full_ScreenOK() ;

    }

    UpdateData(false);
}

void CMOBILE_KEYBOARD::OnKey9()
{
    UpdateData(true);
    int keyl = smoney1.GetLength() ;
    if(keyl < keyLength)
    {
        smoney1 += L"9" ;
        if(KeyBoardOK == 0 )
        {
            m_oldkey = smoney1 ;
        }
        else
        {
            imoney1 = _wtoi(smoney1) ;
            m_oldkey = KEYComma(imoney1) ;
        }
    }
    else
    {
        MessageBox(_T("설정된 필드길이보다 길어서 더 이상 등록 하실 수 없습

```

```

니다!");
        Full_ScreenOK() ;
    }
    UpdateData(false);
}

void CMOBILE_KEYBOARD::OnKey4()
{
    UpdateData(true);
    int keyl = smoney1.GetLength() ;
    if(keyl < keyLength)
    {
        smoney1 += L"4" ;
        if(KeyBoardOK == 0 )
        {
            m_oldkey = smoney1 ;
        }
        else
        {
            imoney1 = _wtoi(smoney1) ;
            m_oldkey = KEYComma(imoney1) ;
        }
    }
    else
    {
        MessageBox(_T("설절된 필드길이보다 길어서 더이상 등록하실수 없습니다..!"));
        Full_ScreenOK() ;
    }
}

```

```

}

    UpdateData(false);
}

void CMOBILE_KEYBOARD::OnKey5()
{
    UpdateData(true);
    int keyl = smoney1.GetLength() ;
    if(keyl < keyLength)
    {
        smoney1 += L"5" ;
        if(KeyBoardOK == 0 )
        {
            m_oldkey = smoney1 ;
        }
        else
        {
            imoney1 = _wtoi(smoney1) ;
            m_oldkey = KEYComma(imoney1) ;
        }
    }
    else
    {
        MessageBox(_T("설절된 필드길이보다 길어서 더이상 등록하실수 없습니다..!"));
        Full_ScreenOK() ;
    }

    UpdateData(false);
}

```



```
}
```

```
void CMOBILE_KEYBOARD::OnKey6()
```

```
{
```

```
    UpdateData(true);
```

```
    int keyl = smoney1.GetLength() ;
```

```
    if(keyl < keyLength)
```

```
    {
```

```
        smoney1 += L"6" ;
```

```
        if(KeyBoardOK == 0 )
```

```
        {
```

```
            m_oldkey = smoney1 ;
```

```
        }
```

```
    else
```

```
    {
```

```
        imoney1 = _wtoi(smoney1) ;
```

```
        m_oldkey = KEYComma(imoney1) ;
```

```
    }
```

```
    }
```

```
    else
```

```
    {
```

```
        MessageBox(_T("설절된 필드길이보다 길어서 더이상 등록하실수 없습니다..!"));
```

```
        Full_ScreenOK() ;
```

```
    }
```

```
        UpdateData(false);
```

```
}
```

```

void CMOBILE_KEYBOARD::OnKey1()
{
    UpdateData(true);
    int keyl = smoney1.GetLength() ;
    if(keyl < keyLength)
    {
        smoney1 += L"1" ;
        if(KeyBoardOK == 0 )
        {
            m_oldkey = smoney1 ;
        }
        else
        {
            imoney1 = _wtoi(smoney1) ;
            m_oldkey = KEYComma(imoney1) ;
        }
    }
    else
    {
        MessageBox(_T("설절된 필드길이보다 길어서 더이상 등록하실수 없습니다..!"));
        Full_ScreenOK() ;
    }
    UpdateData(false);
}

void CMOBILE_KEYBOARD::OnKey2()
{

```

```

UpdateData(true);
int keyl = smoney1.GetLength() ;
if(keyl < keyLength)
{
    smoney1 += L"2" ;
    if(KeyBoardOK == 0 )
    {
        m_oldkey = smoney1 ;
    }
    else
    {
        imoney1 = _wtoi(smoney1) ;
        m_oldkey = KEYComma(imoney1) ;
    }
}
else
{
    MessageBox(_T("설절된 필드길이보다 길어서 더이상 등록하실수 없습니다..!"));
    Full_ScreenOK() ;
}

UpdateData(false);
}

void CMOBILE_KEYBOARD::OnKey3()
{
    UpdateData(true);
    int keyl = smoney1.GetLength() ;

```

```

if(keyl < keyLength)
{
    smoney1 += L"3" ;
    if(KeyBoardOK == 0 )
    {
        m_oldkey = smoney1 ;
    }
    else
    {
        imoney1 = _wtoi(smoney1) ;
        m_oldkey = KEYComma(imoney1) ;
    }
}
else
{
    MessageBox(_T("설절된 필드길이보다 길어서 더이상 등록하실수 없습니다..!"));
    Full_ScreenOK() ;
}

    UpdateData(false);
}

void CMOBILE_KEYBOARD::OnKey0()
{
    UpdateData(true);
    int keyl = smoney1.GetLength() ;
    if(keyl < keyLength)
    {

```

```

        smoney1 += L"0" ;
        if(KeyBoardOK == 0 )
        {
                m_oldkey = smoney1 ;
        }
        else
        {
                imoney1 = _wtoi(smoney1) ;
                m_oldkey = KEYComma(imoney1) ;
        }
}
else
{
        MessageBox(_T("설절된 필드길이보다 길어서 더이상 등록하실수 없습니다..!"));
        Full_ScreenOK() ;
}

        UpdateData(false);
}

void CMOBILE_KEYBOARD::OnKey00()
{
        UpdateData(true);
        int keyl = smoney1.GetLength() +1 ;
        if(keyl < keyLength)
        {
                smoney1 += L"00" ;
                if(KeyBoardOK == 0 )

```

```

        {
            m_oldkey = smoney1 ;
        }
        else
        {
            imoney1 = _wtoi(smoney1) ;
            m_oldkey = KEYComma(imoney1) ;
        }
    }
    else
    {
        MessageBox(_T("설절된 필드길이보다 길어서 더이상 등록하실수 없습니다..!"));
        Full_ScreenOK() ;
    }
    UpdateData(false);
}

void CMOBILE_KEYBOARD::OnKey()
{
    UpdateData(true);
    int keyl = smoney1.GetLength() ;
    if(keyl < keyLength)
    {
        smoney1 += L"-";
        if(KeyBoardOK == 0 )
        {
            m_oldkey = smoney1 ;
        }
    }
}

```

```

        }
        else
        {
            imoney1    = _wtoi(smoney1) ;
            m_oldkey = KEYComma(imoney1) ;
        }
    }
    else
    {
        MessageBox(_T("설절된 필드길이보다 길어서 더이상 등록하실수 없습니다..!"));
        Full_ScreenOK() ;
    }

    UpdateData(false);
}

void CMOBILE_KEYBOARD::OnBackbtn()
{
    UpdateData(true);
    int          keylen ;
    CString Keystr ;
    keylen      = smoney1.GetLength() -1 ;
    Keystr      = smoney1.Mid(0,keylen) ;
    smoney1     = Keystr ;
    imoney1     = _wtoi(smoney1) ;

    if(KeyBoardOK == 0)
    {

```

```

        m_oldkey    = smoney1 ;
    }
    else
    {
        m_oldkey    = KEYComma(imoney1) ;
    }
    UpdateData(false);
}

```

```

void CMOBILE_KEYBOARD::OnCancleBtn()
{
    UpdateData(true);
    strKEY = oldKEY ;
    keyYS  = 0 ;
    UpdateData(false);
    OnOK() ;
}

```

```

void CMOBILE_KEYBOARD::OnClearBtn()
{
    UpdateData(true) ;
    CString chekmin ;
    smoney1 = "" ;
    imoney1 = 0 ;
}

```



```

        m_oldkey = "" ;
        UpdateData(false);
    }

void CMOBILE_KEYBOARD::OnOkBtn()
{
    UpdateData(true) ;
    if(m_oldkey.GetLength() > 0 )
    {
        strKEY = smoney1 ;
        keyYS = 1 ;
    }
    else
    {
        strKEY = smoney1 ;
        keyYS = 0 ;
    }
    UpdateData(false);
    OnOK() ;
}

```

나. 구매입고 모듈

```

// SIW_Q0001.cpp : implementation file
//
#include "stdafx.h"
#include "resource.h"

```

```

#include "exfarmDlg.h"

#include "SIW_Q0001.h"

#include "SIW_T0001.h" // 하위폼

#ifdef _DEBUG
#define new DEBUG_NEW
#undef THIS_FILE
static char THIS_FILE[] = __FILE__;
#endif

extern IADOCERecordset* m_pRecordset;
extern IADOCEConnection* m_pConnection;

extern CString      strDBConn      ;
extern CString      strUserCode ; // 사용자코드
extern CString      strUserName ; // 사용자명칭
extern CString      strPgmNo   ; // 프로그램코드
extern CString      strKEY      ; // 키보드 입력값
extern CString      strKEYTITLE ; // 키보드입력 타이틀
extern int          keyYS       ; // 키보드입력후 1확인/0취소
extern int          KeyBoardOK  ; //(0:패스워드,1:금액,수량) ;
extern int          keyLength   ; // 키보드입력길이
extern HBRUSH brush ;
extern CFont font;

// 수매입고에서 사용될 변수선언 (필드명칭을 따른다..)

```

```

CString CBUY_SER ; // 구매일련번호
CString CMAT_OPT ; // 원료사품구분
CString CBUY_DT ; // 구매일자
CString CITEM_CD ; // 품목코드
CString CITEM_NAME; // 품목명

CString CFARM_CD ; // 농가코드
CString CFARM_NAME; // 농가명
CString CSTORE_DT ; // 입고일자
CString CGRADE_CD ; // 등급코드
CString CCONT_DT ; // 약정일자

CString CAREACD ; // 지역코드
int CQUANTI ; // 수량
int CAMT ; // 금액
int CKG ; // 중량

CString PGM MODE ; // 프로그램모드 (신규, 수정)

////////////////////////////////////
// CSIW_Q0001 dialog

CSIW_Q0001::CSIW_Q0001(CWnd* pParent /*!=NULL*/)
    : CDialog(CSIW_Q0001::IDD, pParent)
{
   //{{AFX_DATA_INIT(CSIW_Q0001)
    m_username = _T("");

```

```

        m_gname = _T("");
        m_pname = _T("");
        //}}AFX_DATA_INIT
    }

// 화면을 풀스크린으로 만든다.
void CSIW_Q0001::SIW_Q0001_ScreenOK()
{
    // START 화면을 전체화면으로 키운다..
    SHINITDLGINFO shidi;

    shidi.dwMask = SHIDIM_FLAGS;

    shidi.dwFlags = SHIDIF_FULLSCREENNOMENUBAR | SHIDIF_SIPDOWN;
    shidi.hDlg = m_hWnd;
    SHInitDialog(&shidi);
    ::CommandBar_Show(m_pWndEmptyCB->m_hWnd, FALSE);
    SetForegroundWindow();
    ::SHFullScreen(m_hWnd, SHFS_HIDETASKBAR | SHFS_HIDESIPBUTTON |
SHFS_HIDESTARTICON);
    MoveWindow(0, 0, 240, 360, TRUE);
    // END 화면을 전체화면으로 키운다..
}

void CSIW_Q0001::ComboAddShow()
{
    CString strSQL;
    HRESULT hr = S_OK;
    long lcount;

```

```

long nState;

long option = adCmdText;

COleVariant varFields;

COleVariant varSQL;

COleVariant varConn;

CursorTypeEnum cte = adOpenDynamic;

LockTypeEnum lte = adLockOptimistic;

IADOCEField          *field    = NULL;

IADOCEFields         *fields   = NULL;

CString CheckStr ;

CheckStr = _T(""); // 중복검사

// 농가정보를 콤보박스에 등록한다.. //

strSQL.Format(_T("SELECT Farm_name FROM TM_STORE WHERE User_yn = '%s'
ORDER BY Farm_name DESC "),L"Y");

varSQL.SetString(strSQL,VT_BSTR);

varConn.SetString(strDBConn,VT_BSTR);

m_pRecordset->get_State(&nState);

if(nState==1)m_pRecordset->Close();

hr = m_pRecordset->Open(varSQL,varConn,cte,lte,option);

if(FAILED(hr))

{

    MessageBox(_T("DB나 쿼리문에 이상이 있습니다..."));

    SIW_Q0001_ScreenOK() ;
}

```

```

        return;
    }

    hr = m_pRecordset->get_RecordCount(&lcount);
// 레코드의 개수를 얻는다.
    if(lcount==0)
    {
        MessageBox(_T("DB나 쿼리문에 이상이 있습니다..."));
        SIW_Q0001_ScreenOK() ;
        return ; // 레코드가 없으면 되돌아간다.
    }

    COleVariant itemFields ; // 필드이름을 저장할 변수

    COleVariant itemFARM_NAME; // 농가명칭
    CString strOutput ; // 리스트에 추가할 문자열

c_gname.AddString(L"" ) ;
    for(int i=0;i<lcount;i++)
    {
        hr = m_pRecordset->get_Fields(&fields);
        // 레코드셋으로부터 Fields 컬렉션 인터페이스를 얻는다.

        // -- 농가명칭 --- //
        itemFields.SetString(_T("Farm_name"),VT_BSTR);
        hr = fields->get_Item(itemFields,&field);
        hr = field->get_Value(&itemFARM_NAME);
c_gname.AddString(itemFARM_NAME.bstrVal) ;
    }

```

```

//                                if(i == 0)
//                                {
//                                m_gname = itemFARM_NAME.bstrVal ;
//                                }

                                hr = m_pRecordset->MoveNext();
                                // 다음레코드로 넘어간다.
                                } // 레코드의 개수만큼 루프를 돌린다.
                                field ->Release(); // Field와 Fields 오브젝트의
                                fields->Release();// 인터페이스를 해제한다.
                                field = 0;
                                fields = 0;
                                hr = m_pRecordset->Close();//레코드셋을 닫는다.
                                // 농가정보를 콤보박스에 등록한다.. //

                                // 품목정보를 콤보박스에 등록한다.. //
                                strSQL.Format(_T("SELECT Item_name FROM TM_ITEM WHERE User_yn = '%s' ORDER
                                BY Item_name DESC "),L"Y") ;

                                varSQL.SetString(strSQL,VT_BSTR);
                                varConn.SetString(strDBConn,VT_BSTR);

                                m_pRecordset->get_State(&nState);
                                if(nState==1)m_pRecordset->Close();
                                hr = m_pRecordset->Open(varSQL,varConn,cte,lte,option);
                                if(FAILED(hr))

```

```

    {
        MessageBox(_T("DB나 쿼리문에 이상이 있습니다..."));
        SIW_Q0001_ScreenOK() ;
        return;
    }
    hr = m_pRecordset->get_RecordCount(&lcount);
// 레코드의 개수를 얻는다.
    if(lcount==0)
    {
        MessageBox(_T("DB나 쿼리문에 이상이 있습니다..."));
        SIW_Q0001_ScreenOK() ;
        return ; // 레코드가 없으면 되돌아간다.
    }

    COleVariant itemITEM_NAME; // 품목명칭

    c_pname.AddString(L"");
    for(int j=0;j<lcount;j++)
    {
        hr = m_pRecordset->get_Fields(&fields);
// 레코드셋으로부터 Fields 컬렉션 인터페이스를 얻는다.

        // -- 품목명칭 --- //
        itemFields.SetString(_T("Item_name"),VT_BSTR);
        hr = fields->get_Item(itemFields,&field);
        hr = field->get_Value(&itemITEM_NAME);
        c_pname.AddString(itemITEM_NAME.bstrVal) ;
    }

```



```

//          if(j == 0)
//          {
//          m_pname = itemITEM_NAME.bstrVal ;
//          }

```

```

          hr = m_pRecordset->MoveNext();
          // 다음레코드로 넘어간다.
        }          // 레코드의 개수만큼 루프를 돌린다.
        field ->Release(); // Field와 Fields 오브젝트의
        fields->Release();// 인터페이스를 해제한다.
        field = 0;
        fields = 0;
        hr = m_pRecordset->Close();//레코드셋을 닫는다.
        // 품목정보를 콤보박스에 등록한다.. //

```

```

        UpdateData(false);
    }

```

```

void CSIW_Q0001::SIW_Q0001ListTitle()
{
    int nColumnCount = m_list.GetHeaderCtrl()->GetItemCount();
    for (int j=0;j < nColumnCount;j++)
    {
        m_list.DeleteColumn(0);
    }
}

```

```

m_list.SetExtendedStyle(LVS_EX_FULLROWSELECT | LVS_EX_GRIDLINES);

int intWid[5];
intWid[0] = 36 ; // 번호
intWid[1] = 36 ; // 구분
intWid[2] = 45 ; // 일자
intWid[3] = 50 ; // 품목
intWid[4] = 60 ; // 농가

TCHAR col[5][40] = {_T("번호"),_T("구분"),_T("일자"),_T("품목"),_T("농가")};
LV_COLUMN lvcolumn;

while(m_list.DeleteColumn(0));
for (int i = 0 ; i < 5 ;i++)
{
lvcolumn.mask = LVCF_FMT | LVCF_SUBITEM | LVIF_TEXT | LVCF_WIDTH;
    lvcolumn.fmt = LVCFMT_CENTER;
    lvcolumn.pszText = col[i];
    lvcolumn.iSubItem = i;
    lvcolumn.cx = intWid[i]; // SubItem is twice as large
    m_list.InsertColumn(i, &lvcolumn); // assumes return value is OK.}
}

}

void CSIW_Q0001::ListShow()
{
    SetCursor(AfxGetApp()->LoadCursor(IDC_WAIT));
}

```

```

        m_list.DeleteAllItems() ;
// 작업변수를 설정한다.-----
        CString strSQL;
        HRESULT hr = S_OK;
        long lcount;
        long nState;
        long option = adCmdText;
        COleVariant varFields;
        COleVariant varSQL;
        COleVariant varConn;
        CursorTypeEnum cte = adOpenDynamic;
        LockTypeEnum lte = adLockOptimistic;

        IADOCEField          *field = NULL;
        IADOCEFields         *fields = NULL;
        CString CheckStr ;
        CheckStr = _T(""); // 중복검사

        strSQL.Format(_T("SELECT Buy_ser, Mat_opt, Buy_dt, Item_name, Farm_name FROM
        TBL_STORE WHERE Next_OK = '%s' ORDER BY Buy_ser DESC"),L"N") ;
        varSQL.SetString(strSQL,VT_BSTR);
        varConn.SetString(strDBConn,VT_BSTR);

        m_pRecordset->get_State(&nState);
        if(nState==1)m_pRecordset->Close();
        hr = m_pRecordset->Open(varSQL,varConn,cte,lte,option);
        if(FAILED(hr))

```

다.

```
{
    SetCursor(AfxGetApp()->LoadCursor(IDC_NO));
    SIW_Q0001_ScreenOK() ;
    return;
}
hr = m_pRecordset->get_RecordCount(&lcount); // 레코드의 개수를 얻는

if(lcount==0)
{
    SetCursor(AfxGetApp()->LoadCursor(IDC_NO));

    SIW_Q0001_ScreenOK() ;
    return ; // 레코드가 없으면 되돌아간다.
}
COleVariant itemFields          ; // 필드이름을 저장할 변수

COleVariant itemgBuy_ser   ; // 구매일련번호
COleVariant itemgMat_opt   ; // 원료상품구분
COleVariant itemeBuy_dt    ; // 구매일자
COleVariant itemeItem_name ; // 품목명
COleVariant itemeFarm_name ; // 상품명

CString  strOutput  ; // 리스트에 추가할 문자열
CString  strDD      ; // 구매일자를 mm.dd로 변경할 변수

// 리스트에 등록될 단속내역
LV_ITEM  lvitem     ;
```

```

int     workok = 0 ;

int     workno = 0 ;
for(int i=0;i<lcount;i++)
{
    // 레코드셋으로부터 Fields 컬렉션 인터페이스를 얻는다.
    hr = m_pRecordset->get_Fields(&fields);
    // -- 수매일련번호 --- //
    itemFields.SetString(_T("Buy_ser"),VT_BSTR);
    hr = fields->get_Item(itemFields,&field);
    hr = field->get_Value(&itemgBuy_ser);
    lvitem.iSubItem = 0;
    lvitem.mask = LVIF_TEXT;
    lvitem.pszText = itemgBuy_ser.bstrVal;
    lvitem.iItem = 0;
    m_list.InsertItem(&lvitem);

    // -- 원료상품구분 --- //
    itemFields.SetString(_T("Mat_opt"),VT_BSTR);
    hr = fields->get_Item(itemFields,&field);
    hr = field->get_Value(&itemgMat_opt);
    lvitem.iSubItem = 1;
    lvitem.mask = LVIF_TEXT;
    lvitem.pszText = itemgMat_opt.bstrVal;
    lvitem.iItem = 0;
    m_list.SetItem(&lvitem);

    // -- 수매일자 --- //
    itemFields.SetString(_T("Buy_dt"),VT_BSTR);

```

```

hr = fields->get_Item(itemFields,&field);
hr = field->get_Value(&itemeBuy_dt);
lvitem.iSubItem = 2;
lvitem.mask = LVIF_TEXT;

strDD = itemeBuy_dt.bstrVal ;
strDD = strDD.Mid(5,2) + L"." + strDD.Mid(8,2) ;

lvitem.pszText = (LPTSTR)strDD.operator LPCTSTR();
lvitem.iItem = 0;
m_list.SetItem(&lvitem);

// -- 품목명 --- //
itemFields.SetString(_T("Item_name"),VT_BSTR);
hr = fields->get_Item(itemFields,&field);
hr = field->get_Value(&itemeItem_name);
lvitem.iSubItem = 3;
lvitem.mask = LVIF_TEXT;
lvitem.pszText = itemeItem_name.bstrVal;
lvitem.iItem = 0;
m_list.SetItem(&lvitem);

// -- 상품명 --- //
itemFields.SetString(_T("Farm_name"),VT_BSTR);
hr = fields->get_Item(itemFields,&field);
hr = field->get_Value(&itemeFarm_name);

```

```

        lvitem.iSubItem = 4;
        lvitem.mask = LVIF_TEXT;
        lvitem.pszText = itemeFarm_name.bstrVal;
        lvitem.iItem = 0;
        m_list.SetItem(&lvitem);
        // 다음레코드로 넘어간다.
        hr = m_pRecordset->MoveNext();
        CheckStr = strOutput ;
    }        // 레코드의 개수만큼 루프를 돌린다.
    UpdateData(false);

    field ->Release(); // Field와 Fields 오브젝트의
    fields->Release();// 인터페이스를 해제한다.
    field = 0;
    fields = 0;
    hr = m_pRecordset->Close();//레코드셋을 닫는다.

    SetCursor(AfxGetApp()->LoadCursor(IDC_NO));
}

```

```

void CSIW_Q0001::DoDataExchange(CDataExchange* pDX)
{
    CDialog::DoDataExchange(pDX);
    //{{AFX_DATA_MAP(CSIW_Q0001)
    DDX_Control(pDX, IDC_LIST1, m_list);

```

```

        DDX_Control(pDX, COMBO_PNAME, c_pname);
        DDX_Control(pDX, COMBO_GNAME, c_gname);
        DDX_Control(pDX, IDC_Q0001_MBTN5, c_btn5);
        DDX_Control(pDX, IDC_Q0001_MBTN4, c_btn4);
        DDX_Control(pDX, IDC_Q0001_MBTN3, c_btn3);
        DDX_Control(pDX, IDC_Q0001_MBTN2, c_btn2);
        DDX_Control(pDX, IDC_Q0001_MBTN1, c_btn1);
        DDX_Text(pDX, IDC_C_USERNAME, m_username);
        DDX_CBString(pDX, COMBO_GNAME, m_gname);
        DDX_CBString(pDX, COMBO_PNAME, m_pname);
    //}}AFX_DATA_MAP
}

```

```

BEGIN_MESSAGE_MAP(CSIW_Q0001, CDialog)
    //{{AFX_MSG_MAP(CSIW_Q0001)
    ON_BN_CLICKED(IDC_Q0001_MBTN5, OnQ0001Mbtn5)
    ON_BN_CLICKED(IDC_Q0001_MBTN2, OnQ0001Mbtn2)
    ON_NOTIFY(NM_CLICK, IDC_LIST1, OnClickList1)
    ON_BN_CLICKED(IDC_Q0001_MBTN1, OnQ0001Mbtn1)
    ON_BN_CLICKED(IDC_Q0001_MBTN3, OnQ0001Mbtn3)
    ON_BN_CLICKED(IDC_Q0001_MBTN4, OnQ0001Mbtn4)
    //}}AFX_MSG_MAP
END_MESSAGE_MAP()

```

```

////////////////////////////////////
// CSIW_Q0001 message handlers

```



```

BOOL CSIW_Q0001::OnInitDialog()
{
    CDialog::OnInitDialog();

    m_username = strUserName ;

    SIW_Q0001_ScreenOK() ;
    SIW_Q0001ListTitle() ;
    ListShow() ;
    ComboAddShow() ; // 품목과 농가를 콤보박스에 등록한다..

    CBUY_SER = L"" ; //PK인 수매일련번호를 초기화한다..(선택 수매일련번호)

    // 사용자명의 글자체를 크게한다..
font.CreateFont(18, 0, 0, 0, FW_BOLD, FALSE, FALSE, 0, 0, 0, 0, 0, 0, _T("Arial"));
((CStatic*)this->GetDlgItem(IDC_C_USERNAME))->SetFont(&font);
    // TODO: Add extra initialization here
    UpdateData(false);
    return TRUE; // return TRUE unless you set the focus to a control
}

void CSIW_Q0001::OnQ0001Mbtn5()
{
    // 수매입고관리를 종료한다..
    OnOK() ;
}

```

```

void CSIW_Q0001::OnQ0001Mbtn2()
{
// 수매입고를 등록한다

// 신규 수매입고번호를 생성한다..
    CString strSQL;
    HRESULT hr = S_OK;

    long lcount;
    long nState;
    long option = adCmdText;
    COleVariant varFields;
    COleVariant varSQL;
    COleVariant varConn;
    CursorTypeEnum cte = adOpenDynamic;
    LockTypeEnum lte = adLockOptimistic;

    IADOCEField          *field = NULL;
    IADOCEFields         *fields = NULL;
    CString CheckStr ;
    CheckStr = _T(""); // 중복검사

    strSQL.Format(_T("SELECT Buy_ser FROM TBL_STORE ORDER BY Buy_ser
DESC"));
    varSQL.SetString(strSQL,VT_BSTR);
    varConn.SetString(strDBConn,VT_BSTR);

    m_pRecordset->get_State(&nState);

```

```

if(nState==1)m_pRecordset->Close();

hr = m_pRecordset->Open(varSQL,varConn,cte,lte,option);

hr = m_pRecordset->get_RecordCount(&lcount); // 레코드의 개수를 얻는다.
if(lcount==0)
{
    CBUY_SER = L"0001" ;
    // 값이 없으면 구매일련번호를 0001로 지정한다.
}

COleVariant itemFields          ; // 필드이름을 저장할 변수

COleVariant itemBUY_SER ; // 구매일련번호
CString      strOutput ; // 리스트에 추가할 문자열
// 레코드셋으로부터 Fields 컬렉션 인터페이스를 얻는다.
hr = m_pRecordset->get_Fields(&fields);

// -- 구매일련번호 --- //
itemFields.SetString(_T("Buy_ser"),VT_BSTR);
hr = fields->get_Item(itemFields,&field);
hr = field->get_Value(&itemBUY_SER);
strOutput = itemBUY_SER.bstrVal ;
int buyno ;
buyno = _wtoi(strOutput) ;
CBUY_SER.Format(_T("%0.4d"),buyno+1) ;
// -- 구매일련번호 --- //

field ->Release(); // Field와 Fields 오브젝트의
fields->Release();// 인터페이스를 해제한다.

```

```

field = 0;
fields = 0;
hr = m_pRecordset->Close();//레코드셋을 닫는다.

PGMMODE = L"신규" ;
UpdateData(false);
// 신규 수매입고번호를 생성한다..(끝)

// 변수 초기화
CMAT_OPT   = L"" ;
CBUY_DT    = L"" ;
CITEM_NAME = L"" ;
CFARM_NAME = L"" ;
CSTORE_DT  = L"" ;
CCONT_DT   = L"" ;
CAREACD    = L"" ;

CQUANTI    = 0 ;
CAMT       = 0 ;
CKG        = 0 ;

// 변수 초기화

// 수매입고 등록화면을 VIEW
CSIW_T0001 T0001Dlg ;
T0001Dlg.DoModal() ;
SIW_Q0001_ScreenOK() ;

```

```

// 수매입고 등록화면을 VIEW
    ListShow() ;
}

void CSIW_Q0001::OnClickList1(NMHDR* pNMHDR, LRESULT* pResult)
{
// LIST의 자료를 선택하면 선택된 수매일련번호를 기억한다.
    NM_LISTVIEW* pList = (NM_LISTVIEW*)pNMHDR;
    ASSERT(pList);
    *pResult = 0;

    if(pList->iItem == -1)
    {
        CBUY_SER = L"" ;
        return;
    }

    CBUY_SER = m_list.GetItemText(pList->iItem,0); // 수매일련번호
    UpdateData(false);
}

void CSIW_Q0001::OnQ0001Mbtn1()
{
// 조회버튼을 클릭 하였을때 선택된 농가와, 품목을 기준으로 수매입고내역을 불러온다..
    UpdateData(true);

    SetCursor(AfxGetApp()->LoadCursor(IDC_WAIT));
    m_list.DeleteAllItems() ;
}

```

```

// 작업변수를 설정한다..-----
CString strSQL;
HRESULT hr = S_OK;
long lcount;
long nState;
long option = adCmdText;
COleVariant varFields;
COleVariant varSQL;
COleVariant varConn;
CursorTypeEnum cte = adOpenDynamic;
LockTypeEnum lte = adLockOptimistic;

IADOCEField *field = NULL;
IADOCEFields *fields = NULL;
CString CheckStr ;
CheckStr = _T(""); // 중복검사

// 상품과 거래처를 선택하지 않았다..
if (m_pname == L"" && m_gname == L"")
{
strSQL.Format(_T("SELECT Buy_ser, Mat_opt, Buy_dt, Item_name, Farm_name FROM
TBL_STORE WHERE Next_OK = '%s' ORDER BY Buy_ser DESC"),L"N") ;
}
// 상품과 거래처를 모두 선택했다..
else if (m_pname != L"" && m_gname != L"")
{
strSQL.Format(_T("SELECT Buy_ser, Mat_opt, Buy_dt, Item_name, Farm_name FROM

```

```

TBL_STORE WHERE Next_OK = '%s' AND Item_name = '%s' AND Farm_name = '%s'
ORDER BY Buy_ser DESC"),L"N",m_pname,m_gname) ;
    }
    // 거래처를 선택했다.
    else if (m_pname == L"")
    {
strSQL.Format(T("SELECT Buy_ser, Mat_opt, Buy_dt, Item_name, Farm_name FROM
TBL_STORE WHERE Next_OK = '%s' AND Farm_name = '%s' ORDER BY Buy_ser
DESC"),L"N",m_gname) ;
    }
    // 상품을 선택했다.
    else if (m_gname == L"")
    {
strSQL.Format(T("SELECT Buy_ser, Mat_opt, Buy_dt, Item_name, Farm_name FROM
TBL_STORE WHERE Next_OK = '%s' AND Item_name = '%s' ORDER BY Buy_ser
DESC"),L"N",m_pname) ;
    }

varSQL.SetString(strSQL,VT_BSTR);
varConn.SetString(strDBConn,VT_BSTR);

m_pRecordset->get_State(&nState);
if(nState==1)m_pRecordset->Close();
hr = m_pRecordset->Open(varSQL,varConn,cte,lte,option);
if(FAILED(hr))
{
    SetCursor(AfxGetApp()->LoadCursor(IDC_NO));
}

```

```

        SIW_Q0001_ScreenOK() ;

        return;
    }

    hr = m_pRecordset->get_RecordCount(&lcount); // 레코드의 개수를 얻는다.
    if(lcount==0)
    {
        SetCursor(AfxGetApp()->LoadCursor(IDC_NO));

        SIW_Q0001_ScreenOK() ;

        return ; // 레코드가 없으면 되돌아간다.
    }

```

```

COleVariant itemFields          ; // 필드이름을 저장할 변수

```

```

COleVariant itemgBuy_ser      ; // 구매일련번호

```

```

COleVariant itemgMat_opt     ; // 원료상품구분

```

```

COleVariant itemeBuy_dt      ; // 구매일자

```

```

COleVariant itemeItem_name   ; // 품목명

```

```

COleVariant itemeFarm_name   ; // 상품명

```

```

CString    strOutput        ; // 리스트에 추가할 문자열

```

```

CString    strDD            ; // 구매일자를 mm.dd로 변경할 변수

```

```

LV_ITEM    lvitem          ; // 리스트에 등록될 단속내역

```

```

int        workok = 0 ;

```

```

int        workno = 0 ;

```

```

for(int i=0;i<lcount;i++)

```

```

{
    // 레코드셋으로부터 Fields 컬렉션 인터페이스를 얻는다.

```



```

hr = m_pRecordset->get_Fields(&fields);

// -- 구매일련번호 --- //
itemFields.SetString(_T("Buy_ser"),VT_BSTR);
hr = fields->get_Item(itemFields,&field);
hr = field->get_Value(&itemgBuy_ser);
lvitem.iSubItem = 0;
lvitem.mask = LVIF_TEXT;
lvitem.pszText = itemgBuy_ser.bstrVal;
lvitem.iItem = 0;
m_list.InsertItem(&lvitem);

// -- 원료상품구분 --- //
itemFields.SetString(_T("Mat_opt"),VT_BSTR);
hr = fields->get_Item(itemFields,&field);
hr = field->get_Value(&itemgMat_opt);
lvitem.iSubItem = 1;
lvitem.mask = LVIF_TEXT;
lvitem.pszText = itemgMat_opt.bstrVal;
lvitem.iItem = 0;
m_list.SetItem(&lvitem);

// -- 구매일자 --- //
itemFields.SetString(_T("Buy_dt"),VT_BSTR);
hr = fields->get_Item(itemFields,&field);
hr = field->get_Value(&itemeBuy_dt);
lvitem.iSubItem = 2;

```

```

lvitem.mask = LVIF_TEXT;

strDD = itemeBuy_dt.bstrVal ;
strDD = strDD.Mid(5,2) + L"." + strDD.Mid(8,2) ;

lvitem.pszText = (LPTSTR)strDD.operator LPCTSTR();
lvitem.iItem = 0;
m_list.SetItem(&lvitem);

// -- 품목명 --- //
itemFields.SetString(_T("Item_name"),VT_BSTR);
hr = fields->get_Item(itemFields,&field);
hr = field->get_Value(&itemItem_name);
lvitem.iSubItem = 3;
lvitem.mask = LVIF_TEXT;
lvitem.pszText = itemItem_name.bstrVal;
lvitem.iItem = 0;
m_list.SetItem(&lvitem);

// -- 농가명 --- //
itemFields.SetString(_T("Farm_name"),VT_BSTR);
hr = fields->get_Item(itemFields,&field);
hr = field->get_Value(&itemFarm_name);
lvitem.iSubItem = 4;
lvitem.mask = LVIF_TEXT;
lvitem.pszText = itemFarm_name.bstrVal;
lvitem.iItem = 0;

```

```

        m_list.SetItem(&lvitem);
        // 다음레코드로 넘어간다.
        hr = m_pRecordset->MoveNext();
        CheckStr = strOutput ;

    }          // 레코드의 개수만큼 루프를 돌린다.
    UpdateData(false);

    field ->Release(); // Field와 Fields 오브젝트의
    fields->Release();// 인터페이스를 해제한다.
    field = 0;
    fields = 0;
    hr = m_pRecordset->Close();//레코드셋을 닫는다.

    SetCursor(AfxGetApp()->LoadCursor(IDC_NO));
}

void CSIW_Q0001::OnQ0001Mbtn3()
{
    // 수매입고 수정
    if(CBUY_SER == L"")
    {
        MessageBox(_T("수정하실 수매내역을 선택하세요!!"));
        SIW_Q0001_ScreenOK() ;
        return;
    }
    else

```

```

    {
//-----
        // 선택한 수매입고번호를 이용하여 수정할 수매입고내역을 변수에 등록한다..
//-----

        UpdateData(true);

        SetCursor(AfxGetApp()->LoadCursor(IDC_WAIT));
        // 작업변수를 설정한다..-----
        CString strSQL;
        HRESULT hr = S_OK;
        long lcount;
        long nState;
        long option = adCmdText;
        COleVariant varFields;
        COleVariant varSQL;
        COleVariant varConn;
        CursorTypeEnum cte = adOpenDynamic;
        LockTypeEnum lte = adLockOptimistic;

        IADOCEField          *field      = NULL;
        IADOCEFields         *fields     = NULL;
        CString CheckStr ;

        strSQL.Format(_T("SELECT * FROM TBL_STORE WHERE
Buy_ser = '%s' "),CBUY_SER) ;

        varSQL.SetString(strSQL,VT_BSTR);
        varConn.SetString(strDBConn,VT_BSTR);

```

```

m_pRecordset->get_State(&nState);
if(nState==1)m_pRecordset->Close();
hr = m_pRecordset->Open(varSQL,varConn,cte,lte,option);
if(FAILED(hr))
{
    SetCursor(AfxGetApp()->LoadCursor(IDC_NO));
    SIW_Q0001_ScreenOK() ;

    return;
}
// 레코드의 개수를 얻는다.
hr = m_pRecordset->get_RecordCount(&lcount);
if(lcount==0)
{
    SetCursor(AfxGetApp()->LoadCursor(IDC_NO));

    SIW_Q0001_ScreenOK() ;
    return ; // 레코드가 없으면 되돌아간다.
}

COleVariant itemFields          ; // 필드이름을 저장할 변

수

COleVariant itemgMat_opt   ; // 원료상품구분
COleVariant itemeBuy_dt    ; // 구매일자
COleVariant itemeItem_name ; // 품목명

```

```

ColeVariant itemeFarm_name ; // 농가명
ColeVariant itemeStore_dt ; // 입고일자
ColeVariant itemeCont_dt ; // 약정일자
ColeVariant itemeAreacd ; // 지역코드
ColeVariant itemeQuanti ; // 수량
ColeVariant itemeAmt ; // 금액
ColeVariant itemeKg ; // 중량
// 레코드셋으로부터 Fields 컬렉션 인터페이스를 얻는다.
hr = m_pRecordset->get_Fields(&fields);
// -- 원료사품구분 --- //
itemFields.SetString(_T("Mat_opt"),VT_BSTR);
hr = fields->get_Item(itemFields,&field);
hr = field->get_Value(&itemgMat_opt);
CMAT_OPT = itemgMat_opt.bstrVal ;

// -- 수매일자 --- //
itemFields.SetString(_T("Buy_dt"),VT_BSTR);
hr = fields->get_Item(itemFields,&field);
hr = field->get_Value(&itemeBuy_dt);
CBUY_DT = itemeBuy_dt.bstrVal ;

// -- 품목명 --- //
itemFields.SetString(_T("Item_name"),VT_BSTR);
hr = fields->get_Item(itemFields,&field);
hr = field->get_Value(&itemeItem_name);
CITEM_NAME = itemeItem_name.bstrVal ;

```

```

// -- 농가명 --- //
itemFields.SetString(_T("Farm_name"),VT_BSTR);
hr = fields->get_Item(itemFields,&field);
hr = field->get_Value(&itemeFarm_name);
CFARM_NAME = itemeFarm_name.bstrVal ;

// -- 입고일자 --- //
itemFields.SetString(_T("Store_dt"),VT_BSTR);
hr = fields->get_Item(itemFields,&field);
hr = field->get_Value(&itemeStore_dt);
CSTORE_DT = itemeStore_dt.bstrVal ;

// -- 약정일자 --- //
itemFields.SetString(_T("Cont_dt"),VT_BSTR);
hr = fields->get_Item(itemFields,&field);
hr = field->get_Value(&itemeAreacd);
CCONT_DT = itemeAreacd.bstrVal ;

// -- 지역코드 --- //
itemFields.SetString(_T("Areacd"),VT_BSTR);
hr = fields->get_Item(itemFields,&field);
hr = field->get_Value(&itemeCont_dt);
CAREACD = itemeCont_dt.bstrVal ;

// -- 수량 --- //
itemFields.SetString(_T("Quanti"),VT_BSTR);
hr = fields->get_Item(itemFields,&field);

```

```

hr = field->get_Value(&itemeQuanti);
CQUANTI = itemeQuanti.intVal ;

// -- 금액 --- //
itemFields.SetString(_T("Amt"),VT_BSTR);
hr = fields->get_Item(itemFields,&field);
hr = field->get_Value(&itemeAmt);
CAMT = itemeAmt.intVal ;

// -- 중량 --- //
itemFields.SetString(_T("Kg"),VT_BSTR);
hr = fields->get_Item(itemFields,&field);
hr = field->get_Value(&itemeKg);
CKG = itemeKg.intVal ;

UpdateData(false);

field ->Release(); // Field와 Fields 오브젝트의
fields->Release();// 인터페이스를 해제한다.
field = 0;
fields = 0;
hr = m_pRecordset->Close();//레코드셋을 닫는다.
SetCursor(AfxGetApp()->LoadCursor(IDC_NO));

```

```

//-----
// 선택한 수매입고번호를 이용하여 수정할 수매입고내역을 변수에 등록한다..
//-----

```



```

    }

    PGMMODE = L"수정" ;

    // 수매입고 등록화면을 VIEW
    CSIW_T0001 T0001Dlg ;
    T0001Dlg.DoModal() ;
    SIW_Q0001_ScreenOK() ;
    ListShow() ;
    UpdateData(false);
}

void CSIW_Q0001::OnQ0001Mbtn4()
{
    // 수매입고 삭제
    UpdateData(true);
    if(CBUY_SER == L"")
    {
        MessageBox(_T("삭제하실 수매내역을 선택하세요!!"));
        SIW_Q0001_ScreenOK() ;
        return;
    }
    else
    {
        int nAnswer = MessageBox(_T("선택하신 수매입고내역을 삭제할까요?"),_T("수매입고 삭제"), MB_ICONINFORMATION|MB_OKCANCEL);
        if(nAnswer == IDOK)
        {

```

```

CString strSQL;
HRESULT hd = S_OK;
long option = adCmdText;
COleVariant varFields;
COleVariant varSQL;
COleVariant varConn;
CursorTypeEnum cte = adOpenDynamic;
LockTypeEnum lte = adLockOptimistic;
IADOCEField *field = NULL;
IADOCEFields *fields = NULL;

strSQL.Format(_T("DELETE TBL_STORE WHERE Buy_ser = '%s'"), CBUY_SER) ;
hd = m_pConnection->Execute(strSQL.GetBuffer(0), NULL, adCmdText, NULL) ;
if(FAILED(hd))
{
    MessageBox(_T("수매입고내역 삭제에 실패하였습
니다..")) ;

    SIW_Q0001_ScreenOK() ;

    return ;
}
}
SIW_Q0001_ScreenOK() ;
}
ListShow() ;
UpdateData(false);
}

```

다. 선별관리 모듈

```
// SEM_Q0001.cpp : implementation file
//
#include "stdafx.h"
#include "resource.h"
#include "exfarmDlg.h"
#include "SEM_Q0001.h"
#include "SEM_T0001.h" // 하위폼

#ifdef _DEBUG
#define new DEBUG_NEW
#undef THIS_FILE
static char THIS_FILE[] = __FILE__;
#endif

extern IADOCERecordset* m_pRecordset;
extern IADOCEConnection* m_pConnection;

extern CString      strDBConn      ;
extern CString      strUserCode ; // 사용자코드
extern CString      strUserName ; // 사용자명칭
extern CString      strPgmNo      ; // 프로그램코드
extern CString      strKEY        ; // 키보드 입력값
extern CString      strKEYTITLE ; // 키보드입력 타이틀
extern int          keyYS          ; // 키보드입력후 1확인/0취소
extern int          KeyBoardOK ; //(0:패스워드,1:금액,수량) ;
```

```

extern int      keyLength  ; // 키보드입력길이
extern HBRUSH brush ;
extern CFont font;

// 선별관리에서 사용될 변수선언 (필드명칭을 따른다..)
CString SEMQ_SEL_SER ; // 선별일련번호
CString SEMQ_BUY_SER ; // 구매일련번호
CString SEMQ_STORE_SER; // 입고일련번호
CString SEMQ_SORT_DT ; // 선별일자

CString SEMQ_FARM_CD ; // 농가코드
CString SEMQ_FARM_NAME; // 농가명
CString SEMQ_ITEM_CD ; // 품목코드
CString SEMQ_ITEM_NAME; // 품목명
CString SEMQ_CONT_DT ; // 약정일자

CString SEM_MMODE ; // 프로그램모드 (신규, 수정)

////////////////////////////////////
// CSEM_Q0001 dialog

CSEM_Q0001::CSEM_Q0001(CWnd* pParent /*=NULL*/)
    : CDialog(CSEM_Q0001::IDD, pParent)
{
    //{{AFX_DATA_INIT(CSEM_Q0001)
    m_username = _T("");
    m_gname = _T("");

```

```

        m_pname = _T("");
        //}}AFX_DATA_INIT
    }

// 화면을 풀스크린으로 만든다..
void CSEM_Q0001:: Full_ScreenOK()
{
    // START 화면을 전체화면으로 키운다..
    SHINITDLGINFO shidi;
    shidi.dwMask = SHIDIM_FLAGS;
    shidi.dwFlags = SHIDIF_FULLSCREENNOMENUBAR | SHIDIF_SIPDOWN;
    shidi.hDlg = m_hWnd;
    SHInitDialog(&shidi);
    ::CommandBar_Show(m_pWndEmptyCB->m_hWnd, FALSE);
    SetForegroundWindow();
    ::SHFullScreen(m_hWnd, SHFS_HIDETASKBAR | SHFS_HIDESIPBUTTON |
SHFS_HIDESTARTICON);
    MoveWindow(0, 0, 240, 360, TRUE);
    // END 화면을 전체화면으로 키운다..
}

void CSEM_Q0001::ComboAddShow()
{
    CString strSQL;
    HRESULT hr = S_OK;

```

```

long lcount;
long nState;
long option = adCmdText;
COleVariant varFields;
COleVariant varSQL;
COleVariant varConn;
CursorTypeEnum cte = adOpenDynamic;
LockTypeEnum lte = adLockOptimistic;

IADOCEField *field = NULL;
IADOCEFields *fields = NULL;
CString CheckStr ;
CheckStr = _T(""); // 중복검사

// 농가정보를 콤보박스에 등록한다.. //
strSQL.Format(_T("SELECT Farm_name FROM TM_STORE WHERE User_yn = '%s'
ORDER BY Farm_name DESC "),L"Y") ;

varSQL.SetString(strSQL,VT_BSTR);
varConn.SetString(strDBConn,VT_BSTR);

m_pRecordset->get_State(&nState);
if(nState==1)m_pRecordset->Close();
hr = m_pRecordset->Open(varSQL,varConn,cte,lte,option);
if(FAILED(hr))
{
    MessageBox(_T("DB나 쿼리문에 이상이 있습니다..."));
}

```

```

        Full_ScreenOK() ;
        return;
    }

hr = m_pRecordset->get_RecordCount(&lcount); // 레코드의 개수를 얻는다.
if(lcount==0)
{
    MessageBox(_T("DB나 쿼리문에 이상이 있습니다..."));
    Full_ScreenOK() ;
    return ; // 레코드가 없으면 되돌아간다.
}

COleVariant itemFields ; // 필드이름을 저장할 변수

COleVariant itemFARM_NAME; // 농가명칭
CString strOutput ; // 리스트에 추가할 문자열

c_gname.AddString(L"" ) ;
for(int i=0;i<lcount;i++)
{
    // 레코드셋으로부터 Fields 컬렉션 인터페이스를 얻는다.
    hr = m_pRecordset->get_Fields(&fields);

    // -- 농가명칭 --- //
    itemFields.SetString(_T("Farm_name"),VT_BSTR);
    hr = fields->get_Item(itemFields,&field);
    hr = field->get_Value(&itemFARM_NAME);
    c_gname.AddString(itemFARM_NAME.bstrVal) ;
    // 다음레코드로 넘어간다.
    hr = m_pRecordset->MoveNext();
}

```

```

    }           // 레코드의 개수만큼 루프를 돌린다.
    field ->Release(); // Field와 Fields 오브젝트의
    fields->Release();// 인터페이스를 해제한다.
    field = 0;
    fields = 0;
    hr = m_pRecordset->Close();//레코드셋을 닫는다.
// 농가정보를 콤보박스에 등록한다.. //

// 품목정보를 콤보박스에 등록한다.. //
strSQL.Format(_T("SELECT Item_name FROM TM_ITEM WHERE User_yn = '%s' ORDER
BY Item_name DESC "),L"Y");

varSQL.SetString(strSQL,VT_BSTR);
varConn.SetString(strDBConn,VT_BSTR);

m_pRecordset->get_State(&nState);
if(nState==1)m_pRecordset->Close();
hr = m_pRecordset->Open(varSQL,varConn,cte,lte,option);
if(FAILED(hr))
{
    MessageBox(_T("DB나 쿼리문에 이상이 있습니다..."));
    Full_ScreenOK() ;
    return;
}
// 레코드의 개수를 얻는다.
hr = m_pRecordset->get_RecordCount(&lcount);

```



```

if(lcount==0)
{
    MessageBox(_T("DB나 쿼리문에 이상이 있습니다..."));
    Full_ScreenOK() ;
    return ; // 레코드가 없으면 되돌아간다.
}

COleVariant itemITEM_NAME; // 품목명칭

c_pname.AddString(L"" ) ;
for(int j=0;j<lcount;j++)
{
    // 레코드셋으로부터 Fields 컬렉션 인터페이스를 얻는다.
    hr = m_pRecordset->get_Fields(&fields);

    // -- 품목명칭 --- //
    itemFields.SetString(_T("Item_name"),VT_BSTR);
    hr = fields->get_Item(itemFields,&field);
    hr = field->get_Value(&itemITEM_NAME);
    c_pname.AddString(itemITEM_NAME.bstrVal) ;
    // 다음레코드로 넘어간다.
    hr = m_pRecordset->MoveNext();
} // 레코드의 개수만큼 루프를 돌린다.
field ->Release(); // Field와 Fields 오브젝트의
fields->Release();// 인터페이스를 해제한다.
field = 0;
fields = 0;

```

```

        hr = m_pRecordset->Close();//레코드셋을 닫는다.
// 품목정보를 콤보박스에 등록한다.. //

        UpdateData(false);
}

void CSEM_Q0001::ListTitle()
{
    int nColumnCount = m_list.GetHeaderCtrl()->GetItemCount();
    for (int j=0;j < nColumnCount;j++)
    {
        m_list.DeleteColumn(0);
    }

    m_list.SetExtendedStyle(LVS_EX_FULLROWSELECT|LVS_EX_GRIDLINES);
    int intWid[5];
    intWid[0] = 36 ; // 선별일련번호
    intWid[1] = 40 ; // 수매일련번호
    intWid[2] = 45 ; // 선별일자
    intWid[3] = 50 ; // 품목
    intWid[4] = 60 ; // 농가

    TCHAR col[5][40] = {_T("번호"),_T("번호"),_T("일자"),_T("품목"),_T("농가")};
    LV_COLUMN lvcolumn;

    while(m_list.DeleteColumn(0));
}

```

```

        for (int i = 0 ; i < 5 ;i++)
        {
lvcolumn.mask = LVCF_FMT | LVCF_SUBITEM | LVIF_TEXT | LVCF_WIDTH;

        lvcolumn.fmt = LVCFMT_CENTER;

        lvcolumn.pszText = col[i];

        lvcolumn.iSubItem = i;

        lvcolumn.cx = intWid[i]; // SubItem is twice as large

        m_list.InsertColumn(i, &lvcolumn); // assumes return value is OK.}

        }
}

void CSEM_Q0001::ListShow()
{

        SetCursor(AfxGetApp()->LoadCursor(IDC_WAIT));

        m_list.DeleteAllItems() ;

// 작업변수를 설정한다.-----

        CString strSQL;

        HRESULT hr = S_OK;

        long lcount;

        long nState;

        long option = adCmdText;

        COleVariant varFields;

        COleVariant varSQL;

        COleVariant varConn;

        CursorTypeEnum cte = adOpenDynamic;

        LockTypeEnum lte = adLockOptimistic;

```

```

IADOCEField          *field    = NULL;
IADOCEFields         *fields   = NULL;
CString CheckStr ;
CheckStr = _T(""); // 중복검사

```

```

strSQL.Format(_T("SELECT Sel_ser,Buy_ser, Sort_dt, Item_name, Farm_name FROM
TBL_SEL WHERE Next_OK = '%s' ORDER BY Sel_ser DESC"),L"N");

```

```

varSQL.SetString(strSQL,VT_BSTR);
varConn.SetString(strDBConn,VT_BSTR);

```

```

m_pRecordset->get_State(&nState);
if(nState==1)m_pRecordset->Close();
hr = m_pRecordset->Open(varSQL,varConn,cte,lte,option);
if(FAILED(hr))
{
    SetCursor(AfxGetApp()->LoadCursor(IDC_NO));
    Full_ScreenOK() ;
    return;
}

```

// 레코드의 개수를 얻는다.

```

hr = m_pRecordset->get_RecordCount(&lcount);
if(lcount==0)
{
    SetCursor(AfxGetApp()->LoadCursor(IDC_NO));

    Full_ScreenOK() ;
    return ; // 레코드가 없으면 되돌아간다.
}

```

```

}

COleVariant itemFields          ; // 필드이름을 저장할 변수

COleVariant itemgSel_ser       ; // 선별일련번호
COleVariant itemgBuy_ser       ; // 구매일련번호
COleVariant itemeSort_dt       ; // 선별일자
COleVariant itemeItem_name     ; // 품목명
COleVariant itemeFarm_name     ; // 상품명

CString  strOutput             ; // 리스트에 추가할 문자열
CString  strDD                 ; // 구매일자를 mm.dd로 변경할 변수
LV_ITEM  lvitem                ; // 리스트에 등록될 단속내역

int      workok = 0 ;
int      workno = 0 ;
for(int i=0;i<lcount;i++)
{
    // 레코드셋으로부터 Fields 컬렉션 인터페이스를 얻는다.
    hr = m_pRecordset->get_Fields(&fields);

    // -- 선별일련번호 --- //
    itemFields.SetString(_T("Sel_ser"),VT_BSTR);
    hr = fields->get_Item(itemFields,&field);
    hr = field->get_Value(&itemgSel_ser);
    lvitem.iSubItem = 0;
    lvitem.mask = LVIF_TEXT;
    lvitem.pszText = itemgSel_ser.bstrVal;
    lvitem.iItem = 0;
}

```

```

m_list.InsertItem(&lvitem);

// -- 구매일련번호 --- //
itemFields.SetString(_T("Buy_ser"),VT_BSTR);
hr = fields->get_Item(itemFields,&field);
hr = field->get_Value(&itemgBuy_ser);
lvitem.iSubItem = 1;
lvitem.mask = LVIF_TEXT;
lvitem.pszText = itemgBuy_ser.bstrVal;
lvitem.iItem = 0;
m_list.SetItem(&lvitem);

// -- 선별일자 --- //
itemFields.SetString(_T("Sort_dt"),VT_BSTR);
hr = fields->get_Item(itemFields,&field);
hr = field->get_Value(&itemeSort_dt);
lvitem.iSubItem = 2;
lvitem.mask = LVIF_TEXT;

strDD = itemeSort_dt.bstrVal ;
strDD = strDD.Mid(5,2) + L"." + strDD.Mid(8,2) ;
lvitem.pszText = (LPTSTR)strDD.operator LPCTSTR();
lvitem.iItem = 0;
m_list.SetItem(&lvitem);

// -- 품목명 --- //

```

```

itemFields.SetString(_T("Item_name"),VT_BSTR);
hr = fields->get_Item(itemFields,&field);
hr = field->get_Value(&itemItem_name);
lvitem.iSubItem = 3;
lvitem.mask = LVIF_TEXT;
lvitem.pszText = itemItem_name.bstrVal;
lvitem.iItem = 0;
m_list.SetItem(&lvitem);

// -- 상품명 --- //
itemFields.SetString(_T("Farm_name"),VT_BSTR);
hr = fields->get_Item(itemFields,&field);
hr = field->get_Value(&itemFarm_name);
lvitem.iSubItem = 4;
lvitem.mask = LVIF_TEXT;
lvitem.pszText = itemFarm_name.bstrVal;
lvitem.iItem = 0;
m_list.SetItem(&lvitem);
// 다음레코드로 넘어간다.
hr = m_pRecordset->MoveNext();
CheckStr = strOutput ;
} // 레코드의 개수만큼 루프를 돌린다.
UpdateData(false);

field ->Release(); // Field와 Fields 오브젝트의
fields->Release();// 인터페이스를 해제한다.
field = 0;

```

```

        fields = 0;

        hr = m_pRecordset->Close();//레코드셋을 닫는다.

        SetCursor(AfxGetApp()->LoadCursor(IDC_NO));
    }

```

```

void CSEM_Q0001::DoDataExchange(CDataExchange* pDX)
{
    CDialog::DoDataExchange(pDX);
    //{{AFX_DATA_MAP(CSEM_Q0001)
    DDX_Control(pDX, COMBO_PNAME, c_pname);
    DDX_Control(pDX, COMBO_GNAME, c_gname);
    DDX_Control(pDX, IDC_Q0001_MBTN5, c_btn5);
    DDX_Control(pDX, IDC_Q0001_MBTN4, c_btn4);
    DDX_Control(pDX, IDC_Q0001_MBTN3, c_btn3);
    DDX_Control(pDX, IDC_Q0001_MBTN2, c_btn2);
    DDX_Control(pDX, IDC_Q0001_MBTN1, c_btn1);
    DDX_Control(pDX, IDC_LIST1, m_list);
    DDX_Text(pDX, IDC_C_USERNAME, m_username);
    DDX_CBString(pDX, COMBO_GNAME, m_gname);
    DDX_CBString(pDX, COMBO_PNAME, m_pname);
    //}}AFX_DATA_MAP
}

```

```

BEGIN_MESSAGE_MAP(CSEM_Q0001, CDialog)

```



```

//{{AFX_MSG_MAP(CSEM_Q0001)
ON_NOTIFY(NM_CLICK, IDC_LIST1, OnClickList1)
ON_BN_CLICKED(IDC_Q0001_MBTN1, OnQ0001Mbtn1)
ON_BN_CLICKED(IDC_Q0001_MBTN4, OnQ0001Mbtn4)
ON_BN_CLICKED(IDC_Q0001_MBTN5, OnQ0001Mbtn5)
ON_BN_CLICKED(IDC_Q0001_MBTN2, OnQ0001Mbtn2)
ON_BN_CLICKED(IDC_Q0001_MBTN3, OnQ0001Mbtn3)
//}}AFX_MSG_MAP
END_MESSAGE_MAP()

////////////////////////////////////

// CSEM_Q0001 message handlers

void CSEM_Q0001::OnClickList1(NMHDR* pNMHDR, LRESULT* pResult)
{
// LIST의 자료를 선택하면 선택된 선별일련번호를 기억한다.
    NM_LISTVIEW* pList = (NM_LISTVIEW*)pNMHDR;
    ASSERT(pList);
    *pResult = 0;

    if(pList->iItem == -1)
    {
        SEMQ_SEL_SER = L"" ;
        SEMQ_BUY_SER = L"" ;
        SEMQ_STORE_SER = L"" ;

        return;
    }
}

```

```

    SEMQ_SEL_SER = m_list.GetItemText(pList->iItem,0); // 선별일련번호
    SEMQ_BUY_SER = m_list.GetItemText(pList->iItem,1); // 수매일련번호
    SEMQ_STORE_SER = m_list.GetItemText(pList->iItem,0); // 입고일련번호
        UpdateData(false);
}

BOOL CSEM_Q0001::OnInitDialog()
{
    CDialog::OnInitDialog();

    m_username = strUserName ;

    Full_ScreenOK() ;
    ListTitle() ;
    ListShow() ;
    ComboAddShow() ; // 품목과 농가를 콤보박스에 등록한다.
    //PK인 수매일련번호를 초기화한다.(선택 수매일련번호)
    SEMQ_SEL_SER = L"" ;

    // 사용자명의 글자체를 크게한다.
font.CreateFont(18, 0, 0, 0, FW_BOLD, FALSE, FALSE, 0, 0, 0, 0, 0, 0, _T("Arial"));
    ((CStatic*)this->GetDlgItem(IDC_C_USERNAME))->SetFont(&font);
    // TODO: Add extra initialization here
    return TRUE; // return TRUE unless you set the focus to a control
}

void CSEM_Q0001::OnQ0001Mbtn1()

```

```

{
// 조회버튼을 클릭 하였을때 선택된 농가와, 품목을 기준으로 선별관리내역을 불러온다..
    UpdateData(true);

    SetCursor(AfxGetApp()->LoadCursor(IDC_WAIT));
    m_list.DeleteAllItems() ;
    // 작업변수를 설정한다..-----
    CString strSQL;
    HRESULT hr = S_OK;
    long lcount;
    long nState;
    long option = adCmdText;
    COleVariant varFields;
    COleVariant varSQL;
    COleVariant varConn;
    CursorTypeEnum cte = adOpenDynamic;
    LockTypeEnum lte = adLockOptimistic;

    IADOCEField *field = NULL;
    IADOCEFields *fields = NULL;
    CString CheckStr ;
    CheckStr = _T(""); // 중복검사

    // 상품과 거래처를 선택하지 않았다..
    if (m_pname == L"" && m_gname == L"")
    {
strSQL.Format(_T("SELECT Sel_ser, Buy_ser, Sort_dt, Item_name, Farm_name FROM

```

```

TBL_SEL WHERE Next_OK = '%s' ORDER BY Buy_ser DESC"),L"N") ;
    }
    // 상품과 거래처를 모두 선택했다.
    else if (m_pname != L"" && m_gname != L"")
    {
strSQL.Format(T("SELECT Sel_ser, Buy_ser, Sort_dt, Item_name, Farm_name FROM
TBL_SEL WHERE Next_OK = '%s' AND Item_name = '%s' AND Farm_name = '%s'
ORDER BY Buy_ser DESC"),L"N",m_pname,m_gname) ;
    }
    // 거래처를 선택했다.
    else if (m_pname == L"")
    {
strSQL.Format(T("SELECT Sel_ser, Buy_ser, Sort_dt, Item_name, Farm_name FROM
TBL_SEL WHERE Next_OK = '%s' AND Farm_name = '%s' ORDER BY Buy_ser
DESC"),L"N",m_gname) ;
    }
    // 상품을 선택했다.
    else if (m_gname == L"")
    {
strSQL.Format(T("SELECT Sel_ser, Buy_ser, Sort_dt, Item_name, Farm_name FROM
TBL_SEL WHERE Next_OK = '%s' AND Item_name = '%s' ORDER BY Buy_ser
DESC"),L"N",m_pname) ;
    }

varSQL.SetString(strSQL,VT_BSTR);
varConn.SetString(strDBConn,VT_BSTR);

```

```

m_pRecordset->get_State(&nState);
if(nState==1)m_pRecordset->Close();
hr = m_pRecordset->Open(varSQL,varConn,cte,lte,option);
if(FAILED(hr))
{
    SetCursor(AfxGetApp()->LoadCursor(IDC_NO));
    Full_ScreenOK() ;
    return;
}
hr = m_pRecordset->get_RecordCount(&lcount); // 레코드의 개수를 얻는다.
if(lcount==0)
{
    SetCursor(AfxGetApp()->LoadCursor(IDC_NO));
    Full_ScreenOK() ;
    return ; // 레코드가 없으면 되돌아간다.
}

```

```

COleVariant itemFields ; // 필드이름을 저장할 변수

```

```

COleVariant itemgSel_ser ; // 선별일련번호

```

```

COleVariant itemgBuy_ser ; // 구매일련번호

```

```

COleVariant itemeSort_dt ; // 선별일자

```

```

COleVariant itemeItem_name ; // 품목명

```

```

COleVariant itemeFarm_name ; // 상품명

```

```

CString strOutput ; // 리스트에 추가할 문자열

```

```

CString strDD ; // 구매일자를 mm.dd로 변경할 변수

```

```

LV_ITEM lvitem      ; // 리스트에 등록될 단속내역

int   workok = 0 ;
int   workno = 0 ;
for(int i=0;i<lcount;i++)
{
    // 레코드셋으로부터 Fields 컬렉션 인터페이스를 얻는다.
    hr = m_pRecordset->get_Fields(&fields);

    // -- 선별일련번호 --- //
    itemFields.SetString(_T("Sel_ser"),VT_BSTR);
    hr = fields->get_Item(itemFields,&field);
    hr = field->get_Value(&itemgSel_ser);
    lvitem.iSubItem = 0;
    lvitem.mask = LVIF_TEXT;
    lvitem.pszText = itemgSel_ser.bstrVal;
    lvitem.iItem = 0;
    m_list.InsertItem(&lvitem);

    // -- 구매일련번호 --- //
    itemFields.SetString(_T("Buy_ser"),VT_BSTR);
    hr = fields->get_Item(itemFields,&field);
    hr = field->get_Value(&itemgBuy_ser);
    lvitem.iSubItem = 1;
    lvitem.mask = LVIF_TEXT;
    lvitem.pszText = itemgBuy_ser.bstrVal;
    lvitem.iItem = 0;
    m_list.SetItem(&lvitem);
}

```

```

// -- 선별일자 --- //
itemFields.SetString(_T("Sort_dt"),VT_BSTR);
hr = fields->get_Item(itemFields,&field);
hr = field->get_Value(&itemSort_dt);
lvitem.iSubItem = 2;
lvitem.mask = LVIF_TEXT;

strDD = itemSort_dt.bstrVal ;
strDD = strDD.Mid(5,2) + L"." + strDD.Mid(8,2) ;
lvitem.pszText = (LPTSTR)strDD.operator LPCTSTR();
lvitem.iItem = 0;
m_list.SetItem(&lvitem);

```

```

// -- 품목명 --- //
itemFields.SetString(_T("Item_name"),VT_BSTR);
hr = fields->get_Item(itemFields,&field);
hr = field->get_Value(&itemItem_name);
lvitem.iSubItem = 3;
lvitem.mask = LVIF_TEXT;
lvitem.pszText = itemItem_name.bstrVal;
lvitem.iItem = 0;
m_list.SetItem(&lvitem);

```

```

// -- 상품명 --- //
itemFields.SetString(_T("Farm_name"),VT_BSTR);
hr = fields->get_Item(itemFields,&field);

```

```

        hr = field->get_Value(&itemeFarm_name);
        lvitem.iSubItem = 4;
        lvitem.mask = LVIF_TEXT;
        lvitem.pszText = itemeFarm_name.bstrVal;
        lvitem.iItem = 0;
        m_list.SetItem(&lvitem);
        // 다음레코드로 넘어간다.
        hr = m_pRecordset->MoveNext();
        CheckStr = strOutput ;
    }        // 레코드의 개수만큼 루프를 돌린다.
    UpdateData(false);

```

```

    field ->Release(); // Field와 Fields 오브젝트의
    fields->Release();// 인터페이스를 해제한다.
    field = 0;
    fields = 0;
    hr = m_pRecordset->Close();//레코드셋을 닫는다.

```

```

    SetCursor(AfxGetApp()->LoadCursor(IDC_NO));

```

```

}

```

```

void CSEM_Q0001::OnQ0001Mbtn4()

```

```

{

```

```

// 선별관리 삭제

```

```

    UpdateData(true);

```

```

    if(SEMQ_SEL_SER == L"")

```



```

    {
        MessageBox(_T("삭제하실 선별관리내역을 선택하세요!!"));
        Full_ScreenOK() ;
        return;
    }
else
{
int nAnswer = MessageBox(_T("선택하신 선별관리내역을 삭제할까요?"),_T("선별관리내역 삭제
"), MB_ICONINFORMATION|MB_OKCANCEL);
    if(nAnswer == IDOK)
    {
        CString strSQL;
        HRESULT hr = S_OK;
        HRESULT hd = S_OK;
        HRESULT hb = S_OK;
        long lcount;
        long nState;
        long option = adCmdText;
        COleVariant varFields;
        COleVariant varSQL;
        COleVariant varConn;
        CursorTypeEnum cte = adOpenDynamic;
        LockTypeEnum lte = adLockOptimistic;
        IADOCEField *field = NULL;
        IADOCEFields *fields = NULL;
        CString CheckStr ;
        CheckStr = _T(""); // 중복검사
    }
}

```

```

strSQL.Format(_T("DELETE TBL_SEL WHERE Sel_ser='%s'"),SEMQ_SEL_SER) ;
hd = m_pConnection->Execute(strSQL.GetBuffer(0),NULL,adCmdText,NULL) ;
        if(FAILED(hd))
        {
                MessageBox(_T("선별관리마스터내역 삭제에 실패하였습니다..")) ;
                Full_ScreenOK() ;
                return ;
        }

strSQL.Format(_T("DELETE TBL_SEL_DETAIL WHERE Sel_ser = '%s'"),SEMQ_SEL_SER) ;
hd = m_pConnection->Execute(strSQL.GetBuffer(0),NULL,adCmdText,NULL) ;
        if(FAILED(hd))
        {
                MessageBox(_T("선별관리상세내역 삭제에 실패하였습니다..")) ;
                Full_ScreenOK() ;
                return ;
        }

//-----
// 수매입고내역의 Next_ok를 "Y"로 업데이트 한다.

// 1. 수매입고내역을 가져와 변수에 등록한다.
strSQL.Format(_T("SELECT * FROM TBL_STORE WHERE Buy_ser = '%s'"),SEMQ_BUY_SER) ;

varSQL.SetString(strSQL,VT_BSTR);
varConn.SetString(strDBConn,VT_BSTR);

```

```

m_pRecordset->get_State(&nState);
if(nState==1)m_pRecordset->Close();
hr = m_pRecordset->Open(varSQL,varConn,cte,lte,option);
if(FAILED(hr))
{
    SetCursor(AfxGetApp()->LoadCursor(IDC_NO));
    Full_ScreenOK() ;
    return;
}
// 레코드의 개수를 얻는다.
hr = m_pRecordset->get_RecordCount(&lcount);
if(lcount==0)    return    ; // 레코드가 없으면 되돌아간

```

다.

```

CString CCMat_opt,CCBuy_dt, CCStore_dt, CCCont_dt, CCAredcd ;
int CCQuanti, CCAmt, CCKg ;
ColeVariant itemFields    ; // 필드이름을 저장할 변

ColeVariant itemeMat_opt  ; // 원료상품구분
ColeVariant itemeBuy_dt   ; // 구매일자
ColeVariant itemeStore_dt ; // 입고일자
ColeVariant itemeCont_dt  ; // 약정일자
ColeVariant itemeAredcd   ; // 지역코드
ColeVariant itemeQuanti   ; // 수량
ColeVariant itemeAmt      ; // 금액
ColeVariant itemeKg       ; // 중량

// 레코드셋으로부터 Fields 컬렉션 인터페이스를 얻는다.
hr = m_pRecordset->get_Fields(&fields);

```

수

```
// -- 원료상품구분 --- //
```

```
itemFields.SetString(_T("Mat_opt"),VT_BSTR);
```

```
hr = fields->get_Item(itemFields,&field);
```

```
hr = field->get_Value(&itemeMat_opt);
```

```
CCMat_opt = itemeMat_opt.bstrVal ;
```

```
// -- 구매일자 --- //
```

```
itemFields.SetString(_T("Buy_dt"),VT_BSTR);
```

```
hr = fields->get_Item(itemFields,&field);
```

```
hr = field->get_Value(&itemeBuy_dt);
```

```
CCBuy_dt = itemeBuy_dt.bstrVal ;
```

```
// -- 입고일자 --- //
```

```
itemFields.SetString(_T("Store_dt"),VT_BSTR);
```

```
hr = fields->get_Item(itemFields,&field);
```

```
hr = field->get_Value(&itemeStore_dt);
```

```
CCStore_dt = itemeStore_dt.bstrVal ;
```

```
// -- 약정일자 --- //
```

```
itemFields.SetString(_T("Cont_dt"),VT_BSTR);
```

```
hr = fields->get_Item(itemFields,&field);
```

```
hr = field->get_Value(&itemeCont_dt);
```

```
CCCont_dt = itemeCont_dt.bstrVal ;
```

```
// -- 지역코드 --- //
```

```
itemFields.SetString(_T("Aredcd"),VT_BSTR);
```

```
hr = fields->get_Item(itemFields,&field);
```

```
hr = field->get_Value(&itemeAredcd);
```

```
CCAredcd = itemeAredcd.bstrVal ;
```

```
// -- 수량 --- //
```

```
itemFields.SetString(_T("Quanti"),VT_BSTR);
```

```
hr = fields->get_Item(itemFields,&field);
```

```
hr = field->get_Value(&itemeQuanti);
```

```
CCQuanti = itemeQuanti.intVal ;
```

```
// -- 금액 --- //
```

```
itemFields.SetString(_T("Amt"),VT_BSTR);
```

```
hr = fields->get_Item(itemFields,&field);
```

```
hr = field->get_Value(&itemeAmt);
```

```
CCAmt = itemeAmt.intVal ;
```

```
// -- 중량 --- //
```

```
itemFields.SetString(_T("Kg"),VT_BSTR);
```

```
hr = fields->get_Item(itemFields,&field);
```

```
hr = field->get_Value(&itemeKg);
```

```
CCKg = itemeKg.intVal ;
```

```

UpdateData(false);

field ->Release(); // Field와 Fields 오브젝트의
fields->Release();// 인터페이스를 해제한다.

field = 0;
fields = 0;

hr = m_pRecordset->Close();//레코드셋을 닫는다.

//-----

// 2. 수매입고내역중 해당 수매내역을 삭제한다..
strSQL.Format(_T("DELETE TBL_STORE WHERE Buy_ser = '%s'",SEM_Q_BUY_SER) ;
hd = m_pConnection->Execute(strSQL.GetBuffer(0),NULL,adCmdText,NULL) ;
if(FAILED(hd))
{
MessageBox(_T("수매입고내역 삭제에 실패하였습니다..")) ;
Full_ScreenOK() ;
return ;
}

//-----

//자료생성,수정일자 저장을 위해
CTime tDate = CTime::GetCurrentTime() ;
CString syear ;

```

```

CString smon ;
CString sday ;
CString shh ;
CString sdd ;

syear.Format(_T("%.4d"),tDate.GetYear() ) ;
smon. Format(_T("%.2d"),tDate.GetMonth() ) ;
sday. Format(_T("%.2d"),tDate.GetDay() ) ;
shh. Format(_T("%.2d"),tDate.GetHour() ) ;
sdd. Format(_T("%.2d"),tDate.GetMinute() ) ;
CString WorkDD ;

WorkDD = syear + L"- " + smon + L"- " + sday + L" " + shh + L":" + sdd ;

// 3. 구매입고내역을 저장한다..
// 자료를 저장한다..

strSQL.Format(_T("INSERT INTO TBL_STORE (Buy_ser, Mat_opt, Buy_dt, Item_cd,
Item_name, Farm_cd, Farm_name, Store_dt, Cont_dt, Areacd, Quanti, Amt, Kg, RegID,
Create_dt, Update_dt,Next_OK) VALUES
('%s','%s','%s','%s','%s','%s','%s','%s','%s','%s','%d','%d','%d','%s','%s','%s','%s')"),SEM
Q_BUY_SER, CCMat_opt, CCBuy_dt, SEMQ_ITEM_CD, SEMQ_ITEM_NAME,
SEMQ_FARM_CD, SEMQ_FARM_NAME, CCStore_dt,CCCont_dt, CCAreacd, CCQuanti,
CCAmt, CCKg, strUserCode, WorkDD, WorkDD, L"Y" ) ;
hb = m_pConnection->Execute(strSQL.GetBuffer(0),NULL,adCmdText,NULL) ;
if(FAILED(hb))
{
MessageBox(_T("자료저장에 실패하였습니다..")) ;
Full_ScreenOK() ;
}

```

```

        return ;
    }

//-----
    // 구매입고내역의 Next_ok를 "Y"로 업데이트 한다..
    }
    Full_ScreenOK() ;
}
ListShow() ;
UpdateData(false);
}

void CSEM_Q0001::OnQ0001Mbtn5()
{
// 선별마스타조회를 종료한다..
    OnOK() ;
}

void CSEM_Q0001::OnQ0001Mbtn2()
{
// 선별마스타를 등록한다..

    CString strSQL;
    HRESULT hr = S_OK;
    long lcount;
    long nState;
    long option = adCmdText;

```



```

COleVariant varFields;
COleVariant varSQL;
COleVariant varConn;
CursorTypeEnum cte = adOpenDynamic;
LockTypeEnum lte = adLockOptimistic;

IADOCEField *field = NULL;
IADOCEFields *fields = NULL;
CString CheckStr ;
CheckStr = _T(""); // 중복검사

```

// 신규 선별일련번호를 생성한다.

```

strSQL.Format(_T("SELECT Sel_ser FROM TBL_SEL ORDER BY Sel_ser DESC"));
varSQL.SetString(strSQL,VT_BSTR);
varConn.SetString(strDBConn,VT_BSTR);

m_pRecordset->get_State(&nState);
if(nState==1)m_pRecordset->Close();
hr = m_pRecordset->Open(varSQL,varConn,cte,lte,option);
hr = m_pRecordset->get_RecordCount(&lcount); // 레코드의 개수를 얻는다.
if(lcount==0)
{
    // 값이 없으면 수매일련번호를 0001로 지정한다..
    SEMQ_SEL_SER = L"0001" ;
}

COleVariant itemFields ; // 필드이름을 저장할 변수
COleVariant itemSEL_SER ; // 선별일련번호

```

```

CString    strOutput ; // 리스트에 추가할 문자열

// 레코드셋으로부터 Fields 컬렉션 인터페이스를 얻는다.
hr = m_pRecordset->get_Fields(&fields);

// -- 선별일련번호 --- //
itemFields.SetString(_T("Sel_ser"),VT_BSTR);
hr = fields->get_Item(itemFields,&field);
hr = field->get_Value(&itemSEL_SER);
strOutput = itemSEL_SER.bstrVal ;
int buyno ;
buyno = _wtoi(strOutput) ;
SEM_Q_SEL_SER.Format(_T("%0.4d"),buyno+1) ;
// -- 선별일련번호 --- //

field ->Release(); // Field와 Fields 오브젝트의
fields->Release();// 인터페이스를 해제한다.
field = 0;
fields = 0;
hr = m_pRecordset->Close();//레코드셋을 닫는다.

SEM_MMODE = L"신규" ;
UpdateData(false);
// 신규 선별일련번호를 생성한다..

// 변수 초기화

```

```

//입고일련번호는 선별일련번호를 따라간다.
SEMQ_STORE_SER = SEMQ_SEL_SER ;
SEMQ_BUY_SER   = L"" ;
SEMQ_SORT_DT   = L"" ;
SEMQ_FARM_CD   = L"" ;
SEMQ_FARM_NAME = L"" ;
SEMQ_ITEM_CD   = L"" ;
SEMQ_ITEM_NAME = L"" ;
SEMQ_CONT_DT   = L"" ;

// 변수 초기화

// 선별마스타 등록화면을 VIEW
    CSEM_T0001 T0001Dlg ;
    T0001Dlg.DoModal() ;
    Full_ScreenOK() ;

// 선별마스타 등록화면을 VIEW
    ListShow() ;
}

void CSEM_Q0001::OnQ0001Mbtn3()
{
// 선별마스타 수정
    if(SEMQ_SEL_SER == L"")
    {
        MessageBox(_T("수정하실 선별마스타 내역을 선택하세요!!"));
        Full_ScreenOK() ;
    }
}

```

```

        return;
    }
    else
    {
//-----
        // 선택한 선별일련번호를 이용하여 수정할 선별관리내역을 변수에 등록한다..
//-----

        UpdateData(true);

        SetCursor(AfxGetApp()->LoadCursor(IDC_WAIT));
        // 작업변수를 설정한다..-----
        CString strSQL;
        HRESULT hr = S_OK;
        long lcount;
        long nState;
        long option = adCmdText;
        COleVariant varFields;
        COleVariant varSQL;
        COleVariant varConn;
        CursorTypeEnum cte = adOpenDynamic;
        LockTypeEnum lte = adLockOptimistic;

        IADOCEField *field = NULL;
        IADOCEFields *fields = NULL;
        CString CheckStr ;

        strSQL.Format(_T("SELECT * FROM TBL_SEL WHERE SEL_SER = '%s'

```

"),SEMQ_SEL_SER) ;

```
varSQL.SetString(strSQL,VT_BSTR);
varConn.SetString(strDBConn,VT_BSTR);

m_pRecordset->get_State(&nState);
if(nState==1)m_pRecordset->Close();
hr = m_pRecordset->Open(varSQL,varConn,cte,lte,option);
if(FAILED(hr))
{
    SetCursor(AfxGetApp()->LoadCursor(IDC_NO));
    Full_ScreenOK() ;
    return;
}
// 레코드의 개수를 얻는다.
hr = m_pRecordset->get_RecordCount(&lcount);
if(lcount==0)
{
    SetCursor(AfxGetApp()->LoadCursor(IDC_NO));

    Full_ScreenOK() ;
    return ; // 레코드가 없으면 되돌아간다.
}

COleVariant itemFields ; // 필드이름을 저장할 변
수

COleVariant itemeSort_dt ; // 선별일자
```

수

```

COleVariant itemCont_dt ; // 약정일자
// 레코드셋으로부터 Fields 컬렉션 인터페이스를 얻는다.
hr = m_pRecordset->get_Fields(&fields);

// -- 선별일자 ---- //
itemFields.SetString(_T("Sort_dt"),VT_BSTR);
hr = fields->get_Item(itemFields,&field);
hr = field->get_Value(&itemSort_dt);
SEMQ_SORT_DT = itemSort_dt.bstrVal ;

// -- 약정일자 --- //
itemFields.SetString(_T("Cont_dt"),VT_BSTR);
hr = fields->get_Item(itemFields,&field);
hr = field->get_Value(&itemCont_dt);
SEMQ_CONT_DT = itemCont_dt.bstrVal ;

UpdateData(false);
field ->Release(); // Field와 Fields 오브젝트의
fields->Release();// 인터페이스를 해제한다.
field = 0;
fields = 0;
hr = m_pRecordset->Close();//레코드셋을 닫는다.
SetCursor(AfxGetApp()->LoadCursor(IDC_NO));

```

//-----

```

}
SEMQ_MMODE = L"수정" ;

```

```

        // 수매입고 등록화면을 VIEW
        CSEM_T0001 T0001Dlg ;
        T0001Dlg.DoModal() ;
        Full_ScreenOK() ;
        ListShow() ;
        UpdateData(false);
    }

```

라. 포장관리 모듈

```

// PAC_Q0001.cpp : implementation file
//
#include "stdafx.h"
#include "resource.h"
#include "exfarmDlg.h"
#include "PAC_T0001.h"
#include "PAC_Q0001.h"
#include "MOBILE_KEYBOARD.h"

#ifdef _DEBUG
#define new DEBUG_NEW
#undef THIS_FILE
static char THIS_FILE[] = __FILE__;
#endif

extern IADOCERRecordset* m_pRecordset;

```

```

extern IADOCEConnection* m_pConnection;

extern CString      strDBConn      ;
extern CString      strUserCode ; // 사용자코드
extern CString      strUserName ; // 사용자명칭
extern CString      strPgmNo      ; // 프로그램코드
extern CString      strKEY        ; // 키보드 입력값
extern CString      strKEYTITLE ; // 키보드입력 타이틀
extern int          keyYS          ; // 키보드입력후 1확인/0취소
extern int          KeyBoardOK ; //(0:패스워드,1:금액,수량) ;
extern int          keyLength     ; // 키보드입력길이
extern HBRUSH brush ;
extern CFont font;

// 포장관리에서 사용될 변수선언(필드명칭을 따른다..)
extern CString PAC_SEL_SER      ; // 선별일련번호
extern CString PAC_PACK_SER     ; // 포장일련번호
extern CString PAC_PACK_DT      ; // 포장일자
extern CString PAC_GOODCD       ; // 상품코드
extern CString PAC_STORECD      ; // 입고처코드
extern int      PAC_AMT         ; // 금액
extern CString PAC_GRADE_CD     ; // 등급
extern int      PAC_QUANTI      ; // 수량
extern int      PAC_UNIT_QTY    ; // 박스
extern CString PAC_MMODE ;      // 프로그램모드 (신규, 수정)
////////////////////////////////////
// CPAC_Q0001 dialog

```



```

CString KEYComma2(int dNum)
{
    int i, j;
    CString strTmp;
    CString strNum;

    strTmp.Format(_T("%d"), dNum);
    for(i = strTmp.GetLength() - 1, j = 1; i >= 0; i--, j++)
    {
        if( ((j % 3) == 1) && (j > 3) )
        {
            strNum.Insert(0, ',');
            strNum.Insert(0, strTmp.GetAt(i));
        }
        else
        {
            strNum.Insert(0, strTmp.GetAt(i));
        }
    }
    return strNum;
}

```

```

CPAC_Q0001::CPAC_Q0001(CWnd* pParent /*=NULL*/)
    : CDialog(CPAC_Q0001::IDD, pParent)
{
    //{{AFX_DATA_INIT(CPAC_Q0001)
    m_s01 = _T("");

```

```

        m_s02 = _T("");
        m_s04 = _T("");
        m_s05 = _T("");
        m_s06 = _T("");
        m_s07 = _T("");
        m_s08 = _T("");
        m_s09 = _T("");
        m_s03 = COleDateTime::GetCurrentTime();
        //}}AFX_DATA_INIT
    }

void CPAC_Q0001::Full_ScreenOK()
{
    // START 화면을 전체화면으로 키운다..
    SHINITDLGINFO shidi;
    shidi.dwMask = SHIDIM_FLAGS;
    shidi.dwFlags = SHIDIF_FULLSCREENNOMENUBAR | SHIDIF_SIPDOWN;
    shidi.hDlg = m_hWnd;
    SHInitDialog(&shidi);
    ::CommandBar_Show(m_pWndEmptyCB->m_hWnd, FALSE);
    SetForegroundWindow();
    ::SHFullScreen(m_hWnd, SHFS_HIDETASKBAR | SHFS_HIDESIPBUTTON |
SHFS_HIDESTARTICON);
    MoveWindow(0, 0, 240, 360, TRUE);
    // END 화면을 전체화면으로 키운다..
}

```

```

void CPAC_Q0001::ComboAddShow()
{
    CString strSQL;
    HRESULT hr = S_OK;

    long lcount;
    long nState;
    long option = adCmdText;
    COleVariant varFields;
    COleVariant varSQL;
    COleVariant varConn;
    CursorTypeEnum cte = adOpenDynamic;
    LockTypeEnum lte = adLockOptimistic;

    IADOCEField          *field = NULL;
    IADOCEFields         *fields = NULL;
    CString CheckStr ;
    CheckStr = _T(""); // 중복검사

    // 선별일련번호를 등록한다.. 구분값이 N인것만.. //
    strSQL.Format(_T("SELECT Sel_ser FROM TBL_SEL WHERE NEXT_OK = '%s' ORDER
    BY Sel_ser DESC "),L"N");

    varSQL.SetString(strSQL,VT_BSTR);
    varConn.SetString(strDBConn,VT_BSTR);

    m_pRecordset->get_State(&nState);
    if(nState==1)m_pRecordset->Close();
}

```

```

hr = m_pRecordset->Open(varSQL,varConn,cte,lte,option);
if(FAILED(hr))
{
    MessageBox(_T("DB나 쿼리문에 이상이 있습니다..."));
    Full_ScreenOK() ;
    return;
}
// 레코드의 개수를 얻는다.
hr = m_pRecordset->get_RecordCount(&lcount);
if(lcount==0)
{
    MessageBox(_T("DB나 쿼리문에 이상이 있습니다..."));
    Full_ScreenOK() ;
    return ; // 레코드가 없으면 되돌아간다.
}
COleVariant itemFields ; // 필드이름을 저장할 변수

COleVariant itemSEL_SER ; // 선별일련번호
CString strOutput ; // 리스트에 추가할 문자열

for(int i=0;i<lcount;i++)
{
    // 레코드셋으로부터 Fields 컬렉션 인터페이스를 얻는다.
    hr = m_pRecordset->get_Fields(&fields);

    // -- 선별일련번호 --- //
    itemFields.SetString(_T("Sel_ser"),VT_BSTR);
    hr = fields->get_Item(itemFields,&field);
}

```

```

        hr = field->get_Value(&itemSEL_SER);
c_s02.AddString(itemSEL_SER.bstrVal) ;

        if(i == 0)
        {
                m_s02 = itemSEL_SER.bstrVal ;
        }
        // 다음레코드로 넘어간다.
        hr = m_pRecordset->MoveNext();
    } // 레코드의 개수만큼 루프를 돌린다.
    field ->Release(); // Field와 Fields 오브젝트의
    fields->Release();// 인터페이스를 해제한다.
    field = 0;
    fields = 0;
    hr = m_pRecordset->Close();//레코드셋을 닫는다.
// 선별일련번호를 등록한다.. 구분값이 N인것만.. //
UpdateData(false);
}

```

```

void CPAC_Q0001::DoDataExchange(CDataExchange* pDX)
{
    CDialog::DoDataExchange(pDX);
    //{{AFX_DATA_MAP(CPAC_Q0001)
    DDX_Control(pDX, IDC_S02, c_s02);
    DDX_Control(pDX, IDC_Q0001_MBTN8, c_btn2);
    DDX_Control(pDX, IDC_Q0001_MBTN7, c_btn1);
    DDX_Text(pDX, IDC_S01, m_s01);

```

```

        DDX_CBString(pDX, IDC_S02, m_s02);
        DDX_Text(pDX, IDC_S04, m_s04);
        DDX_Text(pDX, IDC_S05, m_s05);
        DDX_Text(pDX, IDC_S08, m_s06);
        DDX_Text(pDX, IDC_S09, m_s07);
        DDX_Text(pDX, IDC_S10, m_s08);
        DDX_Text(pDX, IDC_S11, m_s09);
        DDX_DateTimeCtrl(pDX, IDC_S03, m_s03);
    //}}AFX_DATA_MAP
}

BEGIN_MESSAGE_MAP(CPAC_Q0001, CDialog)
    //{{AFX_MSG_MAP(CPAC_Q0001)
    ON_BN_CLICKED(IDC_Q0001_MBTN8, OnQ0001Mbtn8)
    ON_BN_CLICKED(IDC_P04, OnP04)
    ON_BN_CLICKED(IDC_P05, OnP05)
    ON_BN_CLICKED(IDC_P06, OnP06)
    ON_BN_CLICKED(IDC_P07, OnP07)
    ON_BN_CLICKED(IDC_P08, OnP08)
    ON_BN_CLICKED(IDC_P09, OnP09)
    ON_BN_CLICKED(IDC_Q0001_MBTN7, OnQ0001Mbtn7)
    //}}AFX_MSG_MAP
END_MESSAGE_MAP()

////////////////////////////////////
// CPAC_Q0001 message handlers

```

```

void CPAC_Q0001::OnQ0001Mbtn8()
{
    OnOK() ;           // TODO: Add your control notification handler code here
}

BOOL CPAC_Q0001::OnInitDialog()
{
    CDialog::OnInitDialog();

    // 포장일련번호를 등록한다..
    m_s01 = PAC_PACK_SER ;
    // 포장일련번호의 크게한다..
    font.CreateFont(18, 0, 0, 0, FW_BOLD, FALSE, FALSE, 0, 0, 0, 0, 0, 0, _T("Arial"));
    ((CStatic*)this->GetDlgItem(IDC_S01))->SetFont(&font);

    if(PAC_MMODE == L"신규")
    {
        ComboAddShow() ;
        // 날짜를 해당일자로 //
        CTime tDate = CTime::GetCurrentTime() ;
        CString syear ;
        CString smon ;
        CString sday ;
        syear.Format(_T("%.4d"),tDate.GetYear()) ;
        smon.Format(_T("%.2d"),tDate.GetMonth()) ;
        sday.Format(_T("%.2d"),tDate.GetDay()) ;
    }
}

```

```

        int intyear = _wtoi(syear) ;
        int intmon  = _wtoi(smon)  ;
        int intday  = _wtoi(sday)  ;
        m_s03.SetDate(intyear,intmon,intday) ; //포장일자
        // 날짜를 해당일자로
    }
    else // 수정일 경우
    {
        m_s02 = PAC_SEL_SER ;           // 선별일련번호

m_s03.SetDate(_wtoi(PAC_PACK_DT.Mid(0,4)),_wtoi(PAC_PACK_DT.Mid(5,2)),_wtoi(PAC_PAC
K_DT.Mid(8,2))) ; //포장일자

        m_s04 = PAC_GOODCD ;           // 상품코드
        m_s05 = PAC_STORECD;           // 입고처
        m_s06 = KEYComma2(PAC_AMT) ;    // 금액
        m_s07 = PAC_GRADE_CD ;         // 등급
        m_s08 = KEYComma2(PAC_QUANTI) ; // 수량
        m_s09 = KEYComma2(PAC_UNIT_QTY) ;// 박스
    }

    UpdateData(false)    ;
    Full_ScreenOK() ;
    return TRUE; // return TRUE unless you set the focus to a control
                // EXCEPTION: OCX Property Pages should return FALSE
}

void CPAC_Q0001::OnP04()
{

```



```

// 상품코드 등록
        CMOBILE_KEYBOARD KeyBoardDlg ;
        UpdateData(true);

        strKEY      = PAC_GOODCD ;
        strKEYTITLE= L"상품코드" ;
        KeyBoardOK = 0 ;
        keyLength  = 4 ;
        KeyBoardDlg.DoModal() ;
        m_s04 = strKEY;
        PAC_GOODCD = strKEY ;

        UpdateData(false);
        Full_ScreenOK() ;
    }

```

```

void CPAC_Q0001::OnP05()
{
// 입고처 등록
        CMOBILE_KEYBOARD KeyBoardDlg ;
        UpdateData(true);

        strKEY      = PAC_STORECD ;
        strKEYTITLE= L"입고처코드" ;
        KeyBoardOK = 0 ;
        keyLength  = 6 ;
        KeyBoardDlg.DoModal() ;

```

```

        m_s05 = strKEY;
        PAC_STORECD = strKEY ;

        UpdateData(false);
        Full_ScreenOK() ;
    }

void CPAC_Q0001::OnP06()
{
    // 금액 등록
        CMOBILE_KEYBOARD KeyBoardDlg ;
        UpdateData(true);

        strKEY.Format(_T("%d"),PAC_AMT) ;
        strKEYTITLE= L"금 액" ;
        KeyBoardOK = 1 ;
        keyLength = 10 ;
        KeyBoardDlg.DoModal() ;
        PAC_AMT = _wtoi(strKEY) ;
        m_s06 = KEYComma2(PAC_AMT) ;

        UpdateData(false);
        Full_ScreenOK() ;
}

void CPAC_Q0001::OnP07()
{

```

```

// 등급 등록

CMOBILE_KEYBOARD KeyBoardDlg ;
UpdateData(true);

strKEY      = PAC_GRADE_CD ;
strKEYTITLE= L"등급   급" ;
KeyBoardOK = 0 ;
keyLength  = 7 ;
KeyBoardDlg.DoModal() ;
m_s07 = strKEY;
PAC_GRADE_CD = strKEY ;

UpdateData(false);
Full_ScreenOK() ;
}

void CPAC_Q0001::OnP08()
{
// 수량 등록

CMOBILE_KEYBOARD KeyBoardDlg ;
UpdateData(true);

strKEY.Format(_T("%d"),PAC_QUANTI) ;
strKEYTITLE= L"수   량" ;
KeyBoardOK = 1 ;
keyLength  = 10 ;
KeyBoardDlg.DoModal() ;
}

```

```

PAC_QUANTI = _wtoi(strKEY) ;
m_s08 = KEYComma2(PAC_QUANTI) ;

UpdateData(false);
Full_ScreenOK() ;
}

void CPAC_Q0001::OnP09()
{
// 박스 등록
CMOBILE_KEYBOARD KeyBoardDlg ;
UpdateData(true);

strKEY.Format(_T("%d"),PAC_UNIT_QTY) ;
strKEYTITLE= L"박스 스" ;
KeyBoardOK = 1 ;
keyLength = 10 ;
KeyBoardDlg.DoModal() ;
PAC_UNIT_QTY = _wtoi(strKEY) ;
m_s09 = KEYComma2(PAC_UNIT_QTY) ;

UpdateData(false);
Full_ScreenOK() ;
}

void CPAC_Q0001::OnQ0001Mbtn7()
{

```

```

UpdateData(true);
// 포장관리 등록/수정내역을 저장한다.. //
CString strSQL;
HRESULT hr = S_OK;
HRESULT hd = S_OK;
HRESULT hb = S_OK;
long lcount;
long nState;
long option = adCmdText;
COleVariant varFields;
COleVariant varSQL;
COleVariant varConn;
CursorTypeEnum cte = adOpenDynamic;
LockTypeEnum lte = adLockOptimistic;
IADOCEField *field = NULL;
IADOCEFields *fields = NULL;
CString CheckStr ;
CheckStr = _T(""); // 중복검사

// 수정일때 기존자료를 삭제한다.
if(PAC_MMODE == L"수정")
{
strSQL.Format(_T("DELETE TBL_PACK WHERE PACK_SER = '%s'"),PAC_PACK_SER) ;
hd = m_pConnection->Execute(strSQL.GetBuffer(0),NULL,adCmdText,NULL) ;
if(FAILED(hd))
{
MessageBox(_T("이전자료 삭제에 실패하였습니다..")) ;
}
}

```

```

        Full_ScreenOK() ;
        return ;
    }
}

// 변수값을 확인하여 등록한다..
PAC_SEL_SER = m_s02 ;
PAC_PACK_DT  .Format(_T("%4d-%2d-%2d"), m_s03.GetYear(),
m_s03.GetMonth(), m_s03.GetDay()) ;
PAC_GOODCD  = m_s04 ;
PAC_STORECD = m_s05 ;
PAC_GRADE_CD= m_s07 ;

//자료생성,수정일자 저장을 위해
CTime tDate = CTime::GetCurrentTime() ;
CString syear ;
CString smon  ;
CString sday  ;
CString shh   ;
CString sdd   ;
syear.Format(_T("%4d"),tDate.GetYear()) ;
smon. Format(_T("%2d"),tDate.GetMonth()) ;
sday. Format(_T("%2d"),tDate.GetDay()) ;
shh. Format(_T("%2d"),tDate.GetHour()) ;
sdd. Format(_T("%2d"),tDate.GetMinute()) ;

```

```

CString WorkDD ;
WorkDD = syear + L"- " + smon + L"- " + sday + L" " + shh + L":" + sdd ;

// 자료를 저장한다..
strSQL.Format(_T("INSERT INTO TBL_PACK (SEL_SER, PACK_SER, PACK_DT, GOODCD,
STORECD, AMT, GRADE_CD, QUANTI, UNIT_QTY, RegID, Create_dt, Update_dt) VALUES
('%s', '%s', '%s', '%s', '%s', '%d', '%s', '%d', '%d', '%s', '%s', '%s')"), PAC_SEL_SER,
PAC_PACK_SER, PAC_PACK_DT, PAC_GOODCD, PAC_STORECD, PAC_AMT,
PAC_GRADE_CD, PAC_QUANTI, PAC_UNIT_QTY, strUserCode, WorkDD, WorkDD) ;
hb = m_pConnection->Execute(strSQL.GetBuffer(0), NULL, adCmdText, NULL) ;
if(FAILED(hb))
{
    MessageBox(_T("포장관리 자료저장에 실패하였습니다..")) ;
    Full_ScreenOK() ;
    return ;
}
// 포장관리 등록/수정내역을 저장한다.. //

//-----
// 선별관리내역의 Next_ok를 "Y"로 업데이트 한다..

// 1. 선별관리내역을 가져와 변수에 등록한다..
strSQL.Format(_T("SELECT * FROM TBL_SEL WHERE SEL_SER = '%s' "), PAC_SEL_SER)
;

varSQL.SetString(strSQL, VT_BSTR);
varConn.SetString(strDBConn, VT_BSTR);

```

```

m_pRecordset->get_State(&nState);

if(nState==1)m_pRecordset->Close();
hr = m_pRecordset->Open(varSQL,varConn,cte,lte,option);
if(FAILED(hr))
{
    SetCursor(AfxGetApp()->LoadCursor(IDC_NO));
    Full_ScreenOK() ;
    return;
}
// 레코드의 개수를 얻는다.
hr = m_pRecordset->get_RecordCount(&lcount);
if(lcount==0)      return ; // 레코드가 없으면 되돌아간

```

다.

```

CString CSEL_SER, CBUY_SER, CSTORE_SER, CSORT_DT, CFARM_CD, CFARM_NAME,
CITEM_CD, CITEM_NAME, CCONT_DT ;

```

수

```

ColeVariant itemFields      ; // 필드이름을 저장할 변

ColeVariant itemeSEL_SER   ; // 선별일련번호
ColeVariant itemeBUY_SER   ; // 구매일련번호
ColeVariant itemeSTORE_SER ; // 입고일련번호
ColeVariant itemeSORT_DT   ; // 선별일자
ColeVariant itemeFARM_CD   ; // 농가코드

ColeVariant itemeFARM_NAME ; // 농가명

```



```

ColeVariant itemeITEM_CD ; // 품목코드
ColeVariant itemeITEM_NAME ; // 품목명
ColeVariant itemeCONT_DT ; // 약정일자

// 레코드셋으로부터 Fields 컬렉션 인터페이스를 얻는다.
hr = m_pRecordset->get_Fields(&fields);

// -- 선별일련번호 --- //
itemFields.SetString(_T("SEL_SER"),VT_BSTR);
hr = fields->get_Item(itemFields,&field);
hr = field->get_Value(&itemeSEL_SER);
CSEL_SER = itemeSEL_SER.bstrVal ;

// -- 구매일련번호 --- //
itemFields.SetString(_T("BUY_SER"),VT_BSTR);
hr = fields->get_Item(itemFields,&field);
hr = field->get_Value(&itemeBUY_SER);
CBUY_SER = itemeBUY_SER.bstrVal ;

// -- 입고일련번호 --- //
itemFields.SetString(_T("STORE_SER"),VT_BSTR);
hr = fields->get_Item(itemFields,&field);
hr = field->get_Value(&itemeSTORE_SER);
CSTORE_SER = itemeSTORE_SER.bstrVal ;

// -- 선별일자 --- //
itemFields.SetString(_T("SORT_DT"),VT_BSTR);

```

```

hr = fields->get_Item(itemFields,&field);
hr = field->get_Value(&itemeSORT_DT);
CSORT_DT = itemeSORT_DT.bstrVal ;

// -- 농가코드 --- //
itemFields.SetString(_T("FARM_CD"),VT_BSTR);
hr = fields->get_Item(itemFields,&field);
hr = field->get_Value(&itemeFARM_CD);
CFARM_CD = itemeFARM_CD.bstrVal ;

// -- 농가명 --- //
itemFields.SetString(_T("FARM_NAME"),VT_BSTR);
hr = fields->get_Item(itemFields,&field);
hr = field->get_Value(&itemeFARM_NAME);
CFARM_NAME = itemeFARM_NAME.bstrVal ;

// -- 품목코드 --- //
itemFields.SetString(_T("ITEM_CD"),VT_BSTR);
hr = fields->get_Item(itemFields,&field);
hr = field->get_Value(&itemeITEM_CD);
CITEM_CD = itemeITEM_CD.bstrVal ;

// -- 품목명 --- //
itemFields.SetString(_T("ITEM_NAME"),VT_BSTR);
hr = fields->get_Item(itemFields,&field);
hr = field->get_Value(&itemeITEM_NAME);
CITEM_NAME = itemeITEM_NAME.bstrVal ;

```

```

// -- 약정일자 --- //
itemFields.SetString(_T("CONT_DT"),VT_BSTR);
hr = fields->get_Item(itemFields,&field);
hr = field->get_Value(&itemCONT_DT);
CCONT_DT = itemCONT_DT.bstrVal ;

UpdateData(false);

field ->Release(); // Field와 Fields 오브젝트의
fields->Release();// 인터페이스를 해제한다.
field = 0;
fields = 0;
hr = m_pRecordset->Close();//레코드셋을 닫는다.

//-----

// 2. 선별관리 내역중 해당내역을 삭제한다.
strSQL.Format(_T("DELETE TBL_SEL WHERE SEL_SER = '%s'"),PAC_SEL_SER) ;
hd = m_pConnection->Execute(strSQL.GetBuffer(0),NULL,adCmdText,NULL) ;
if(FAILED(hd))
{
    MessageBox(_T("선별관리내역 삭제에 실패하였습니다..")) ;
    Full_ScreenOK() ;
    return ;
}

```

//-----

//자료생성,수정일자 저장을 위해

syear.Format(_T("%4d"),tDate.GetYear()) ;

smon. Format(_T("%2d"),tDate.GetMonth()) ;

sday. Format(_T("%2d"),tDate.GetDay()) ;

shh. Format(_T("%2d"),tDate.GetHour()) ;

sdd. Format(_T("%2d"),tDate.GetMinute()) ;

WorkDD = syear + L"- " + smon + L"- " + sday + L" " + shh + L":" + sdd ;

// 3. 선별관리 내역을 저장한다..

```
strSQL.Format(_T("INSERT INTO TBL_SEL (Sel_ser, Buy_ser, Store_ser, Sort_dt, Farm_cd, Farm_name, Item_cd, Item_name, Cont_dt, RegID, Create_dt, Update_dt, Next_OK) VALUES ('%s','%s','%s','%s','%s','%s','%s','%s','%s','%s','%s','%s','%s','%s')"),CSEL_SER, CBUY_SER, CSTORE_SER, CSORT_DT, CFARM_CD, CFARM_NAME, CITEM_CD, CITEM_NAME, CCONT_DT, strUserCode, WorkDD, WorkDD, L"Y" );
```

```
hr = m_pConnection->Execute(strSQL.GetBuffer(0),NULL,adCmdText,NULL) ;
```

```
if(FAILED(hr))
```

```
{
```

```
    MessageBox(_T("선별관리내역 자료저장에 실패하였습니다..")) ;
```

```
    Full_ScreenOK() ;
```

```
    return ;
```

```
}
```

//-----

// 선별관리의 Next_ok를 "Y"로 업데이트 한다..

```
UpdateData(false);  
OnOK() ;  
}
```

마. 출하관리 모듈

```
// SHP_Q0001.cpp : implementation file
```

```
//
```

```
#include "stdafx.h"
```

```
#include "resource.h"
```

```
#include "exfarmDlg.h"
```

```
#include "SHP_T0001.h"
```

```
#include "MOBILE_KEYBOARD.h"
```

```
#include "SHP_Q0001.h"
```

```
#ifdef _DEBUG
```

```
#define new DEBUG_NEW
```

```
#undef THIS_FILE
```

```
static char THIS_FILE[] = __FILE__;
```

```
#endif
```

```
extern IADOCERecordset* m_pRecordset;
```

```
extern IADOCEConnection* m_pConnection;
```

```
extern CString      strDBConn      ;
```

```
extern CString      strUserCode ; // 사용자코드
```

```
extern CString      strUserName ; // 사용자명칭
```

```
extern CString      strPgmNo   ; // 프로그램코드
```

```

extern CString    strKEY      ; // 키보드 입력값
extern CString    strKEYTITLE ; // 키보드입력 타이틀
extern int        keyYS      ; // 키보드입력후 1확인/0취소
extern int        KeyBoardOK ; //(0:패스워드,1:금액,수량) ;
extern int        keyLength  ; // 키보드입력길이
extern HBRUSH     brush ;
extern CFont      font;

// 출하관리에서 사용될 변수선언 (필드명칭을 따른다..)
extern CString    SHP_ORDER_SER    ; // 주문번호
extern CString    SHP_ORD_DT      ; // 주문일자
extern CString    SHP_GOODCD      ; // 상품코드
extern CString    SHP_CHUL_END_OPT ; // 주문확정구분

extern CString    SHP_PDA_CONFIRM ; // PDA출하확정승인
extern CString    SHP_CHULHA_DT   ; // 출하일자
extern CString    SHP_CHULHA_TYPE ; // 출하형태
extern CString    SHP_CHULHA_MOD_DT; // 출하정정요청일자
extern CString    SHP_CHULHA_MOD_OPT; // 출하정정구분
extern CString    SHP_MOD_OPT     ; // 정정사유

extern int        SHP_ORD_QTY ;      // 주문수량
extern int        SHP_CHULHA_QTY ;  // 출하수량
extern CString    SHP_MODE ;      // 프로그램모드 (신규, 수정)
////////////////////////////////////
// CSHP_Q0001 dialog

```

```

CSHP_Q0001::CSHP_Q0001(CWnd* pParent /*=NULL*/)
    : CDialog(CSHP_Q0001::IDD, pParent)
{
   //{{AFX_DATA_INIT(CSHP_Q0001)
    m_s01 = _T("");
    m_s03 = _T("");
    m_s04 = _T("");
    m_s07 = COleDateTime::GetCurrentTime();
    m_s02 = COleDateTime::GetCurrentTime();
    m_s08 = _T("");
    m_s10 = COleDateTime::GetCurrentTime();
    m_s09 = _T("");
    m_s11 = _T("");
    m_s12 = _T("");
    //}}AFX_DATA_INIT
}

```

```

void CSHP_Q0001::Full_ScreenOK()
{
    // START 화면을 전체화면으로 키운다..
    SHINITDLGINFO shidi;
    shidi.dwMask = SHIDIM_FLAGS;
    shidi.dwFlags = SHIDIF_FULLSCREENNOMENUBAR | SHIDIF_SIPDOWN;
    shidi.hDlg = m_hWnd;
    SHInitDialog(&shidi);
    ::CommandBar_Show(m_pWndEmptyCB->m_hWnd, FALSE);
}

```

```

SetForegroundWindow();

::SHFullScreen(m_hWnd, SHFS_HIDETASKBAR | SHFS_HIDESIPBUTTON |
SHFS_HIDESTARTICON);

MoveWindow(0, 0, 240, 360, TRUE);

// END 화면을 전체화면으로 키운다..
}

```

```

void CSHP_Q0001::DoDataExchange(CDataExchange* pDX)

```

```

{
    CDialog::DoDataExchange(pDX);
   //{{AFX_DATA_MAP(CSHP_Q0001)
    DDX_Control(pDX, IDC_CK2_2, c_chk2_2);
    DDX_Control(pDX, IDC_CK2_1, c_chk2_1);
    DDX_Control(pDX, IDC_CK1_2, c_chk1_2);
    DDX_Control(pDX, IDC_CK1_1, c_chk1_1);
    DDX_Text(pDX, IDC_SH01, m_s01);
    DDV_MaxChars(pDX, m_s01, 4);
    DDX_Text(pDX, IDC_SH03, m_s03);
    DDV_MaxChars(pDX, m_s03, 6);
    DDX_Text(pDX, IDC_SH04, m_s04);
    DDX_DateTimeCtrl(pDX, IDC_SH07, m_s07);
    DDX_DateTimeCtrl(pDX, IDC_SH02, m_s02);
    DDX_Text(pDX, IDC_SH08, m_s08);
    DDX_DateTimeCtrl(pDX, IDC_SH10, m_s10);
    DDX_CBString(pDX, IDC_SH09, m_s09);
    DDX_CBString(pDX, IDC_SH11, m_s11);
}

```



```

        DDX_CBString(pDX, IDC_SH12, m_s12);
    //}}AFX_DATA_MAP
}

BEGIN_MESSAGE_MAP(CSHP_Q0001, CDialog)
    //{{AFX_MSG_MAP(CSHP_Q0001)
    ON_BN_CLICKED(IDC_Q0001_MBTN5, OnQ0001Mbtn5)
    ON_BN_CLICKED(IDC_Q0001_MBTN1, OnQ0001Mbtn1)
    ON_BN_CLICKED(IDC_P3, OnP3)
    ON_BN_CLICKED(IDC_P4, OnP4)
    ON_BN_CLICKED(IDC_P8, OnP8)
    ON_BN_CLICKED(IDC_P9, OnP9)
    ON_BN_CLICKED(IDC_P11, OnP11)
    ON_BN_CLICKED(IDC_P12, OnP12)
    ON_BN_CLICKED(IDC_CK1_1, OnCk11)
    ON_BN_CLICKED(IDC_CK1_2, OnCk12)
    ON_BN_CLICKED(IDC_CK2_1, OnCk21)
    ON_BN_CLICKED(IDC_CK2_2, OnCk22)
    //}}AFX_MSG_MAP

```

```

END_MESSAGE_MAP()

```

```

////////////////////////////////////

```

```

// CSHP_Q0001 message handlers

```

```

void CSHP_Q0001::OnQ0001Mbtn5()

```

```

{

```

```

        OnOK() ;
    }

void CSHP_Q0001::OnQ0001Mbtn1()
{
    UpdateData(true);

    // 출서관리 등록/수정내역을 저장한다. //
    CString strSQL;
    HRESULT hr = S_OK;
    HRESULT hd = S_OK;
    HRESULT hb = S_OK;
    long option = adCmdText;
    COleVariant varFields;
    COleVariant varSQL;
    COleVariant varConn;
    CursorTypeEnum cte = adOpenDynamic;
    LockTypeEnum lte = adLockOptimistic;
    IADOCEField          *field = NULL;
    IADOCEFields         *fields = NULL;
    CString CheckStr ;
    CheckStr = _T(""); // 중복검사

    // 수정일때 기존자료를 삭제한다.
    if(SHP_MODE == L"수정")
    {
        strSQL.Format(_T("DELETE TBL_SHIP WHERE ORDER_SER = '%s'"),SHP_ORDER_SER) ;
    }
}

```

```

hd = m_pConnection->Execute(strSQL.GetBuffer(0),NULL,adCmdText,NULL) ;

        if(FAILED(hd))
        {
                MessageBox(_T("이전자료 삭제에 실패하였습니다..")) ;
                Full_ScreenOK() ;
                return ;
        }
}

// 변수값을 확인하여 등록한다..

SHP_ORD_DT.Format(_T("%4d-%2d-%2d"),m_s02.GetYear(),m_s02.GetMonth(),m_s02.GetDay()
) ;

SHP_GOODCD      = m_s03 ;
SHP_ORD_QTY     = _wtoi(m_s04) ;

SHP_CHULHA_DT.Format(_T("%4d-%2d-%2d"),m_s07.GetYear(),m_s07.GetMonth(),m_s07.Get
Day() ) ;

SHP_CHULHA_QTY  = _wtoi(m_s08) ;
SHP_CHULHA_TYPE = m_s09 ;

SHP_CHULHA_MOD_DT.Format(_T("%4d-%2d-%2d"),m_s10.GetYear(),m_s10.GetMonth(),m_s
10.GetDay() ) ;

SHP_CHULHA_MOD_OPT= m_s11 ;
SHP_MOD_OPT      = m_s12 ;

```

```

//자료생성,수정일자 저장을 위해
CTime tDate = CTime::GetCurrentTime() ;
CString syear ;
CString smon ;
CString sday ;
CString shh ;
CString sdd ;

syear.Format(_T("%4d"),tDate.GetYear()) ;
smon. Format(_T("%2d"),tDate.GetMonth()) ;
sday. Format(_T("%2d"),tDate.GetDay()) ;
shh. Format(_T("%2d"),tDate.GetHour()) ;
sdd. Format(_T("%2d"),tDate.GetMinute()) ;

CString WorkDD ;
WorkDD = syear + L"- " + smon + L"- " + sday + L" " + shh + L":" + sdd ;

// 자료를 저장한다.
strSQL.Format(_T("INSERT INTO TBL_SHIP (ORDER_SER, ORD_DT, GOODCD,
CHUL_END_OPT, PDA_CONFIRM, CHULHA_DT, CHULHA_TYPE, CHULHA_MOD_DT,
CHULHA_MOD_OPT, MOD_OPT, ORD_QTY, CHULHA_QTY, RegID, Create_dt, Update_dt)
V A L U E S
('%s','%s','%s','%s','%s','%s','%s','%s','%s','%s','%d','%d','%s','%s','%s')"),SHP_ORDER_S
ER, SHP_ORD_DT, SHP_GOODCD, SHP_CHUL_END_OPT, SHP_PDA_CONFIRM,
SHP_CHULHA_DT, SHP_CHULHA_TYPE, SHP_CHULHA_MOD_DT,
SHP_CHULHA_MOD_OPT, SHP_MOD_OPT, SHP_ORD_QTY, SHP_CHULHA_QTY,
strUserCode, WorkDD, WorkDD) ;
hb = m_pConnection->Execute(strSQL.GetBuffer(0),NULL,adCmdText,NULL) ;

```

```

        if(FAILED(hb))
        {
            MessageBox(_T("출하관리 자료저장에 실패하였습니다..") );
            Full_ScreenOK() ;
            return ;
        }
// 출하관리 등록/수정내역을 저장한다.. //

    UpdateData(false);
    OnOK() ;
}

BOOL CSHP_Q0001::OnInitDialog()
{
    CDialog::OnInitDialog();

    // 포장일련번호를 등록한다..
    m_s01 = SHP_ORDER_SER ;
    // 포장일련번호의 크게한다..
font.CreateFont(18, 0, 0, 0, FW_BOLD, FALSE, FALSE, 0, 0, 0, 0, 0, 0, _T("Arial"));
    ((CStatic*)this->GetDlgItem(IDC_SH01))->SetFont(&font);

    if(SHP_MODE == L"신규")
    {
        // 날짜를 해당일자로 //
        CTime tDate = CTime::GetCurrentTime() ;
        CString syear ;

```

```

CString smon ;
CString sday ;
syear.Format(_T("%4d"),tDate.GetYear() ) ;
smon. Format(_T("%2d"),tDate.GetMonth() ) ;
sday. Format(_T("%2d"),tDate.GetDay() ) ;
int intyear = _wtoi(syear) ;
int intmon  = _wtoi(smon)  ;
int intday  = _wtoi(sday)  ;
m_s02.SetDate(intyear,intmon,intday) ; //주문일자
m_s07.SetDate(intyear,intmon,intday) ; //출하일자
m_s10.SetDate(intyear,intmon,intday) ; //출하정정요청일자
// 날짜를 해당일자로
}
else // 수정일 경우
{

    if(SHP_CHUL_END_OPT == L"Y")
    {
        c_chk1_1.SetCheck(1) ;
        c_chk1_2.SetCheck(0) ;
    }
    else
    {
        c_chk1_1.SetCheck(0) ;
        c_chk1_2.SetCheck(1) ;
    }
}

```

```

if(SHP_PDA_CONFIRM == L"Y")
{
    c_chk2_1.SetCheck(1);
    c_chk2_2.SetCheck(0);
}
else
{
    c_chk2_1.SetCheck(0);
    c_chk2_2.SetCheck(1);
}

```

```

m_s02.SetDate(_wtoi(SHP_ORD_DT.Mid(0,4)),_wtoi(SHP_ORD_DT.Mid(5,2)),_wtoi(SHP_ORD_DT
.Mid(8,2))); //주문일자

```

```

m_s03 = SHP_GOODCD ; // 상품코드
m_s04.Format(_T("%d"),SHP_ORD_QTY); // 주문수량

```

```

m_s07.SetDate(_wtoi(SHP_CHULHA_DT.Mid(0,4)),_wtoi(SHP_CHULHA_DT.Mid(5,2)),_wtoi(SHP
_CHULHA_DT.Mid(8,2))); //출하일자

```

```

m_s08.Format(_T("%d"),SHP_CHULHA_QTY); // 출하수량
m_s09 = SHP_CHULHA_TYPE ; // 출하형태

```

```

m_s10.SetDate(_wtoi(SHP_CHULHA_MOD_DT.Mid(0,4)),_wtoi(SHP_CHULHA_MOD_DT.Mid(5,2
)),_wtoi(SHP_CHULHA_MOD_DT.Mid(8,2))); //출하정정요청일자

```

```

m_s11 = SHP_CHULHA_MOD_OPT ; // 출하정정구분
m_s12 = SHP_MOD_OPT ; // 정정사유

```

```

    }

    UpdateData(false)    ;
    Full_ScreenOK() ;

    return TRUE; // return TRUE unless you set the focus to a control
                // EXCEPTION: OCX Property Pages should return FALSE
}

```

```

void CSHP_Q0001::OnP3()
{
    // 상품코드 등록

    CMOBILE_KEYBOARD KeyBoardDlg ;
    UpdateData(true);

    strKEY      = SHP_GOODCD ;
    strKEYTITLE= L"상품코드" ;
    KeyBoardOK = 0 ;
    keyLength  = 6 ;
    KeyBoardDlg.DoModal() ;
    m_s03 = strKEY;
    SHP_GOODCD = strKEY ;

    UpdateData(false);
    Full_ScreenOK() ;
}

```

```

void CSHP_Q0001::OnP4()
{

```



```

// 주문수량 등록

CMOBILE_KEYBOARD KeyBoardDlg ;
UpdateData(true);

strKEY.Format(_T("%d"),SHP_ORD_QTY) ;
strKEYTITLE= L"주문수량" ;
KeyBoardOK = 1 ;
keyLength = 10 ;
KeyBoardDlg.DoModal() ;
m_s04 = strKEY;
SHP_ORD_QTY = _wtoi(strKEY) ;

UpdateData(false);
Full_ScreenOK() ;
}

void CSHP_Q0001::OnP8()
{
// 출하수량 등록

CMOBILE_KEYBOARD KeyBoardDlg ;
UpdateData(true);

strKEY.Format(_T("%d"),SHP_CHULHA_QTY) ;
strKEYTITLE= L"출하수량" ;
KeyBoardOK = 1 ;
keyLength = 10 ;
KeyBoardDlg.DoModal() ;
}

```

```

        m_s08 = strKEY;
        SHP_CHULHA_QTY = _wtoi(strKEY) ;

        UpdateData(false);
        Full_ScreenOK() ;
    }

```

```

void CSHP_Q0001::OnP9()
{
    // 출하형태 등록
    CMOBILE_KEYBOARD KeyBoardDlg ;
    UpdateData(true);

    strKEY      = SHP_CHULHA_TYPE ;
    strKEYTITLE= L"출하형태" ;
    KeyBoardOK = 0 ;
    keyLength  = 11 ;
    KeyBoardDlg.DoModal() ;
    m_s09 = strKEY;
    SHP_CHULHA_TYPE = strKEY ;

    UpdateData(false);
    Full_ScreenOK() ;
}

```

```

void CSHP_Q0001::OnP11()
{
// 출하정정구분 등록
    CMOBILE_KEYBOARD KeyBoardDlg ;
    UpdateData(true);

    strKEY      = SHP_CHULHA_MOD_OPT ;
    strKEYTITLE= L"정정구분" ;
    KeyBoardOK = 0 ;
    keyLength  = 1 ;
    KeyBoardDlg.DoModal() ;
    m_s11 = strKEY;
    SHP_CHULHA_MOD_OPT = strKEY ;

    UpdateData(false);
    Full_ScreenOK() ;
}

```

```

void CSHP_Q0001::OnP12()
{
// 출하정정구분 등록
    CMOBILE_KEYBOARD KeyBoardDlg ;
    UpdateData(true);

    strKEY      = SHP_MOD_OPT ;
    strKEYTITLE= L"정정사유" ;
    KeyBoardOK = 0 ;

```

```

        keyLength = 4 ;
        KeyBoardDlg.DoModal() ;
        m_s12 = strKEY;
        SHP_MOD_OPT = strKEY ;

        UpdateData(false);
        Full_ScreenOK() ;

    }

void CSHP_Q0001::OnCk11()
{
    c_chk1_1.SetCheck(1) ;
    c_chk1_2.SetCheck(0) ;
    SHP_CHUL_END_OPT = L"Y" ;
    UpdateData(false);
}

void CSHP_Q0001::OnCk12()
{
    c_chk1_1.SetCheck(0) ;
    c_chk1_2.SetCheck(1) ;
    SHP_CHUL_END_OPT = L"N" ;
}

void CSHP_Q0001::OnCk21()
{

```

```
        c_chk2_1.SetCheck(1) ;  
        c_chk2_2.SetCheck(0) ;  
        SHP_PDA_CONFIRM = L"Y" ;  
    UpdateData(false);  
}
```

```
void CSHP_Q0001::OnCk22()  
{  
    c_chk2_1.SetCheck(0) ;  
    c_chk2_2.SetCheck(1) ;  
    SHP_PDA_CONFIRM = L"N" ;  
    UpdateData(false);  
}
```

제 4 장의 부록: ARPC 모바일 유통관리 시스템

ARPC 모바일 유통관리 시스템

사용자 메뉴얼


1. 시스템 사용 환경

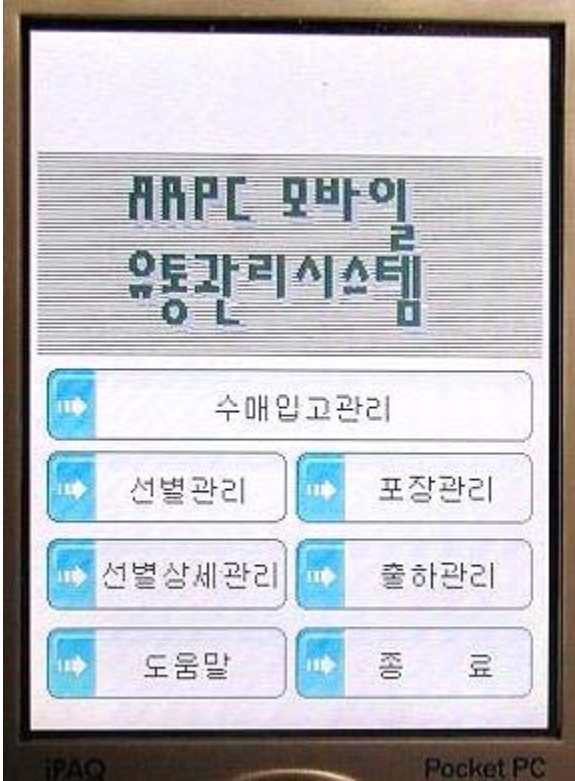
모바일산지유통시스템은 사용자의 일반 모바일 PDA와 Server가 인터넷환경으로 연결되어 운용된다.

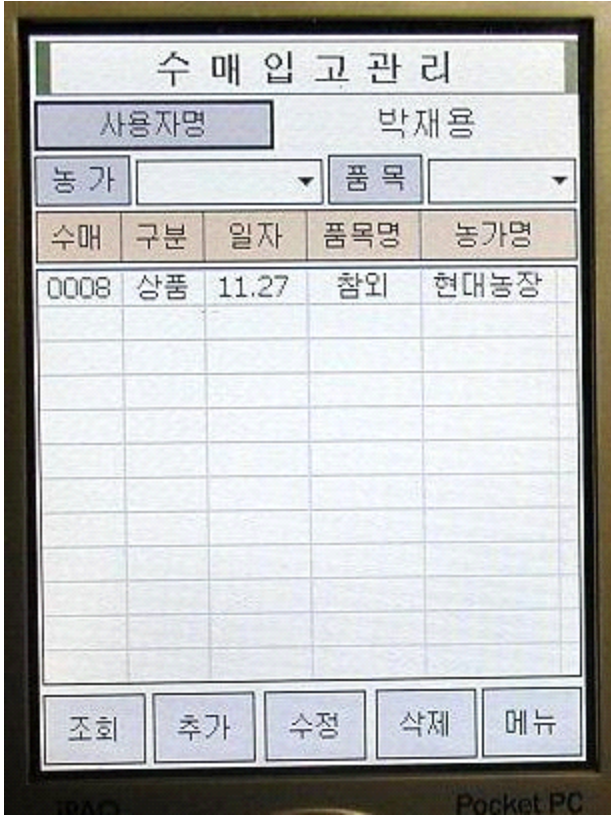
사용자 모바일 PDA는다음 사양을 기준으로 구축되었다.


- 가. 시스템 : WindowCE OS 기반의 PDA전기종
- 나. 메모리 : PDA 내장메모리 32Mb 권장
- 다. 운용환경 : 한글 Windows CE3.0 이상
- 라. 기타 : 무선 인터넷이 지원되는 업무 환경

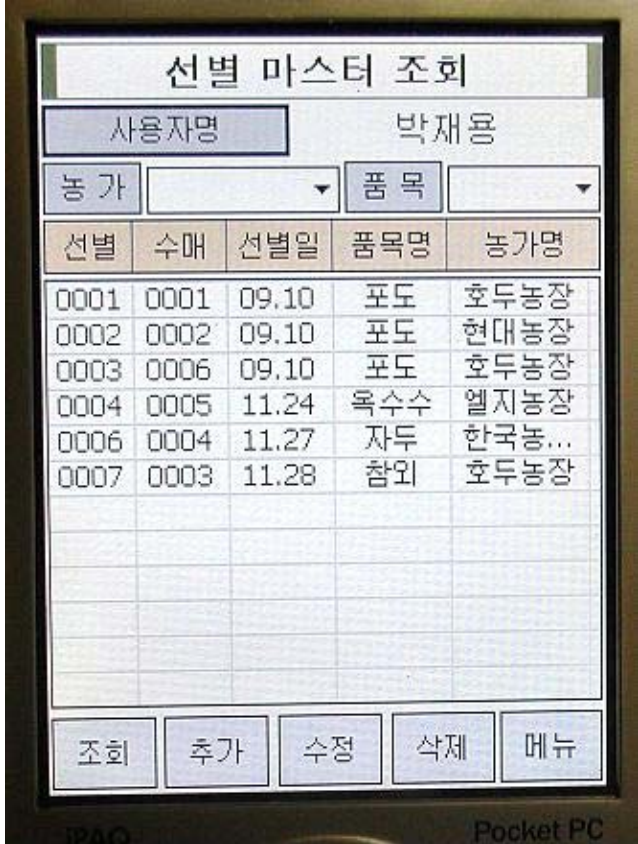
2. 프로그램 사용안내

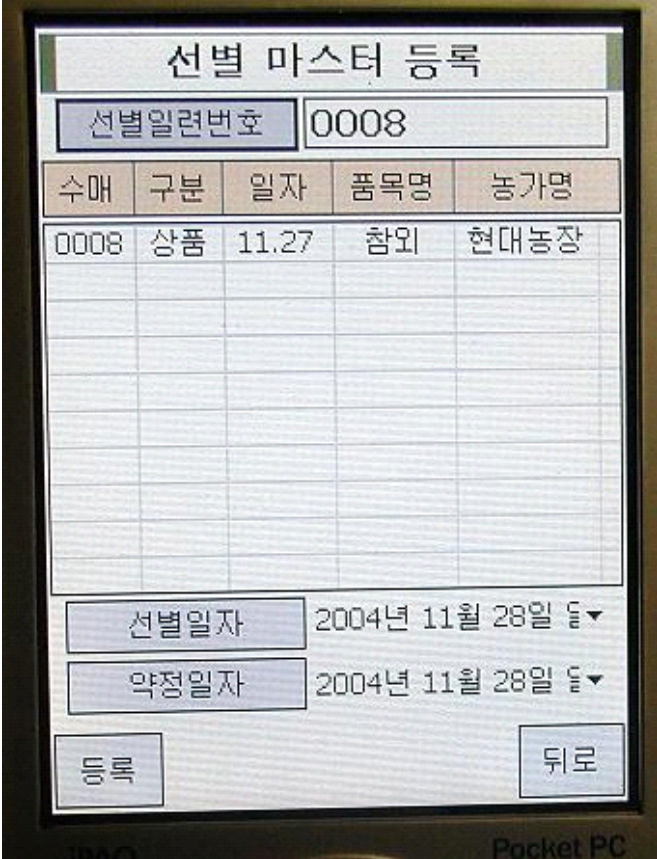
화면명	사용자 로그인 화면
화면	
설명	<ol style="list-style-type: none"> 로그인 화면으로 사용자명의 콤보를 클릭하여 사용자를 지정합니다. 비밀번호 버튼을 클릭하면 우측의 비밀번호 입력용 키패드 화면이 나옵니다. 키패드를 이용하여 비밀번호를 입력하고 확인 버튼을 누르면 비밀번호가 입력됩니다. 키패드는 일반 키보드 입력과 동일한 형식으로 본 키패드는 모바일시스템 전체 화면에서 사용되어지는 공통 키패드 화면입니다.

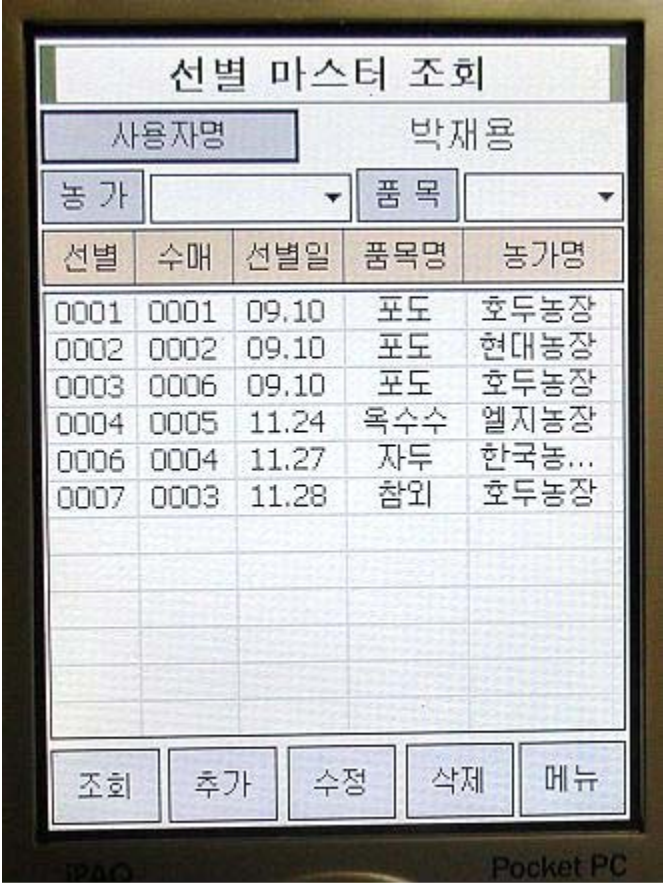
화면명	메인 메뉴 화면
화면	
설명	<p>1. 로그인을 하면, 본 화면으로 접근합니다.</p> <p>2. 원하는 화면에 접근하고자 한다면 해당 버튼을 클릭하시면 원하는 화면으로 접근합니다.</p>

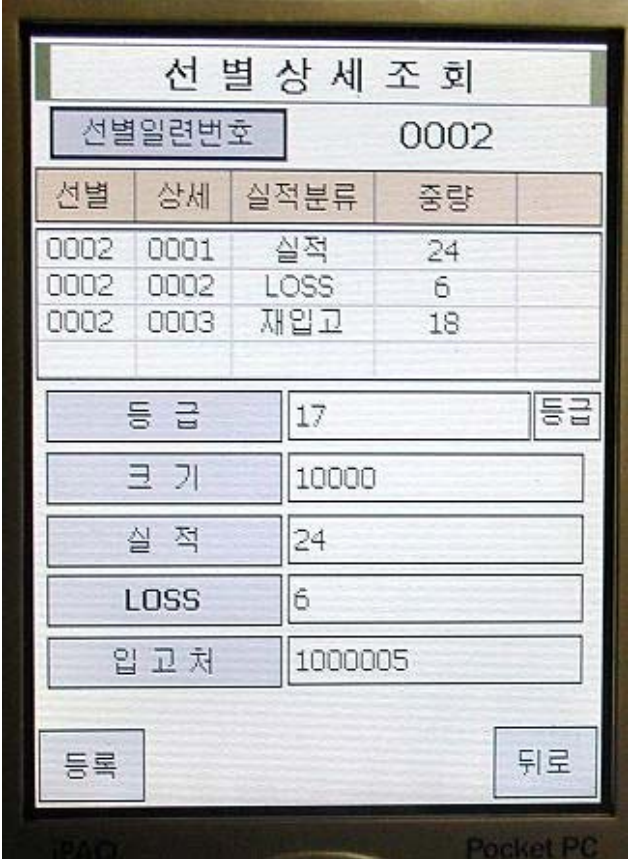
화면명	수매입고 조회
화면	
설명	<ol style="list-style-type: none"> 1. 상단의 조회 조건인 농가와 품목을 지정한 후 조회를 클릭하면 해당되는 수매입고 자료가 표시됩니다. 2. 표시된 자료를 기준으로 '추가', '수정' 버튼은 수매입고 창을 열어 입력, 또는 수정할 수 있게 됩니다. 3. '삭제' 버튼은 표시된 자료에 대해 삭제를 시행합니다. 4. '메뉴' 버튼을 클릭하면 상위의 메뉴 화면으로 이동합니다.


화면명	수매입고 등록/수정
화면	
설명	<ol style="list-style-type: none"> 1. 수매 입고의 추가와 수정을 위한 공통 화면으로 자료의 검색 및 입력에 사용됩니다. 2. 수매입고번호를 확인 한 후 원료/상품을 구분하는 자료 분리와 수매일자, 품목, 농가를 해당하는 코드로 지정하면 자동 입력됩니다. 3. 입고일자, 약정일자, 지역, 수량, 금액, 중량을 입력합니다. 4. '등록' 버튼을 누르면 자료는 자동 저장됩니다. 5. 뒤로 버튼은 조회화면으로 이동합니다.

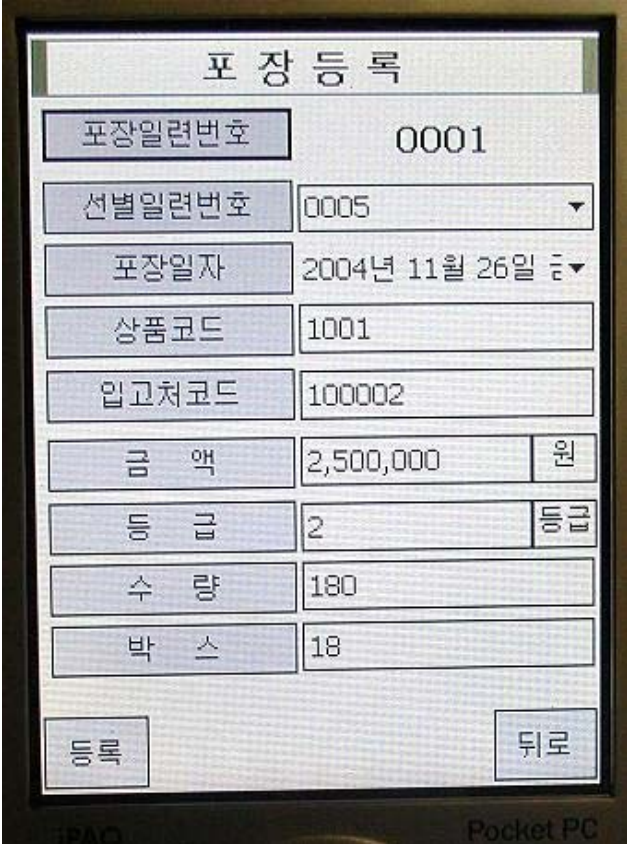
화면명	선별관리 조회																																																		
화면	 <table border="1" data-bbox="463 369 1016 1025"> <thead> <tr> <th>선별</th> <th>수매</th> <th>선별일</th> <th>품목명</th> <th>농가명</th> </tr> </thead> <tbody> <tr> <td>0001</td> <td>0001</td> <td>09.10</td> <td>포도</td> <td>호두농장</td> </tr> <tr> <td>0002</td> <td>0002</td> <td>09.10</td> <td>포도</td> <td>현대농장</td> </tr> <tr> <td>0003</td> <td>0006</td> <td>09.10</td> <td>포도</td> <td>호두농장</td> </tr> <tr> <td>0004</td> <td>0005</td> <td>11.24</td> <td>옥수수</td> <td>엘지농장</td> </tr> <tr> <td>0006</td> <td>0004</td> <td>11.27</td> <td>자두</td> <td>한국농...</td> </tr> <tr> <td>0007</td> <td>0003</td> <td>11.28</td> <td>참외</td> <td>호두농장</td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> </tbody> </table>	선별	수매	선별일	품목명	농가명	0001	0001	09.10	포도	호두농장	0002	0002	09.10	포도	현대농장	0003	0006	09.10	포도	호두농장	0004	0005	11.24	옥수수	엘지농장	0006	0004	11.27	자두	한국농...	0007	0003	11.28	참외	호두농장															
선별	수매	선별일	품목명	농가명																																															
0001	0001	09.10	포도	호두농장																																															
0002	0002	09.10	포도	현대농장																																															
0003	0006	09.10	포도	호두농장																																															
0004	0005	11.24	옥수수	엘지농장																																															
0006	0004	11.27	자두	한국농...																																															
0007	0003	11.28	참외	호두농장																																															
설 명	<ol style="list-style-type: none"> 1. 상단의 조회 조건인 농가와 품목을 지정한 후 조회를 클릭하면 해당되는 선별자료가 표시됩니다. 2. 표시된 자료를 기준으로 '추가','수정' 버튼은 수매입고 창을 열어 입력, 또는 수정을 할 수 있게 됩니다. 3. '삭제' 버튼은 표시된 자료에 대해 삭제를 시행 합니다. 4. '메뉴' 버튼을 클릭하면 상위의 메뉴 화면으로 이동합니다. 																																																		


화면명	선별관리 등록/수정
화면	
설 명	<ol style="list-style-type: none"> 1. 선별관리의 추가와 수정을 위한 공통 화면으로 자료의 검색 및 입력에 사용됩니다. 선별 마스터의 등록과 수정은 수매입고의 자료를 기준으로 작성되어 집니다. 2. 선별하려는 일자와 약정일자를 등록한 후 저장하면.. 자동 저장됩니다. 3. 선별마스터는 이러한 마스터만 관리하고 선별 상세관리에서 자료를 선별상세자료로 등록하고 수정합니다. 4. 뒤로 버튼은 조회화면으로 이동 합니다.

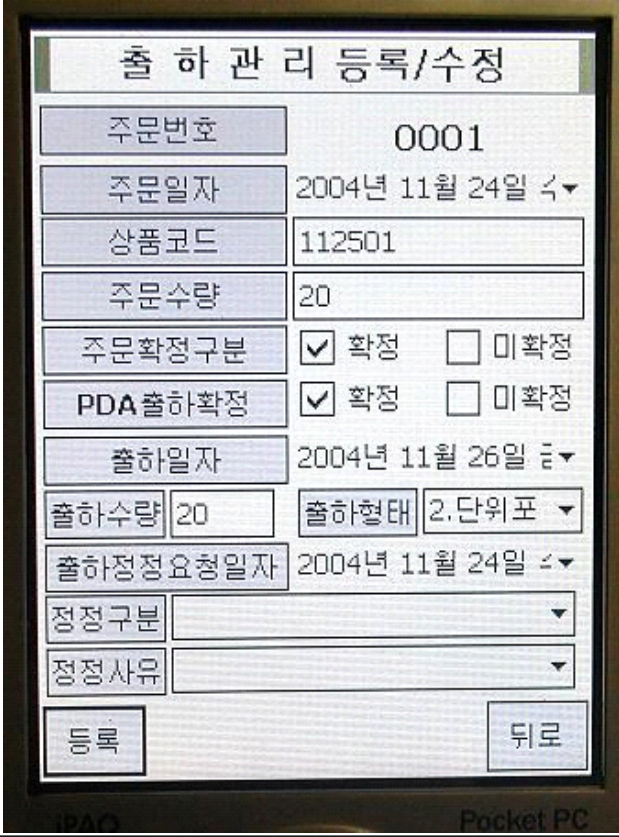
화면명	선별상세관리 조회																																																							
화면	 <p style="text-align: center;">선별 마스터 조회</p> <p>사용자명: 박재용</p> <p>농가: [] 품목: []</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>선별</th> <th>수매</th> <th>선별일</th> <th>품목명</th> <th>농가명</th> </tr> </thead> <tbody> <tr> <td>0001</td> <td>0001</td> <td>09.10</td> <td>포도</td> <td>호두농장</td> </tr> <tr> <td>0002</td> <td>0002</td> <td>09.10</td> <td>포도</td> <td>현대농장</td> </tr> <tr> <td>0003</td> <td>0006</td> <td>09.10</td> <td>포도</td> <td>호두농장</td> </tr> <tr> <td>0004</td> <td>0005</td> <td>11.24</td> <td>옥수수</td> <td>엘지농장</td> </tr> <tr> <td>0006</td> <td>0004</td> <td>11.27</td> <td>자두</td> <td>한국농...</td> </tr> <tr> <td>0007</td> <td>0003</td> <td>11.28</td> <td>참외</td> <td>호두농장</td> </tr> <tr> <td> </td> <td> </td> <td> </td> <td> </td> <td> </td> </tr> <tr> <td> </td> <td> </td> <td> </td> <td> </td> <td> </td> </tr> <tr> <td> </td> <td> </td> <td> </td> <td> </td> <td> </td> </tr> <tr> <td> </td> <td> </td> <td> </td> <td> </td> <td> </td> </tr> </tbody> </table> <p style="text-align: center;"> <input type="button" value="조회"/> <input type="button" value="추가"/> <input type="button" value="수정"/> <input type="button" value="삭제"/> <input type="button" value="메뉴"/> </p>	선별	수매	선별일	품목명	농가명	0001	0001	09.10	포도	호두농장	0002	0002	09.10	포도	현대농장	0003	0006	09.10	포도	호두농장	0004	0005	11.24	옥수수	엘지농장	0006	0004	11.27	자두	한국농...	0007	0003	11.28	참외	호두농장																				
선별	수매	선별일	품목명	농가명																																																				
0001	0001	09.10	포도	호두농장																																																				
0002	0002	09.10	포도	현대농장																																																				
0003	0006	09.10	포도	호두농장																																																				
0004	0005	11.24	옥수수	엘지농장																																																				
0006	0004	11.27	자두	한국농...																																																				
0007	0003	11.28	참외	호두농장																																																				
설 명	<ol style="list-style-type: none"> 1. 상단의 조회 조건인 농가와 품목을 지정한 후 조회를 클릭하면 해당되는 선별자료가 표시됩니다. 2. 표시된 자료를 기준으로 '추가','수정' 버튼은 수매입고 창을 열어 입력, 또는 수정을 할 수 있게 됩니다. 3. '삭제' 버튼은 표시된 자료에 대해 삭제를 시행 합니다. 4. '메뉴' 버튼을 클릭하면 상위의 메뉴 화면으로 이동합니다. 																																																							

화면명	선별상세관리 등록/수정
화면	
설명	<p>1. 선별상세관리의 추가와 수정을 위한 공통 화면으로 자료의 검색 및 입력에 사용됩니다.</p> <p>2. 선별 상세에서는 등급,크기,실적,LOSS 를 입력하면 자동으로 재입고의 자료를 산정하고 등록됩니다. 본 모바일에서는 선별에 대한 정보와 선별 후 잔여 량에 대한 자료만을 전송하고 난해한 선별의 많은 공정을 저장하지는 않습니다.</p> <p>3. 뒤로 버튼은 조회화면으로 이동합니다.</p>

화면명	포장관리 조회
화면	
설명	<ol style="list-style-type: none"> 1. 폼이 로딩될 때 포장자료가 표시됩니다. 2. 표시된 자료를 기준으로 '추가', '수정' 버튼은 수매입고 창을 열어 입력, 또는 수정을 할 수 있게 됩니다. 3. '삭제' 버튼은 표시된 자료에 대해 삭제를 시행 합니다. 4. '메뉴' 버튼을 클릭하면 상위의 메뉴 화면으로 이동합니다.

화면명	포장관리 등록/수정
화면	
설명	<p>1. 포장관리의 추가와 수정을 위한 공통 화면으로 자료의 검색 및 입력에 사용됩니다. 포장정보의 레코드를 클릭하고 등록/수정을 하면 상기의 화면이 나오는데, 이에 대하여 선별일련번호를 지정하고 일자, 상품코드, 입고처 및 하단의 자료 입력을 시행합니다.</p> <p>* 현재는 상품 코드, 입고처 코드는 코드 형태로만 지원합니다. 이는 코드의 부여가 시스템마다 다르기 때문에 현재의 내용으로 전개하고, 향후 바코드의 스캐너를 이용한 입력 방식으로 전환하면 본 내용의 화면과는 다르게 변경될 수 있다.</p> <p>2. 뒤로 버튼은 조회화면으로 이동합니다.</p>

화면명	출하관리 조회
화면	
설명	<ol style="list-style-type: none"> 1. 폼이 로딩될 때 출하정보가 표시됩니다. 출하정보는 출하를 시행하는 정보를 우선적으로 등록하며 출하에 대한 주문정보를 기준으로 합니다. 출하정보를 입력 할 경우에는 본 서버로 할당받은 주문정보를 기준으로 처리하는 방식으로 모바일 독립적으로 주문출하를 시행할 수 없습니다. 자료는 상기와 같은 조회 구조를 가지며 출하 승인에 대한 PDA 확정 루틴이 있어서 이를 확인해 주어야 출하가 승인됩니다. 2. 표시된 자료를 기준으로 '추가','수정' 버튼은 수메입고 창을 열어 입력, 또는 수정을 할 수 있게 됩니다. 3. '삭제' 버튼은 표시된 자료에 대해 삭제를 시행 합니다. 4. '메뉴' 버튼을 클릭하면 상위의 메뉴 화면으로 이동합니다.

화면명	출하관리 등록/수정
화면	
설 명	<p>1. 레코드를 클릭하거나 추가를 등록하면 주문번호에 대한 자료를 기준으로 출하정보를 입력합니다. 주문 확정이 된 내용을 확인하거나 확인시키면서 PDA에서 내린 출하정보라는 확정시행 합니다.</p> <p>만일 출하등록이 된 상태에서 출하정정 요청이 있을 경우는 출하 정정자료를 입력하고 등록합니다.</p> <p>이는 출하가 중지 되었거나 출하의 형식이 변경 되었을 경우 자료를 재입력하는 경우입니다.</p> <p>2. 뒤로 버튼은 조회화면으로 이동 합니다.</p>

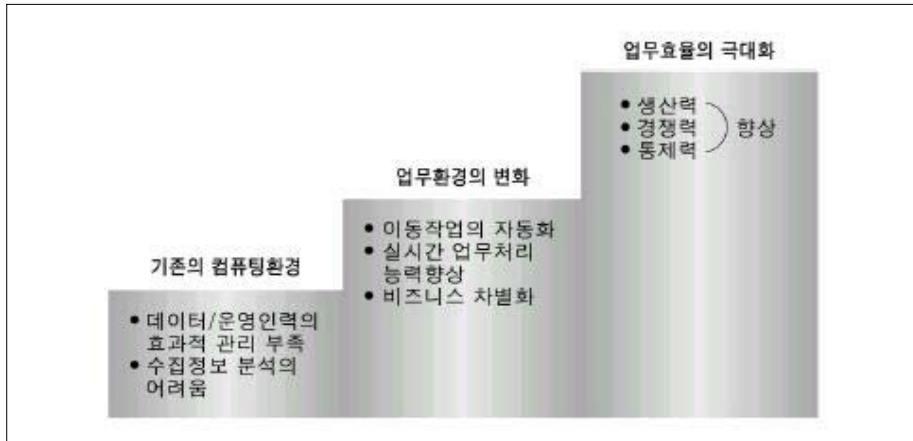
제 5 장 기대효과 및 평가

제 1 절 기대효과

본 연구를 통해 산지유통전문조직의 업무능률향상에 의한 생산성 및 효율성 증대가 기대되며, 특히 다음과 같은 사항들의 효과가 예상된다.

- ① 기존의 고정된 전산망에서는 유동적인 업무에 대응하는 능력이 결여되어 있었다면 모바일 컴퓨팅 환경의 구현으로 이에 대한 능력이 급격히 향상
- ② 업무처리단계 간소화 및 일관성 유지
- ③ 품질기능 강화 및 업무처리능률 향상
- ④ 부가 업무활동 전개 및 사내 업무 효율 향상
- ⑤ 작업영역의 확대, 통제능력의 증가
- ⑥ 산지유통 전문조직의 비즈니스 프로세스 개선, 시간, 인력, 정보 등 자원의 효율적 관리를 통해 생산성 향상
- ⑦ 업무현장의 불특정지역에서 빈번하게 발생하는 데이터 및 운영인력에 대한 효과적인 관리 및 수집된 정보의 신속 정확한 분석과 활용 가능
- ⑧ 실시간 데이터 처리에 따른 신속한 의사결정 지원 및 경영관리 실현
- ⑨ Data 재처리를 위한 인력 절감(별도의 전표 정리요원 불필요)
- ⑩ 효율적인 물류 및 재고관리에 의한 유통비용 절감
- ⑪ 경영 및 관리효율 증대로 고정비용 절감
- ⑫ 종이기반 작업구조 개선 및 관리업무 부담 감소
- ⑬ 저렴한 장비가격과 시스템 교체 및 업그레이드 용이
- ⑭ 저렴한 시스템 관리비용과 업무증가에 따른 추가 개발비 절감
- ⑮ 비용대비 높은 정보 활용 및 실질적인 수익 연계

<표 5-1> 모바일컴퓨팅 도입의 효과



제 2 절 활용방안

본 연구를 통해 향후 산지유통전문조직의 APC 종합관리가 용이해 질뿐만 아니라 데이터를 기반으로 경영이 가능해 진다. 특히 농축산물분야의 산지유통시설을 도입한 주체는 물론 가공업체, 식품업체, 유통업체 등에 확산 가능하다. 다만 본 산출물인 정보시스템을 도입하기 위해서 각 조직들의 특성과 시장 환경을 반영해야 하기 때문에 그대로 사용하기 보다는 커스터마이징 (Customizing)이 필요하며, 아직 정보화에 대한 인식이 산지에서 폭넓게 자리 잡지 못하고 있기 때문에 이에 대한 홍보 및 교육 등이 절실하게 요구되기도 하다.

또한 지금까지의 시설위주의 하드웨어적인 정부지원에서 벗어나 경쟁력을 갖춘 산지유통전문조직들이 들어서기 위해서는 본 연구의 정보시스템 등 소프트웨어적인 지원의 적극적인 고려가 필요하다.

제 3 절 효과 분석

제3장, 4절에서 언급한 정보시스템 평가방법에 따라 정량화가 가능한 몇 가지 지표를 토대로 분석한 결과는 다음과 같다

1. 구매, 주문, 재고 등의 업무처리 소요시간 단축으로 연간 업무 부담률(%) 감소

<표 5-2> 업무감소율 상세내역 분석

지표명(Measure)		상지무통관권 업무부담률		척도(Scale)	%연간업무부담률		
지표정의(Definition)		구매, 주문, 재고 등의 업무처리 소요시간 단축으로 연간 업무 부담률(%) 감소					
정보원 효과	지표값 변화	경보화 이전	경보화 이후	차이	확신도		
		80%	40%	40%	매		
	경보화기여도(%)	최대: 90 최선: 80 최소: 60	신술근거	경보화에 의한 업무처리 부담률 감소 기여도			
	경보화 영향	최대: 0.36 최선: 0.32 최소: 0.24	신술근거	경보화 전후 차이 × 경보화 기여도			
사업적 가치	금액	27,126천원	적용기법	회계기법			
	신술근거	<ul style="list-style-type: none"> 가치 산출 <ul style="list-style-type: none"> - 최대: 0.36 × 9명 × 1,507천원 = 27,126 천원 - 최선: 0.32 × 9명 × 1,507천원 = 24,112 천원 - 최소: 0.24 × 9명 × 1,507천원 = 18,084 천원 해당업무 담당자 수: 9명 연간 국민평균임금: 1,507천원(2008년 기준(1인당 국민소득(명목), 통계청 기준)) ※ 산출된 금액만큼 부가해치는 업무 절감 및 집중 					
가치 실현도	연도별 실현도 (%)	2005	2006	2007	2008	2009	2010
		80	80	100	100	100	100
	연도별 가치 (천원)	21,701천원	27,126천원	27,126천원	27,126천원	27,126천원	27,126천원

<표 5-3> 업무감소율 사업적 가치분석

지표명(Measure)	신자유를관리 업무부담률			척도(Scale)	% (연간 업무부담률)	
지표정의(Definition)	수발, 주문, 재고 등의 업무처리 소요시간 단축으로 연간 업무 부담률(%) 감소					
지표명 변경	현재 지표값	90 %				
	목표값	80 %	목표달성가능연도		2005	
정보화효과	<ul style="list-style-type: none"> 신자유를관리조직의 주요 업무들로 농가로부터 수발, 거래처부터 주문관리, 재고관리 등을 수행하고 있음 지금까지의 업무처리는 담당처가 직접 수거를 기록하거나 전화, 팩스 등을 대응하여 처리함으로써 업무처리에 필요한 시간 및 비용이 과도하게 소요되고 있음 온라인을 통한 신자유를관리 업무 처리의 경우 총 11명이 출사하고 있으며, 이 중 수발, 주문, 재고 등의 업무를 수행하는 인원은 5명임 현행 업무처리방법 <ul style="list-style-type: none"> - 전화, 팩스 - 수거 기록 정보화 후 처리 수집 방법 <ul style="list-style-type: none"> - 모바일을 통해 정보 교환 및 기록 목표값은 현재 담당자들의 해당 업무부담률 80%로 보고 이 중 모바일을 통해 수발, 주문처리 및 재고관리를 할 경우 60% 정도로 업무부담률의 낮아지는 것으로 설정 					
가치 실현도(%)	2004	2005	2007	2008	2009	2010
	90	100	100	100	100	100
* 가치실현도는 실제 사용해야 한 직원에 잘 적용되는 시점을 기준으로 2005년 설정						

2. 기초데이터 및 실시간 정보로 의사결정에 소요되는 시간 단축

<표 5-4> 효과적인 경영의사결정시간단축 상세내역

지표명(Measure)	효과적인 경영의사결정시간			척도(Scale)	해시간	
지표정의(Definition)	거초데이터 및 실시간 정보로 의사결정에 소요되는 시간 단축					
지표명 변경	현재 지표값	3시간				
	목표값	1시간 이내	목표달성가능연도		2006	
정보화효과	<ul style="list-style-type: none"> 신자유를관리조직의 담당상무, 단장, 부단장을 비롯한 경영자는 주요경영에 필요한 의사결정을 수행하고 있음 현재의 경영의사결정은 대부분 직원이 팩스지도를 최대한 작성한 후 그에 대해 필요한 관리지도를 작성하여 이를 기초로 하여 의사결정을 내리고 있음 의사결정에 필요한 자료를 생성하고 분석하는 담당자는 대부분 과할급 이상이 수행하고 있으며, 총 6명이 업무를 담당 하고 있음 현행 경영의사결정에 필요한 자료 생성 및 분석 <ul style="list-style-type: none"> - 팩스지도를 기록 및 분석 정보화 후 자료 수집 방법 <ul style="list-style-type: none"> - 데이터베이스 및 문자시스템 목표값은 현재 경영의사결정에 필요한 거초(90)보다 기록 및 분석 소요시간을 3시간에서 정보시스템을 도입한 경우 1시간내로 단축 가능 					
가치 실현도(%)	2005	2006	2007	2008	2009	2010
	90	100	100	100	100	100
* 가치실현도는 실제 사용해야 한 직원에 잘 적용되는 시점을 기준으로 2005년 설정						

<표 5-5> 효과적인 경영의사결정시간단축 사업적 가치분석

지표명(Measure)		효과적인 경영의사결정시간			척도(Scale)	기준(B)	
지표정의(Definition)		기후대타당 및 실시간 정보에 의사결정비 소요되는 시간 단축					
경영활동 효과	지표값 변화	정보화 이전	정보화 이후	차이	확신도		
		3시간	1시간	2시간	높		
	감소할기여도(%)	최대: 90 최선: 80 최소: 60	산출근거	정보화(비) 의한 업무처리 부담을 감소 기여도			
	정보화 영향	최대: 1.8 최선: 1.6 최소: 1.2	산출근거	정보화 전후 처리 × 정보화 기여도			
사업적 가치	금액	20,344 천원	정량기법	회계기법			
	산출근거	<ul style="list-style-type: none"> • 가치 산출 <ul style="list-style-type: none"> - 최대: 1.8 × 6명 × 6,279원 × 300일 = 20,344 천원 - 최선: 1.6 × 6명 × 6,279원 × 300일 = 18,084 천원 - 최소: 1.2 × 6명 × 6,279원 × 300일 = 13,563 천원 • 근무일수: 6명 • 연간 국민 평균 소득: 1,807만원(2008년 기준 1인당 국민소득(달러) 통계청 기준) • 시간당 평균임금: 6,279원(시간당 평균임금 = 1인당 국민소득/12월/25일/8시간) 					
가치 실현도	면도별 실현도 (%)	2005	2006	2007	2008	2009	2010
		10	90	100	100	93	100
	면도별 가치 (천원)	16,275천원	20,344 천원	20,344 천원	20,344 천원	20,344 천원	20,344 천원

도출 가능한 지표를 토대로 분석한 결과, 순천농협 농산물산지유통센터의 경우 첫째 도입되는 2005년의 경우 37,976천원의 효과가 발생하였으며, 시스템 도입비용이 1억원으로 가정할 경우 시스템 도입 후 3년 정도 사용할 경우 정량적 지표에 의한 비용대비 사업성고가 발생하는 것으로 나타났다.

<표 5-6> 정량적 효과지표에 의한 투자효과(천원)

구 분	2005년	2006년	2007년	2008년	2009년	2010년
업무부담률 감소	21,701	27,126	27,126	27,126	27,126	27,126
의사결정 시간단축	16,275	20,344	20,344	20,344	20,344	20,344
합 계	37,976	47,470	47,470	47,470	47,470	47,470

제4절 사용자 만족도 조사 결과

1. 만족도 측정에 사용된 항목

사용자 만족은 Bailey & Pearson(1983)의 측정도구를 근간으로 하는 Ives et al.(1983)의 도구와 Barou야 & Orlikowski(1988)의 13개 항목으로 측정되었다.

Bailey & Pearson은 사용자의 정보시스템 만족에 영향을 주는 39개의 요인을 확인하고 그것을 특정하는 도구를 제안하였는데 기술적·인간적 측면 모두 포함하고 있다.

- 시스템의 질 : 응답시간/ 접근편이성/ 사용자 요구조건 형상화/ 오류정정/ 데이터와 모델의 보안/ 시스템의 문서화/ 시스템의 유연성/ 시스템의 통합
- 정보의 질 : 산출물의 정확성/ 신속성/ 적합성/ 신뢰성/ 형태/ 현실성/ 완벽성
- 서비스 질 : 정보부서의 기술적 지원/ 정보부서의 태도/ 정보부서 산출물과 서비스의 계획/ 개발요구시간/ 시스템 개선을 위한 요구조건의 수용/ 공급자의 유지보수지원/ 시스템에 대한 사용자의 이해/ 사용자를 위한 훈련
- 갈등해소 : 정보부서와 비 정보부서간의 경쟁/ 정보부서 자원의 할당우선순위/ 사용자와 정보부서와의 관계/ 정보부서의 통제/ 조직 내 정보부서의 위치
- 정보사용 : 결과물의 양
- 사용자 만족 : 최고사용자의 참여/ 서비스의 질에 대한 보상 방법/ 사용자의 신뢰/ 사용자 참여
- 개인효과 : 사용자의 기대/ 업무효과/ 인지된 유용성

본 연구에서 이용된 사용자 만족도 조사 측정 항목은 <표 5-7>와 같다.

<표 5-7> 사용자 만족도 조사 항목

번호	측정항목(변수설정)
1	최고경영자의 참여도
2	구축요원의 기술적 능력
3	구축요원의 태도
4	변경처리 요구
5	처리 속도
6	보고서의 정확성
7	정보시스템 이해의 용이성
8	시스템에 대한 신뢰
9	직무에 대한 효과
10	오류수정
11	문서화
12	시스템 개발에 소요되는 시간
13	보고서의 신뢰성
14	보고서의 적시성
15	보고서의 완전성
16	사용자 요구와 보고서의 목적적합성
17	정밀성
18	컴퓨터 교육훈련의 정도
19	시대성
20	새로운 상황에 대한 시스템의 유연성
21	정보시스템 사용의 편이성
22	정보시스템 유용성의 인식
23	정보시스템 기능에 대한 사용자의 참여정도

2. 사용자 만족도 조사 결과

사용자 만족도는 순천농협 담당자 10명을 대상으로 한 설문조사 결과에

의해 측정되었다. 측정항목에 따른 만족도 분석 결과는 다음과 같다.

- 최고경영자의 참여도 부분: 매우 강함(6), 강함(4)로서 최고경영층의 시스템 개발에 따른 관심과 추진의지가 강한 것으로 분석되었다.
- 구축요원의 기술적 능력 부분: 좋다(8), 그저 그렇다(1), 나쁨(1)로서 종합시스템 개발자들의 기술적 능력 및 지식수준에 만족한 것으로 분석되었다.
- 구축요원의 태도 부분 : 친절(12), 그저 그렇다(3), 불친절(2)로서 참여 개발자들의 친절도는 높은 것으로 분석되었다.
- 변경처리 요구 부분 : 신속하다(5), 그저 그렇다(4), 느리다(1)로서 사용자들의 변경요구사항에 대하여 대체로 신속하게 처리된 것으로 분석되었다.
- 처리속도 부분 : 빠르다(3), 그리 그렇다(5), 느리다(2)로서 컴퓨터의 처리속도가 보통인 것으로 분석되어, 차후 사용자들의 컴퓨터를 업그레이드할 필요가 있다.
- 보고서의 정확성 부분 : 매우정확(2), 정확(4), 그저 그렇다(4)로서 종합관리시스템에서 제공되는 보고서의 정확성이 높은 것으로 분석되었다.
- 정보시스템 이해의 용이성 부분: 쉽게 이해한다(2), 그저 그렇다(5), 어렵다(3)로서 사용자들이 종합관리시스템의 이해도가 조금 낮은 것으로 분석되어 차후 지속적인 사용자 교육 등을 통한 사용자의 이해를 높여야 할것이다.
- 시스템에 대한 신뢰 부분 : 매우 높다(1), 높다(2), 그저그렇다(3), 낮다(4)로서 시스템에 대한 신뢰도가 낮은 것으로 나타나 시스템 신뢰성을 높이는 방안을 모색해야 할 것이다.
- 직무에 대한 효과 부분 : 매우 높다(2), 높다(3), 그저 그렇다(5)로서 모바일 시스템의 업무 적용성이 아직 낮은 것으로 평가되었다.
- 오류수정 부분 : 매우 신속하다(2), 신속하다(3), 그저 그렇다(3), 느리다(2)로서 컴퓨터 출력오류 수정 방법이 대체로 만족스럽지 못한 것으로

분석되었다.

- 문서화 부분 : 매우충분(3), 충분(5), 그저 그렇다(2)로서 사용자들의 위
한 설명서 및 서류의 내용이 충분 것으로 분석되었다.
- 시스템 개발에 소요되는 시간 : 짧다(8), 적당한다(2)로 시스템의 개발
소요 시간이 대체로 짧다고 평가하여 시스템 개발시 충분한 시간의 확
보가 필요하다.
- 보고서의 신뢰성 부분 : 매우 높다(3), 높다(7)로 모바일시스템에서 제
공되는 보고서의 신뢰성이 상당히 높은 것으로 분석되었다.
- 보고서의 적시성 부분 : 매우 잘된다(3), 높다(7)로서 모바일시스템에서
제공되는 보고서가 적시에 활용되는 것으로 분석되었다.
- 보고서의 완전성 : 매우 완전(1), 완전(2), 그저 그렇다(2), 불완전(5)로
서 제공된 보고서의 완전성이 낮아 이에 대한 보완이 필요하다.
- 사용자 요구와 보고서의 목적적합성 부분 : 매우적합(2), 적합(7), 그저
그렇다(1)로 사용자의 요구와 시스템에서 제공되는 보고서 등이 용구에
상당히 적합한 것으로 분석되었다.
- 시스템의 정밀성 : 매우 높다(3), 높다(7)로서 시스템의 정밀성이 상당
히 높은 것으로 분석되었다.
- 컴퓨터 교육훈련의 정도 : 충분(3), 그저 그렇다(4), 불충분(3)로서 사
용자의 정보교육이 적당한 것으로 나타났으나 좀 더 많은 사용자 교육
을 통하여 사용자의 시스템 활용에 도움을 주어야할 것으로 판단된다.
- 새로운 상황에 대한 시스템의 유연성 : 충분(7), 그저 그렇다(3)로서 여
건변화에 대한 정보시스템의 적응능력이 상당히 충분한 것으로 분석된
다.
- 사용자 편의성 부분 : 편리(3), 그저 그렇다(4), 불편(3)로서 사용자 편
이성이 다소 낮은 것으로 나타났으나 이는 수작업에 익숙한 사용자들
이 시스템 도입에 따른 업무훈련 등에서 오는 것으로 판단되며 향후
사용자 교육과 사용시간이 길어짐으로 해결될 것으로 판단된다.

- 시스템의 유용성 부분 : 매우 높다(2), 높다(5), 그저 그렇다(3)로 모바일시스템의 유용성이 상당히 높은 것으로 분석된다.
- 시스템 기능에 대한 사용자의 참여정도 : 적극적(7), 그저 그렇다(3)로 시스템 개발이나 교육에 대한 참여도가 높은 것으로 평가되었다.

제 6 장 결 론

모바일 산지유통시스템이라는 차세대 운용시스템은 체계적 연구와 표준이 국제적으로 정해지지 않은 상황에서 구축되고 현장에 적용되고 있는 실정이다. 아울러 농가 소프트웨어 사용이 기피되고 있는 상황에서 모바일이라는 새로운 운용방식은 사용상의 어려움을 극복하지 못 하고 있었고 업무적용에 생소함을 드러냈다. 그러나 모바일 산지유통시스템의 현장 업무자의 요구조사를 바탕으로 철저한 업무분석과 재설계를 수행하였고 이를 바탕으로 업무적용성과 활용성이 높은 범용시스템을 개발하였다.

기능적인 측면에서 보면 기존의 모바일 프로그램들이 현재 복잡하고 다양한 정보표현하려는 모습에서 주어진 업무의 가장 표준적인 내용을 적용하고 이에 따라 입력화면을 구성, 사용자의 자료 입력 부담을 최소화 하도록 노력하였다. 모바일시스템이라는 기기적인 특성인 작은 화면 구조에서 사용자 중심의 입력사용자인터페이스와 기능구현에 충실하였다.

또한 내용적인 측면에서는 시스템의 복잡구조를 배제하고 단위업무로 재구성하고 모바일시스템의 적용이 현장 업무를 방해 하지 않는 범위에서 현장 업무를 연결하도록 사용자 편의성을 제공하고 있다. 또한, 주요 기능 중 가장 업무에 필요로 하는 부분인 수매입고, 선별, 포장, 출하를 중심으로 기능을 구현하고 있다.

모바일산지유통시스템은 업무의 전산화, 정보화를 통해 운영효율성을 높이고 업무운용 효율화를 도모할 것으로 기대되고 있으나 개발과정 중 여러 가지 시행착오를 겪어 추후 개발 시에 참고해야 할 것이다.

먼저 개발방식에 있어서 현행 업무모델이 없었던 관계로 시스템 개발 시 갖은 설계변경과 소규모 화면에서 보여 질 수 있는 산지유통 업무 항목을 결정하는 과정이 가장 힘든 부분이었다. 이는 향후 업무적용에 있어서 각 단위 업무를 모듈화하여 모바일 뿐 만 아니라 ERP시스템에서 어떠한 이기종간의 정보교환에 따른 패키지 모듈 형식으로 개발할 방법을 모색하는 것이 바람직하다.

또한, 산지유통센터의 단위업무 프로세스가 명확하게 정의되어야 할 것이다. 현재 대부분의 산지유통센터는 업무가 정형화되지 않아 시스템 개발 시 어려움이 가중되고 개발 이후에도 빈번한 시스템 재설계 요구가 발생하고 있는 실정이다.

이러한 점에서 모바일 산지유통은 업무 결합이 다양하게 일어나는 부분을 모두 소화하기에는 시스템적인 한계점을 너무 많이 가지고 있다.

하지만 업무 중 신속하게 처리 하거나 자연스런 정보를 파악하고 대응하기 위한 조건으로는 모바일 시스템의 도입은 적극 권장할 만한 일이다.

다양한 물류, 행정, 의료 분야에서 모바일의 사용이 확대 되고 업무비중이 높아져 가고 있는 실정에서 산지유통의 모바일 적용은 한층 더 산지유통의 업무 자동화를 앞당길 수 있다고 본다.

가장 중요한 것은 산지유통센터의 경영진과 업무담당자 모두가 시스템 도입의 목표와 필요성을 공감하고 체계적으로 추진할 수 있는 내부환경을 만드는 것이다. 시스템 개발업무에 대한 비전을 최고경영자뿐 아니라 말단 직원까지 공유할 수 있도록 원활한 내부 토론과 교육, 훈련이 절실히 필요하다.

참 고 문 헌

- 고재관, 「Starting PDA Programming」, 삼각형프레스, 2001.
- 고재관, 「팔아라 실전 PDA 프로그래밍」, 삼각형프레스, 2002.
- 김동환·김병률·김재식, 「농산물유통론」, 농민신문사, 2003.
- 김성환·여운영 「미래를 지향하는 모바일 네트워크」, 야스미디어, 2003.
- 김윤호·황홍선·박준호, 「모바일 콘텐츠 비즈니스로 가는 성공 로드맵」, 비비컴, 2003.
- 농수산물유통공사, 「농산물 산지유통센터 운영사례집」, 2001.
- 농수산물유통공사 유통정보부, “대형유통업체와 산지출하조직의 농산물거래실태 조사 분석,” 2004.
- 농식품신유통연구원, 「산지유통전문조직 유통활성화사업의 성과와 개선방안」, 2003.
- 농식품신유통연구원, 「산지유통전문조직의 수확후관리 평가와 발전과제」, 2002.
- 농협조사연구소, 「유비쿼터스 혁명과 전망」, 2004.
- 민병석, “고객만족정보시스템의 모형개발에 관한 연구,” 경희대 경영대학원, 2000.
- 박찬욱, 「모바일 마케팅:움직이는 고객 움직이는 마케팅」, 시그마인사이트컴, 2002.
- 배갑선, 「정보시스템 사용자의 만족도에 관한 실증적 연구」, 1998.
- 여인춘, 「임베디드 VC++」, 정보문화사, 2002.
- 오세조·박진용, 「시장지향적 유통관리」, 박영사, 2001.
- 윤종원, “모바일 무선인터넷서비스의 이용행태와 추구혜택에 관한 연구,” 단국대 대학원, 2001.
- 이정환외, 「농산물유통종합정보시스템 개발연구」, 한국농촌경제연구원, 2001.

이종우·윤종완 , 「Windows CE MFC Programming」 , 사이텍미디어 , 2002.

정훈 , 「Window CE API Bible」 , 글로벌 , 2001.

조재만·곽선정, 「임베디드 Window CE 프로그래밍」 , 2002.

최양부외, 「농식품 신유통과 산지유통센터의 기능 정립」 , 제 10차 신유통토론회 자료집, 2000.

최양부·김동환, 「농식품 신유통시스템의 패러다임과 발전과제」, 농업경제연구40(2) pp. 189-218, 2000.

최양부외, 「신유통시스템 구축을 위한 산지유통센터 발전방향」 , 농식품신유통연구회, 2000.

최양부외, 「유통환경 변화에 대응한 농식품 신유통정책의 방향과 과제」 , 농식품신유통연구회, 2001.

최양부외, 「산지 유통전문조직 평가시스템 개발」 , 농식품신유통연구회, 2002.

최정용·안동규, 「무선인터넷개론」 , 사이텍미디어 , 2004.

최준혁, 「PDA 때려잡기」 , 심마니, 2001.

최재혁, 「전자상거래를 위한 모바일 에이전트 기반의 상인 중개 시스템」 , 포항공과대 대학원, 2000.

한국전산원, 「농산물 전자상거래 활성화 방안 연구」 , 2002.

한맥/하이텔 PDA동호회, 「PDA활용 및 무선인터넷 길라잡이」 , 정보문화사, 2002.

한빛IT, 「PDA 제대로 활용하기」 , 사이버출판사, 2002.

Douglas Boling , *Programming MicroSoft WINDOWS CE2/E*, 정보문화사 , 2002

Terence Goggin , *Inside Secrets 03 Windows CE*, 삼각형프레스 , 1999.

<http://blog.naver.com/heyzoo/100008763771>

주 의

1. 이 보고서는 농림부에서 시행한 농림기술개발사업의 연구보고서입니다.
2. 이 보고서의 내용을 발표할 때에는 반드시 농림부에서 시행한 농림기술개발사업의 연구결과임을 밝혀야 합니다.
3. 국가과학기술 기밀유지에 필요한 내용은 대외적으로 발표 또는 공개하여서는 아니됩니다.