ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ....3
1. Управление складскими запасами на предприятии ООО «Газпром Добыча Ноябрьск»....4
1.1 Общая характеристика предприятия ООО «Газпром добыча Ноябрьск»....4
1.2 Описание процесса управления складскими запасами в ООО «Газпром добыча Ноябрьск»....9
2. Проект оптимизации стратегии управления складскими запасами....11
2.1 Проектная деятельность на начальном этапе....11
2.2 Архитектура системы и схема базы данных....12
2.3 Топология взаимодействия Mobile SMARTS и ИУС П Axapta....15
2.4 Регламент синхронизации данных между Mobile SMARTS и ИУС П Axapta....16
2.5 Функциональные требования к механизму штрихкодирования....19
2.6 Принципы разработки в Mobile SMARTS....26
3. Описание функционала управления складскими запасами с применением процесса штрихкодирования....31
3.1 Настройка параметров....31
3.2 Функциональность механизма штрихкодирование....33
ЗАКЛЮЧЕНИЕ....61
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ....64
ПРИЛОЖЕНИЕ А
ПРИЛОЖЕНИЕ Д
ПРИЛОЖЕНИЕ Г
ПРИЛОЖЕНИЕ Б
ПРИЛОЖЕНИЕ В

ВВЕДЕНИЕ

Состояние и эффективность использования производственных запасов, как самой значительной части оборотного капитала - является одним из основных условий успешной деятельности предприятия. Развитие рыночных отношений определяет новые условия их организации.

Инфляция, неплатежи и другие кризисные явления вынуждают предприятия изменять свою политику по отношению к производственным запасам, искать новые источники пополнения, изучать проблему эффективности их использования.

Поэтому для предприятия все возможные способы рационального расходования средств, одним из которых является, определение оптимальной величины производственных запасов приобретают все большую значимость.

Вашему вниманию предлагается дипломная работа по оптимизации стратегии управления складскими запасами в MS Dynamics AX (далее по тексту - ИУС П Axapta) с применением механизма штрихкодирования на базе Mobile Smarts.

Предпосылками для разработки явились следующие факторы:

- отсутствие прозрачности;

- не востребованность материалов;

- трудоемкость процесса и другие.

Все это привело к недостаточно оперативному выполнению требований бизнеса и запросов руководства. В процессе анализа выяснилось, что появилась необходимость внесения изменений в процесс снабжения.

Целью данной работы является разработка на основе анализа эффективности управления производственными запасами ОАО «Газпром Добыча Ноябрьск», а именно:

- эффективное использование ресурсов;

- снижение рисков ошибок;

- контроль на всех этапах процесса снабжения и др.

Работа состоит из введения, трёх разделов, заключения, списка использованной литературы и приложений.

Во введении определены цели и задачи, объект и предмет исследования.

Первая глава – теоретические аспекты управления запасами, на примере предприятия ОАО «Газпром Добыча Ноябрьск»: сущность и особенности.

Во второй главе – практической, проводится основная характеристика компании, проводится анализ существующей системы управления проектами. В третьей главе, проектной, представлены основные предложения и мероприятия по совершенствованию системы управления запасами в компании ОАО «Газпром Добыча Ноябрьск».

В заключении рассматриваются основные выводы по работе и степень достижения поставленной во введении цели и выполнения поставленных во введении задач.

Функциональность, разработанная в ходе дипломной работы, предназначена для оптимизации процесса материально-технического снабжения и обеспечивает прозрачность процессов планирования, поставки и распределения МТР в подразделения предприятия ОАО «Газпром Добыча Ноябрьск».

Работа содержит современные методы управления производственными запасами. В рамках работы разработан ряд практических предложений по

совершенствованию механизма управления производственными запасами предприятия.

Информационную базу исследования составляют: учебники и монографии отечественных и зарубежных специалистов в области финансового менеджмента и экономики предприятия, материалы периодической печати по наиболее значимым особенностям управления запасами на современном этапе в России [1, С.29] , данные бухгалтерской и финансовой отчетности предприятия, внутризаводская нормативно-техническая документация, практические рекомендации специалистов по управлению производственными запасами.

1. Управление складскими запасами на предприятии ООО «Газпром Добыча Ноябрьск»

1.1 Общая характеристика предприятия ООО «Газпром добыча Ноябрьск»

Общество с ограниченной ответственностью ООО «Газпром добыча Ноябрьск» создано на основании решения Совета директоров ОАО «Газпром» от 25 мая 1999 года и решения Учредителя от 30 июня 1999 года путем реорганизации в форме преобразования дочернего предприятия «Ноябрьскгаздобыча» со 100-процентным уставным капиталом ОАО «Газпром».

История ООО «Газпром добыча Ноябрьск» начинает свой отсчет с 31 мая 1977 года, когда был издан приказ № 273 о создании Газопромыслового управления «Вынгапургаз» Производственного объединения «Тюменьтрансгаз».

11 декабря 2003 года Общество зарегистрировано в качестве юридического лица в инспекции МНС России по городу Ноябрьску Ямало-Ненецкого автономного округа.

Право на осуществление производственной деятельности подтверждено наличием соответствующих лицензий (Государственного комитета РФ).

В 2010 году общая валовая добыча газа предприятием составила 83,5 млрд. м3 газа. В том числе собственная валовая добыча 52,8 млрд. куб. м., оказано услуг по добыче и подготовке газа в объеме 30,7 млрд. куб.м. На сегодняшний день предприятие занимает третье место в России по объемам добычи природного газа.

Руководство предприятием, его основными службами и подразделениями сосредоточено в руках настоящих профессионалов.

Фирменным знаком предприятия стали мобильность, концентрация усилий на быстром и качественном обустройстве и комплексном освоении небольших и средних по объемам запасов газовых месторождений. Сделать это позволяет хорошая транспортная схема, высококвалифицированный кадровый состав, рациональное расположение структурных подразделений, материально-технических баз и рабочих поселков Общества.

Низкая себестоимость добычи газа - один из главных отличительных признаков работы предприятия. Это позволяет Обществу выступать в роли генерального оператора при оказании услуг по добыче и подготовке газа для независимых недропользователей.

ООО «Газпром добыча Ноябрьск» имеет собственную программу социального развития и социальной защиты работников. Основные права работников защищены Генеральным коллективным договором ОАО «Газпром» и коллективным договором между администрацией и трудовым коллективом предприятия, который предусматривает ряд дополнительных гарантий и льгот для работников ООО «Газпром добыча Ноябрьск».

Являясь одним из градообразующих предприятий города, Общество принимает активное участие в социальной жизни региона. Один из примеров - участие в строительстве Храма Архистратига Михаила и Соборной мечети в городе Ноябрьск. Важное направление социальной политики предприятия - помощь коренному населению севера. В частности, в поселке Халясовэй Пуровского района, месте проживания родоплеменной общины Еты Яля, предприятием ведется работа по улучшению жизненных и бытовых условий коренных ненцев.

Основной профильной единицей предприятия является газовый промысел. В составе ООО «Газпром добыча Ноябрьск» входит 6 газодобывающих

промыслов: Вынгапуровский ГП, Вынгаяхенский ГП, Губкинский ГП, Комсомольский ГП, Заподно-Таркасалинский ГП, Камчатский ГП.

В своей деятельности промысел руководствуется «Положением о Газовом промысле ООО «Газпром добыча Ноябрьск», приказами и распоряжениями ООО «Газпром добыча Ноябрьск.

В состав промысла входят следующие службы и участки:

- служба добычи газа;
- геологическая служба;
- газокомпрессорная служба;
- служба энерговодоснабжения;
- участок КИПиА;
- механо-ремонтный участок;
- химлаборатория;
- ремонтно-хозяйственный участок.

Основными целями газового промысла являются: рациональная и эффективная эксплуатация газового месторождения, обеспечение заданных объемов добычи газообразных углеводородов, подготовка к транспорту согласно действующим нормам и стандартам, формирование сырьевой базы, соблюдение природоохранных мер, получение прибыли.

В соответствии с основными целями газовый промысел осуществляет следующие виды деятельности:

- добыча, подготовка к дальнейшему транспорту газообразного углеводородного сырья.
- капитальный ремонт, модернизация, реконструкция и техническое перевооружение действующих объектов, ввод новых мощностей;

- выработка теплоэнергии, электроэнергии, добыча воды, очистка стоков, организация теплоснабжения объектов;
- обеспечение производства автоматизированными системами управления;
- своевременное снабжение материально-техническими ресурсами, необходимыми для бесперебойной работы промысла;
- анализ производственно-хозяйственной деятельности промысла. Осуществление мероприятий по непревышению плановых затрат;
- содержание в исправном состоянии объектов жилищно-коммунального хозяйства, объектов социально-культурного значения;
- внедрение новых средств и методов предупреждения производственного травматизма, профзаболеваний и аварий.

Управление по обеспечению производства осуществляет Управление материально-техническим снабжением далее по тексту УМТС.

В состав УМТС входят следующие службы:

- отдел фондов и реализации;
- отдел сводной отчетности, аналитики и ценообразования.

Основными целями УМТС являются: централизация процесса управления материально техническими ресурсами, обеспечение единой системы учета договоров по закупкам материалам, реализация единого складского учета в рамках дочернего общества.

В соответствии с основными целями УМТС осуществляет следующие виды деятельности:

- обеспечение поставки МТР для ПЭН;
- обеспечение поставки МТР для КР;
- обеспечение поставки МТР для Внереализационных расходов;

- обеспечение поставки МТР для Аварийного запаса;
- обеспечение поставки МТР для ГОиЧС;
- обеспечение поставки МТР для строек, не требующих монтажа.

Служба по обеспечению автоматизации бизнес-процессов осуществляет СИУС.

В состав службы ИУС входят следующие отделы:

- отдел функционального сопровождения ИУС;
- отдел программного обеспечения;
- отдел администрирования и ведения НСИ;
- отдел системно-технического обеспечения.

Основными целями СИУС являются: разработка и внедрение информационных систем, функциональное сопровождение информационных систем, разработка и моделирование бизнес процессов, информационно-технологическая поддержка проектного управления.

В соответствии с основными целями СИУС осуществляет следующие виды деятельности:

- моделирование алгоритмов и функций системы ИУС П Axapta;
- разработка системы электронного документооборота;
- автоматизация системы бюджетного управления.

А так же в состав предприятия входят другие службы и подразделения находящиеся за рамками предметной области диплома.

1.2 Описание процесса управления складскими запасами в ООО «Газпром добыча Ноябрьск»

Одним из самых важных и сложных направлений жизнедеятельности крупного предприятия, влияющего на работу производственных подразделений по выполпнению производственного плана, является обеспечение материально-техническими ресурсами.

Процессы материальных потоков, протекающие в УМТС при поступлении МТР, представлены в приложение А на рисунке А.1.

Главной задачей УМТС является обеспечение потребности МТР промыслов и структурных подразделений для производственно-эксплуатационных нужд, капитального ремонта, выполняемого собственными силами, аварийного запаса и поставки оборудования, не входящего в смету строек и не требующего монтажа. Для выполнения обеспечения материально техническими ресурсами, далее по тексту МТР, в соответствии с производственными нуждами УМТС совместно с промыслами и подразделениями осуществляет мероприятия связанные с планированием и приобретением МТР.

На сегодняшний день операции поступления, движения и распределения МТР в подразделения Общества отражаются в информационно-управляющей системе Общества Axapta, далее ИУС П Axapta.

До внедрения проекта процесс снабжения выглядел следующим образом:

- В структурных подразделениях осуществлялся ввод заявок потребности МТР, в зависимости от планируемого периода вовлечения в производство. Затем перераспределялись имеющиеся на складах Общества остатки, и определялось количество к закупке, данные передавали инженеру УМТС, который формировал свод по Обществу в целом, на основании которого в Управлении по комплектации заключались договора на поставку.

- При поступлении МТР в Общество, на основании Контракта формировались закупки в системе, а так же автоматически создавались бухгалтерские и складские проводки. Материалы проходили входной контроль и поступали на Центральный склад.
- Сложности начинались при определении заказчика поставки. Особо трудоемким был процесс определения наибольшего приоритета, при поступлении недостаточного количества материалов. Требовалось ручное сопоставление полученных номенклатур с данными заявочной кампании и формирование документов распределения. Возникали ошибки человеческого фактора из-за необходимости контролировать недопоставку МТР подразделениям, особенно при поступлении номенклатур разными партиями.
- После того как распределили кому и сколько материалов необходимо отпустить, инженер УМТС по почте сообщал промыслам о необходимости их отгрузки. Экспедиторы подразделений, совместно с кладовщиками центральных складов, согласно документам распределения комплектовали МТР для отгрузки.

Все вышеперечисленные факторы приводили к ухудшению работы УМТС по всем направлениям деятельности, начиная с планирования МТР и заканчивая распределением МТР на промысла, что в целом ставило под угрозу работу предприятия и порождало постоянное недовольство производственных служб по поводу работы УМТС.

2. Проект оптимизации стратегии управления складскими запасами

2.1 Проектная деятельность на начальном

Открытие проекта по оптимизации стратегии управления складскими запасами в ИУС П Axapta состоялось на заседании проектного комитета при участии руководителей отделов ПЭО, ОФИР, СИУС. Назначена рабочая группа, в составе сотрудников от УМТС и СИУС. Возложена ответственность за написание функциональных требований, разработку функционала, подготовку и загрузку справочных данных на сотрудников службы ИУС и обеспечение разъяснений и оказания методологической помощи в реализации проекта на сотрудников УМТС.

Для внедрения проекта был изучен процесс материально-технического обеспечения в целом. Обозначены слабые стороны. Изучалось оборудование и программные продукты изложенные в приложении Д таблица Д.1., представленные на рынки для внедрения механизма штрихкодирования.

Предпочтение было отдано программной платформе Mobile SMARTS.

Mobile SMARTS - это программная платформа для разработки корпоративных мобильных решений под мобильные терминалы сбора данных (ТСД), микрокиоски (прайс-чекеры) и КПК. Mobile SMARTS обеспечивает разработку, внедрение, собственно работу и последующую поддержку мобильной части (front-end) транспортных, складских, торговых и производственных систем.

В состав платформы входит:

- панель управления (средство разработки и администрирования);
- сервер мобильных устройств и сервер печати;

- средства интеграции (коннектор к серверу, компоненты COM, бизнес-коннекторы к базам 1С, Axapta, Navision, MS SQL);
- программа-установщик клиента Mobile SMARTS под мобильные устройства;
- клиент Mobile SMARTS под персональный компьютер для тестирования и отладки (своего рода «эмулятор»).

Использование Mobile SMARTS в качестве средства интеграции мобильных устройств с информационной базой предприятия имеет ряд технических преимуществ. Главным образом все они сводятся к грамотному использованию возможностей такого дорогостоящего аппаратного обеспечения, как ТСД со встроенным сканером штрих-кода.

После закупки оборудования и ПО, была изучена документация, разработаны мероприятия по интеграции закупленного ПО и оборудования с учетной системой Общества ИУС П Axapta. В рамках проекта был разработан механизм штрихкодирования.

2.2 Архитектура системы и схема базы данных

Архитектура системы клиент-серверная представленная на рисунке 2.2, но клиент Mobile SMARTS может работать в полностью автономном режиме (оффлайн) вообще без какого-либо сервера. Это удобно для решений в стиле загрузка\ выгрузка файлов напрямую между ТСД и учетной базой (выездная торговля, выездная инвентаризация и т.п. без Wi-Fi и мобильного интернета).

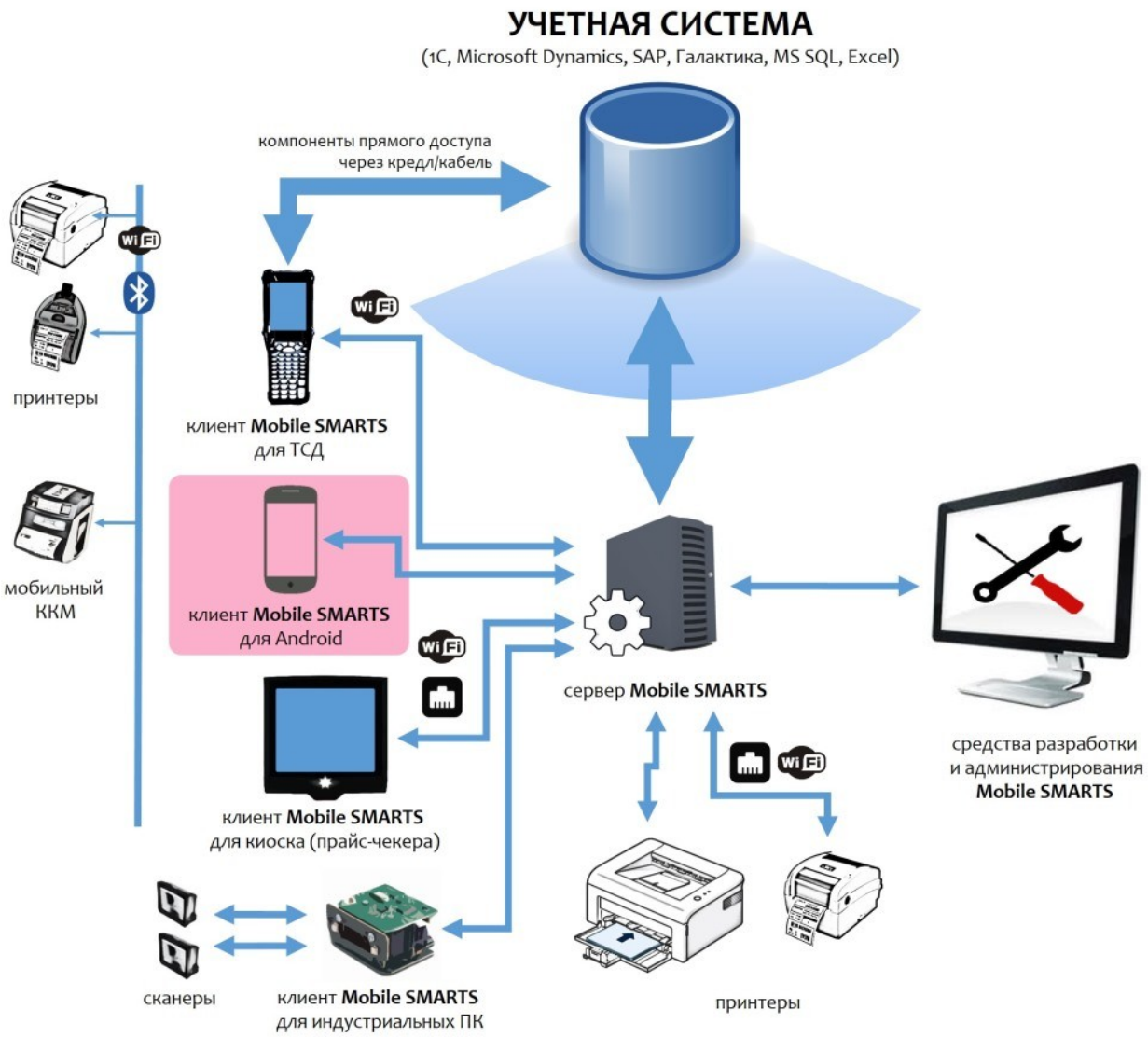

Рисунок 2.2 - Схема интеграции Mobile SMARTS и Axapta

Сервер Mobile SMARTS обязателен для беспроводных вариантов решения, когда используется Wi-Fi или мобильный интернет. Основная задача сервера- обслуживание запросов на получение\ отправку документов, номенклатуры и других данных от клиентов с мобильных ТСД. Сервер раздает клиентам документы\ задания в соответствии с указаниями, очередностью, приоритетом, и получает от них результаты работы.

Помимо этого сервер занимается хранением серверных справочников, хранением локальных справочников ТСД для отправки клиентам, а также временным хранением документов ТСД.

Сервер Mobile SMARTS - всего лишь промежуточное хранилище, и он не реализует никакой собственной прикладной логики по работе с документами и данными. Обработкой документов занимаются мобильные устройства, а подготовкой данных для мобильных устройств и обработкой результата должна заниматься прикладная учетная система, в нашем случае ИУС П Axapta, которая использует Mobile SMARTS для решения задач мобильной автоматизации.

Для реализации проекта были разработаны процедуры выгрузки и загрузки данных из учетной системы ИУС П Axapta на сервер Mobile SMARTS. Разработаны и доработаны существующие приложения по обработке принимаемых данных и корректному их размещению в информационном пространстве учетной системы ИУС П Axapta. Так же в программной среде Mobile SMARTS была разработана структура удовлетворяющая потребностям по обработке складских операций. Подключены и настроены ТСД и принтеры (штрих-кодов).

Описание функционирования объекта производится в нотации методологии функционального моделирования SADT, как подмножества стандарта IDEF0 *[12, С. 200-500].* Эта нотация позволяет представить функции предприятия, функциональные связи и данные (информацию или объекты), которые связывают эти функции.

На функциональной модели, изображенной на рисунке 2.3, представлена взаимосвязь справочников и таблиц, хранящих данные об обеспечении предприятия необходимыми МТР. Структура таблиц изображена в приложении Г.

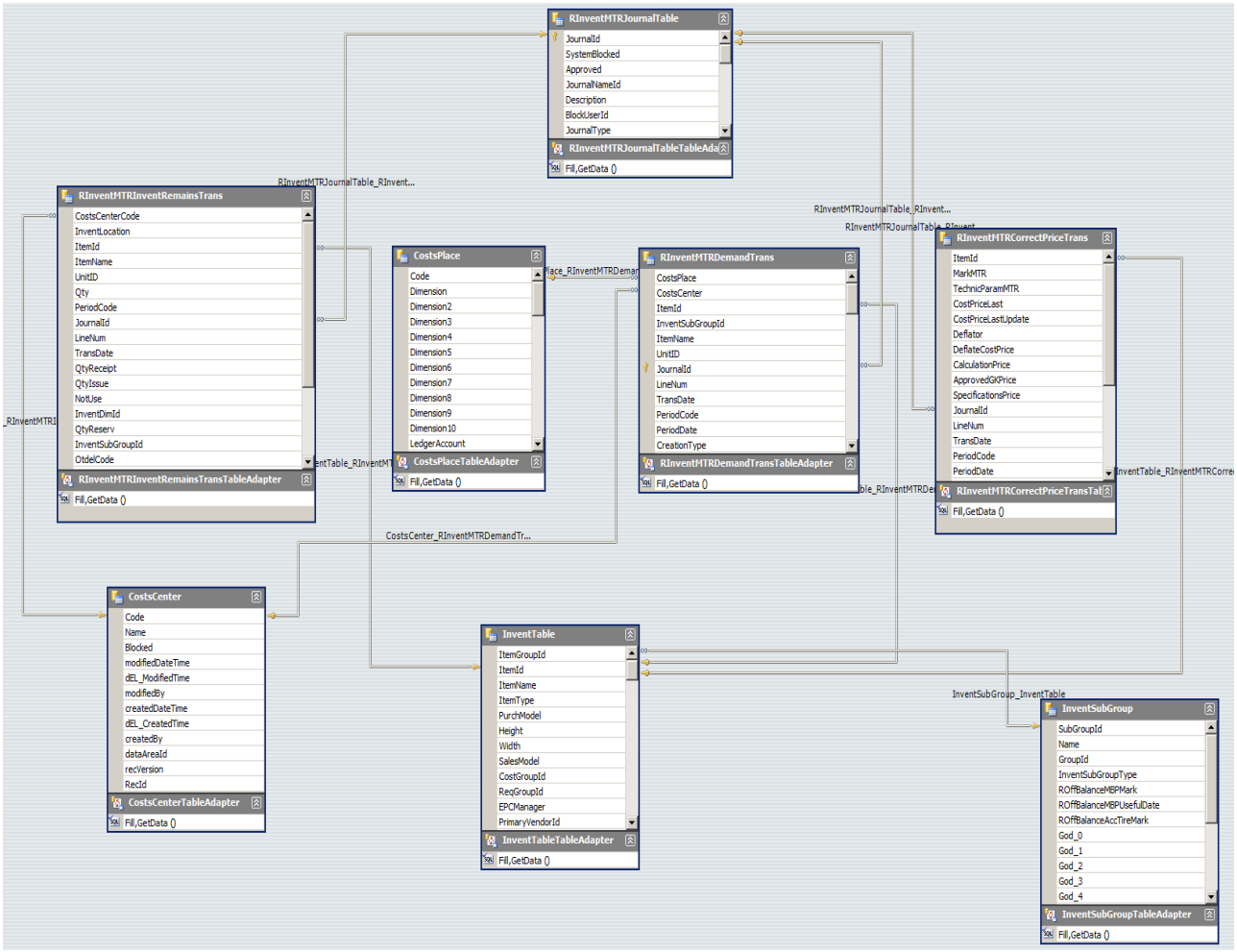

Рисунок 2.3 - Представлена концептуальная модель базы данных

2.3 Топология взаимодействия Mobile SMARTS и ИУС П Axapta

Mobile SMARTS - это программный продукт, который позволяет осуществлять передачу данных из учетной системы на ТСД и обратно. Для этого разворачивается сервер Mobile SMARTS, на ТСД устанавливается клиентская часть, разработанная на платформе Microsoft. NET Compact Framework. Обновление клиентского приложения происходит декларативно через установку набора свойств в нужные значения на сервере Mobile SMARTS, а затем обмена между сервером и клиентом по протоколу HTTP. Связь сервера Mobile SMARTS и учетной системы ИУС П Axapta осуществляется через компоненту доступа на основе механизма OLE Automation.



Рисунок 2.4 - Схема движения информационных потоков

Последовательность передачи данных из учетной системы ИУС П Axapta на ТСД и обратно выглядит следующим образом:

- из учетной системы, посредством разработанных в ней процедур, на сервер Mobile SMARTS выгружается конфигурационная информация, склады, пользователи, их права и данные о не разнесенных Закупках;
- на терминал загружается информация с сервера Mobile SMARTS. И с его помощью пользователи выполняют присвоение штрих-кодов материалам из закупки, а так же отражают операции по их движению. На ТСД формируются журналы и выгружаются на сервер. Далее с помощью разработанных процедур загрузки, данные с сервера Mobile SMARTS, загружаются в учетную систему.

В рамках проекта созданы новые и доработаны существующие процедуры в учетной системе и на сервере Mobile SMARTS.

2.4 Регламент синхронизации данных между Mobile SMARTS и ИУС П Axapta

Первоначальная загрузка данных предусматривает операцию по загрузке справочников:

- номенклатурный справочник;

- склады;
- сотрудник;
- аналитика;
- единицы измерения в полном объеме.

Периодические операции по загрузке данных представлены ниже в таблице.

Таблица 1 - Операции по загрузке данных

Данные	Область применения	Частота обновления	Примечания
Справочники			
Номенклатурный справочник	Обеспечит связь виртуального номенклатурного номера и штрих-кода. Позволит идентифицировать штрих-код, отобразит наименование номенклатуры.	При каждом соединении ТСД с сервером Mobile SMARTS, при вызове процедуры загрузке.	Загрузке подлежат данные, созданные вновь или измененные со времени последней загрузки.
Склады	Будет использоваться при формировании журналов Отпуска в подразделения, Перенос, Акт списания, Выдача СиОС, Выдача спецодежды.	По требованию.	Загрузка справочника в полном объеме.
Сотрудник (Сотрудник, ФИО, ЦЗ, МВЗ, Должность)	Будет использоваться при формировании журналов Выдача СиОС, Выдача спецодежды.	По требованию.	Загрузке подлежат данные, созданные вновь или измененные со времени последней загрузки.
Аналитика МВЗ и ЦЗ		По требованию.	Загрузке подлежат данные, созданные вновь или измененные со времени последней загрузки.
Единицы измерения	Будет использоваться при формировании журналов: - отпуска в подразделения; - перенос; - акт списания;	По требованию.	Загрузка справочника в полном объеме.

	- выдача СиОС; - выдача спецодежды.		
Ежедневные данные			
Закупки	Необходимо для выбора обрабатываемых в текущий момент номенклатур для формирования штрих-кода.	По требованию.	Загрузке подлежат данные закупок в статусе «Открыто», у которых признак штрих-кода «Нет», созданные вновь или измененные со времени последней загрузки.
Журналы	В ТСД предусмотреть признак «Загружен в ИУС П Axapta». Признак проставлять автоматически после загрузки журнала в ИУС П Axapta .	По требованию.	Журналы ТСД с признаком «Загружен в ИУС П Axapta » редактировать запрещено.

2.5 Функциональные требования к механизму штрихкодирования

В таблице 2 описаны функциональные требования к операциям, осуществляемым на рабочем месте кладовщика:

Перенос: операция переноса связана с изменением складской аналитики номенклатурной единицы: склада, табельного номера в рамках одного подразделения.

Отпуск в подразделения: операция связана с перемещением МТР с одного подразделения на другое, связана с изменением складской аналитики номенклатурной единицы: центра затрат, склада.

Выдача спецодежды: операция используется для выдачи спецодежды в эксплуатацию. В процессе формирования журнала формируются две складские проводки: расход со склада и приход на табельный номер сотрудника.

Выдача СиОС: операция используется для выдачи смывающих и обезвреживающих средств в эксплуатацию. В процессе формирования журнала

формируются две складские проводки: расход со склада и приход на табельный номер сотрудника

Акт списания: операция связана со списанием МТР со склада.

Ограничения:

- Для ранее оприходованных номенклатур, имеющих заводской штрих-код, необходимо генерировать штрих-код в ИУСП Axapta. Использование заводского штрих-кода возможно только с внедрением данной функциональности, при обработке закупок в статусе «Открыто».
- Журналы по расходу, созданные в ИУС П Axapta, не отображаются на ТСД. Необходимо создавать журналы, используя ТСД.
- Для номенклатурного номера есть возможность хранения нескольких штрих-кодов, при этом актуальным будет являться только один с признаком «Актуален».
- Автоматизация описанных процессов применимы, только к складам, участвующим в проекте. В справочнике Склады не будет проставлен признак штрих-код.

Таблица 2 - Функциональность механизма штрихкодирования для рабочего места кладовщика

№	Наименование операции	Описание	Примечание
1	Предварительные операции		
1.1	Выгрузка справочников из ИУС П Axapta на сервер Mobile SMARTS Справочники: - сведения о номенклатуре; - склады; -аналитики; - единицы измерения.	Разработать процедуры для первоначальной загрузки справочников и обновлений в пакетном режиме и по запросу пользователя в соответствии с Регламентом предприятия. Первоначальная загрузка включает в себя передачу данных в полном объеме. При обновлении в пакетном режиме загружаются вновь введенные или измененные на Дату обновления	Рисунок 2.4.

		данные.	
1.2	Доработка формы «Закупка»	Добавить поле «Штрих-код» для признака того, что в закупке присутствуют номенклатуры, у которых нет штрих-кода с признаком «Актуален».	Если на закупке установлен признак штрих-кода «Нет», то обработка Отборочной накладной невозможна до тех пор, пока всем номенклатурным единицам закупки не присвоен штрих-код.
3	Доработка формы «Журналы»	Добавить признак «Загружен из ТСД»	
2	Учет операций		
2.1	Закрепление штрих-кода за номенклатурой	Инженер формирует в ИУС П закупки в статусе «открыто», признак «ШК» устанавливается в «Нет», если на номенклатурах закупки отсутствует штрих-код с признаком «Актуален». Кладовщик не может обработать на закупке отборочную накладную, пока в ней есть строки с номенклатурами без штрих-кода.	Для инженера бизнес-процесс по прежней схеме, ничего не изменяется
2.2	Генерация штрих-кода в ИУС П Axapta.	1. Формирование списка номенклатур из Закупок в статусе «Открыто». В меню Расчеты с поставщиками\ Обычные формы\ Закупка выбрать закупки, в которых есть номенклатуры с незаполненным штрих-кодом. Это можно сделать, наложив фильтр по полю «ШК». Затем перейти от конкретной номенклатуры в справочник номенклатур, выбрать пункт Настройки\Штрих-код и сгенерировать штрих-код. 2. Формирование штрих-кода на номенклатурном справочнике В Управлении запасами\ Обычные формы\ Сведения о номенклатуре – Настройки\ Штрих-код для приходуемых номенклатур сгенерировать штрих-код.	При поступлении материалов без штрих-кода завода изготовителя
2.3	Загрузка штрих-кода завода изготовителя в ИУС П Axapta	Формирование списка номенклатур из Закупок в статусе «Открыто». В меню Расчеты с поставщиками\ Обычные формы\ Закупка\Функции разработать функцию выгрузки	При поступлении материалов, имеющих штрих-код завода изготовителя в соответствии с

		номенклатур открытых закупок, у которых отсутствует штрих-код, т.е. поле «ШК» с признаком «Нет», на сервер на Mobile SMARTS. При этом необходимо проверять выгружалась ли данная Закупка ранее, и если да, то предлагать повторную загрузку либо отмену. Синхронизация с сервером Mobile SMARTS. 1. В результате синхронизации на экране ТСД пользователю должен быть доступен список открытых закупок (номер закупки, признак «ШК»). При выборе закупки отображаться должны все номенклатурные номера и штрих-коды к ним (при наличии). 2. Сканирование штрих-кода Необходимо выбрать пункт на ТСД Присвоение номенклатуре штрих-кода, затем выбрать закупку, откроется список номенклатурных номеров Закупки. Находясь на номенклатуре, отсканировать штрих-код на МТР, нажать «ОК». При этом данные сохранятся на ТСД. 3. После того, как ко всем номенклатурам будут привязаны штрих-коды, у закупки признак «ШК» устанавливается в «ДА» и данные можно передавать в ИУС П Axapta. Загрузка отсканированных штрих-кодов в ИУС П с Mobile SMARTS. 4. В Управлении запасами\Обычные формы\Сведения о номенклатуре\Функции разработать процедуру по загрузке номеров штрих-кодов по обработанным номенклатурам. 5. Реализовать проверку на уникальность загружаемых штрих-кодов. В случае неуникальности штрих-кода предлагать сгенерировать штрих код системой. Отражать принятые штрих-коды на форме Управлении запасами\Обычные формы\Сведения о номенклатуре – Настройки\Штрих-код.	рисунком 9.

2.4	Печать этикеток штрих-кода	1. В Расчетах с поставщиками\ Обычные формы\ Закупка\Функции добавить пункт Печать Этикетки» 2. Разработать процедуру печати операции этикетки	Печать штрих-кода с признаком «Актуален»
2.5	Регистрация операций по расходу МТР в ТСД	1. Для регистрации данных о движении МТР разработать в ТСД процедуры для Переноса, Отпуска в подразделение, Акта списания, Выдачи СиОС. 2. Предусмотреть наличие функций по СОЗДАНИЮ, УДАЛЕНИЮ или РЕДАКТИРОВАНИЮ журнала.	На ТСД будут отображаться только журналы, созданные на ТСД
2.6	Функция СОЗДАНИЕ	Создание нового журнала. 1. Необходимо указать дату операции 2. Выбрать склад, с которого будет осуществлено движение 3. Дополнительно для операций: • Перенос, Отпуск в подразделение выбрать склад, на который будут переданы МТР; • Акт списания указать МВЗ списания; • Выдача СиОС, Выдача спецодежды указать Табельный номер сотрудника. 4. Для каждого МТР необходимо отсканировать штрих-код. На экране ТСД отобразятся данные о номенклатурах, сопоставленных с штрих-кодом (номенклатурный номер, наименование, единица измерения). 5. В поле «Количество» внести значение для каждой номенклатуры. 6. Нажать кнопку «Сформировать журнал» или «Отмена». 7. Кнопка ««Отмена отменит прежние операции и вернет форму к выбору складских операций. 8. Полностью сформированный журнал помечается как готовый, устанавливается признак «Одобрен». 9. На одобренном журнале для дальнейшей загрузки в ИУС П Axapta необходимо установить	Рисунок 2.5, 2.6, 2.7, 2.8. Предусмотреть возможность выбора аналитики из справочника и ручной ввод. При ручном вводе осуществлять проверку на соответствие введенного параметра данным справочников ТСД.

		признак «Загрузить в АХ».	
2.7	Функция РЕДАКТИРОВАНИЕ	Изменение количества в строках журнала, удаление или добавление строк. В случае, когда на журнале стоит признак «Загрузить в АХ», редактировать журнал будет запрещено.	
2.8	Функция УДАЛЕНИЕ	Удаление журнала возможно при выключенном признаке «Одобрен».	
2.9	Экспорт данных с ТСД на сервер Mobile SMARTS	Для передачи данных по операциям движения МТР и штрих-кодам, от завода-изготовителя, разработать синхронизацию ТСД и сервера Mobile SMARTS	
2.10	Экспорт данных с сервер Mobile SMARTS в ИУС П АХ	1. Для журналов Перенос, Отпуск в подразделение, Акт списания, Выдача СиОС, расположенных в Управлении запасами\Журналы\ Проводки по номенклатуре, разработать операцию загрузки Загрузка данных с сервера Mobile SMARTS в ИУС П Axapta. 2. После загрузки данных с ТСД на сервер Mobile SMARTS выводить на ТСД список загруженных журналов. 3. Реализовать возможность выбора одного, нескольких или всех журналов, загруженных на сервер Mobile SMARTS для удаления их с ТСД. 4. На всех журналах в ИУС П Axapta, созданных на основании журналов ТСД, установить признак «Загружен из ТСД». 5. После операции Загрузка данных с сервера Mobile SMARTS возможно дальнейшая работа с журналом в ИУС П Axapta, например, редактирование поля документ-основание.	Загружаться будут лишь журналы, у которых будет установлен признак «Загрузить в АХ».

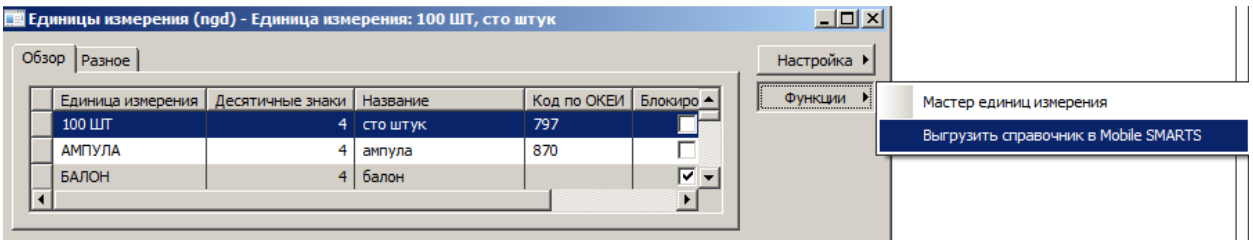

Рисунок 2.5 - Выгрузка справочников в ТСД

2.6 Принципы разработки в Mobile SMARTS

Mobile SMARTS предоставляет разработчику своеобразные «кубики Lego», называемые действия**,** из которых он может построить алгоритм работы мобильного устройства. Действия – это достаточно высокоуровневые блоки, и позволяют настроить как логику работы, так и пользовательский интерфейс приложения.

«Кружки и стрелочки», нарисованные на листе бумаги, согласно рисунку 5, в процессе обсуждения того, как кладовщик будет работать с мобильным устройством, переводятся программистом в «кубики Lego», согласно рисунку 6, действий Mobile SMARTS, а Mobile SMARTS исполняет их как экраны на мобильном устройстве как отображено на рисунке 7.

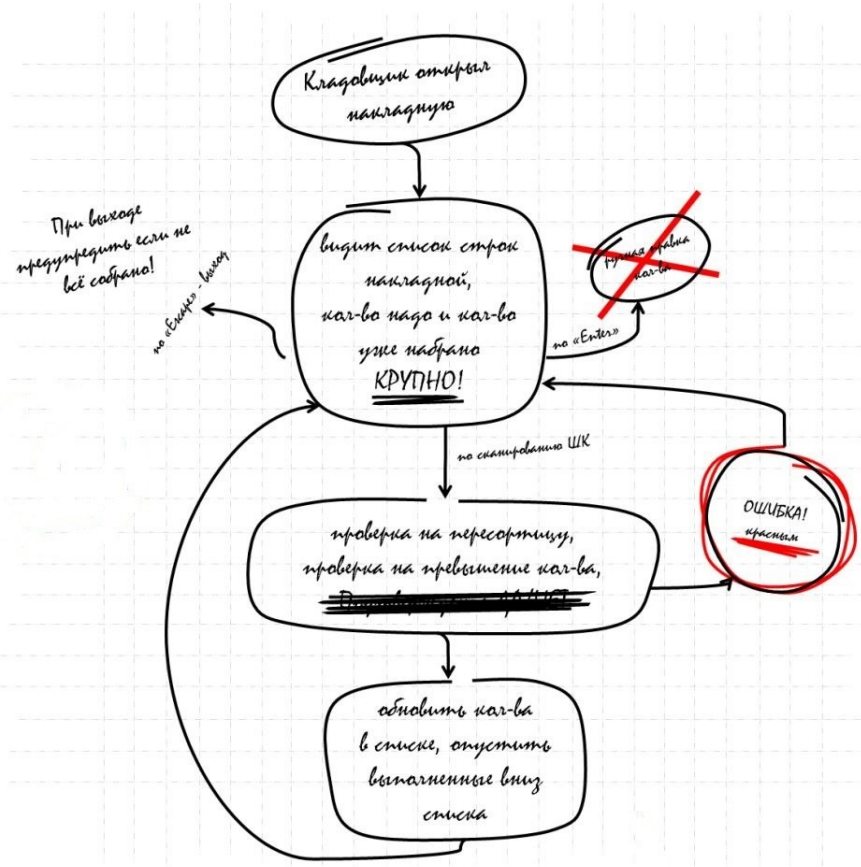

Рисунок 2.6 – «Черновая» схема процесса

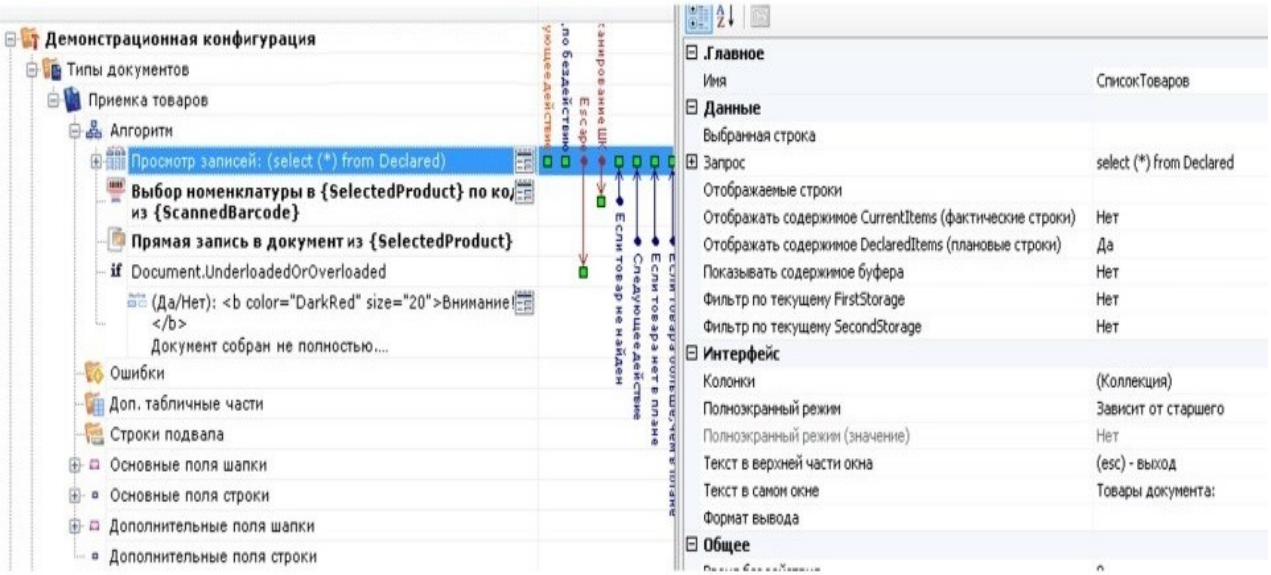

Рисунок 2.7 - Панель управления Mobile Smarts

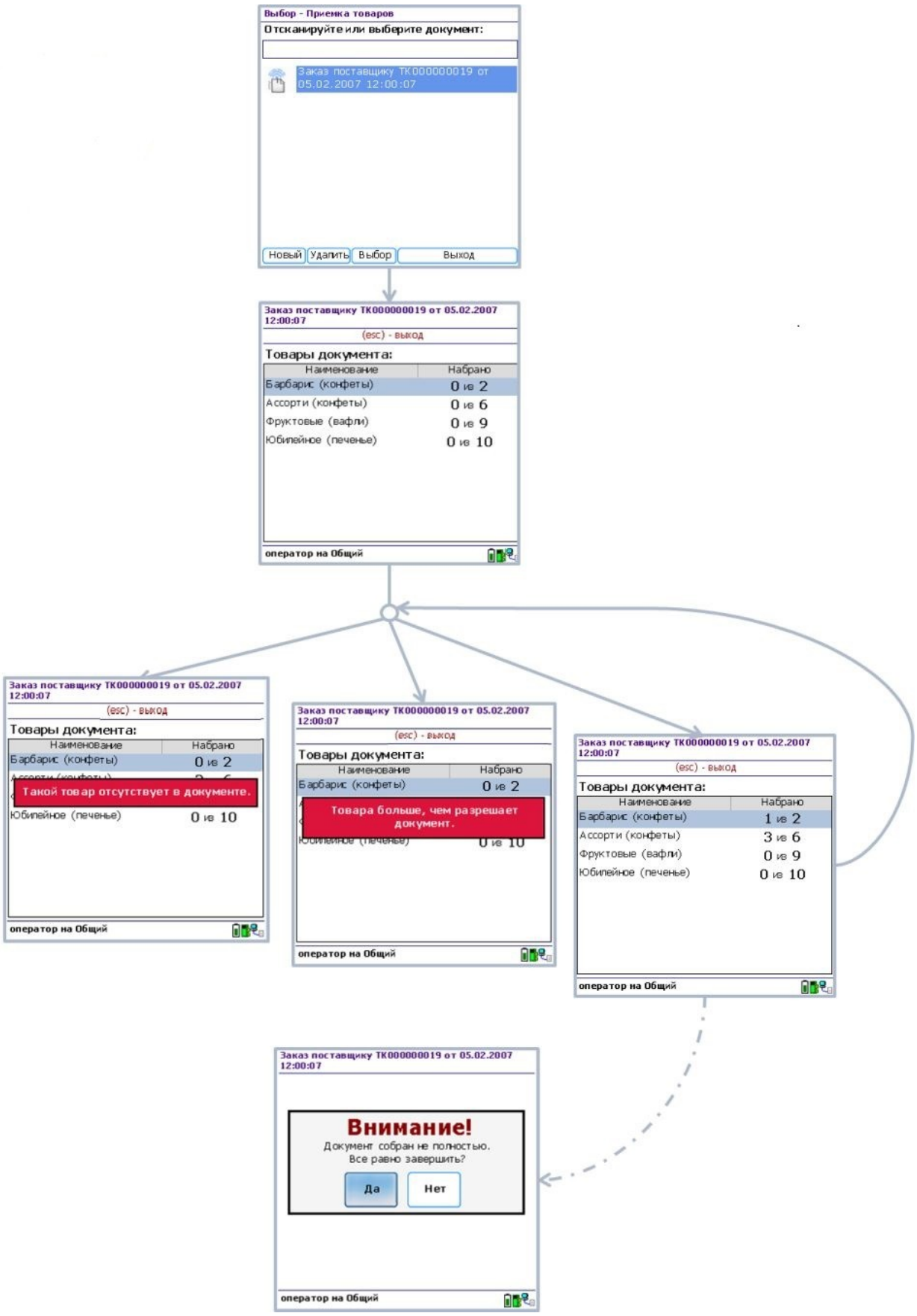

Рисунок 2.8 - Интерфейс пользователя на ТСД

Mobile SMARTS изначально «гибка» под разработку учетных бизнес-приложений, которые будут иметь дело с товарами, штрих-кодами, документами и т.п. Поэтому такие понятия как справочник номенклатуры, пользователи, типы документов и т.п. - уже заранее заложены в системе и не требуют программирования. Однако, это не мешает разработчику добавить любые дополнительные поля в карточку номенклатуры или шапку документа, добавить в документ новые колонки, новую табличную часть и т.п. Кроме того, предусмотрено добавление любых пользовательских справочников.

3. Описание функционала управления складскими запасами с применением процесса штрихкодирования

3.1 Настройка параметров

Настройки штрих-кодов

Перед началом работы со штрих-кодами необходимо создать настройки для используемых типов штрих-кодов в форме в соответствии с рисунком 3.1. Настройка штрих-кода (ИУС П Axapta\ Основное\ Настройка\ Настройка штрих-кода):

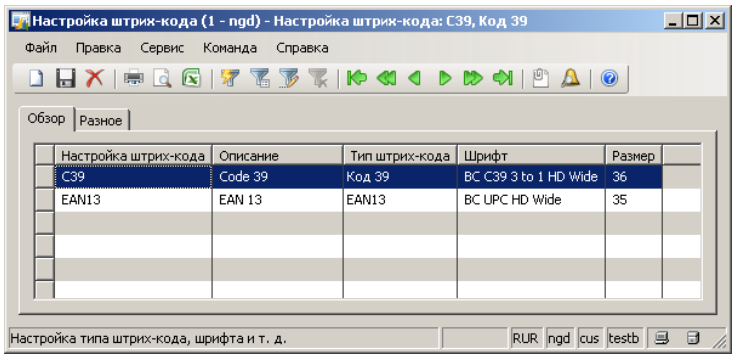

Рисунок 3.1 - Форма настройки штрих-кода

Настройка списка складов

Чтобы ограничить список доступных складов в форме, как отображено на рисунке 3.2., выбор операции (ИУС П Axapta\ Управление запасами\ Периодические операции\ Выбор операции), назначьте складам альтернативные адреса с типом «Сотрудник».

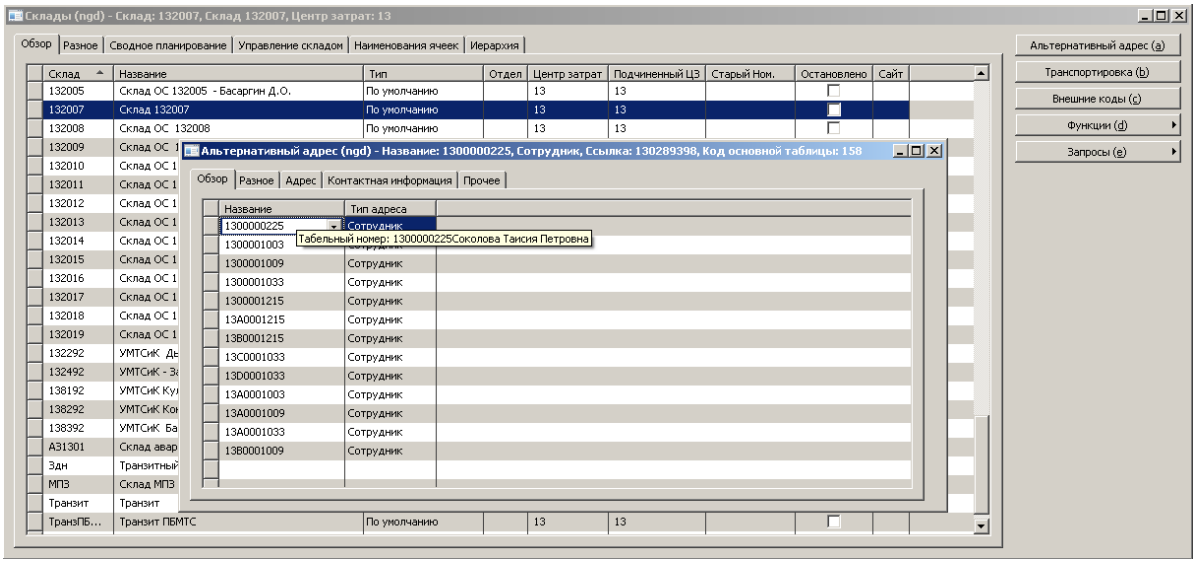

Рисунок 3.2 - Форма настройки складов

Если у склада нет ни одного альтернативного адреса с типом «Сотрудник», то в форме Выбор операции этот склад будет доступен всем сотрудникам.

Примечание. Перечень отображаемых складов также может быть ограничен доступом на уровне записей в модуле «Администрирование».

Настройка терминала сбора данных

В свойствах терминала сбора данных необходимо настроить префикс «Ctrl+L», отправляемый приложениям перед отсканированными данными.

Автоматический запуск формы Выбор операции

Для автоматического открытия формы Выбор операции (Управление запасами\ Периодические операции\ Выбор операции) при запуске клиента Microsoft Dynamics AX в конфигурации клиента, согласно рисунка 3.3., необходимо задать команду «autorun» и путь к файлу настроек автозапуска в параметре «startupcmd» (команда, которая должна выполняться при запуске):

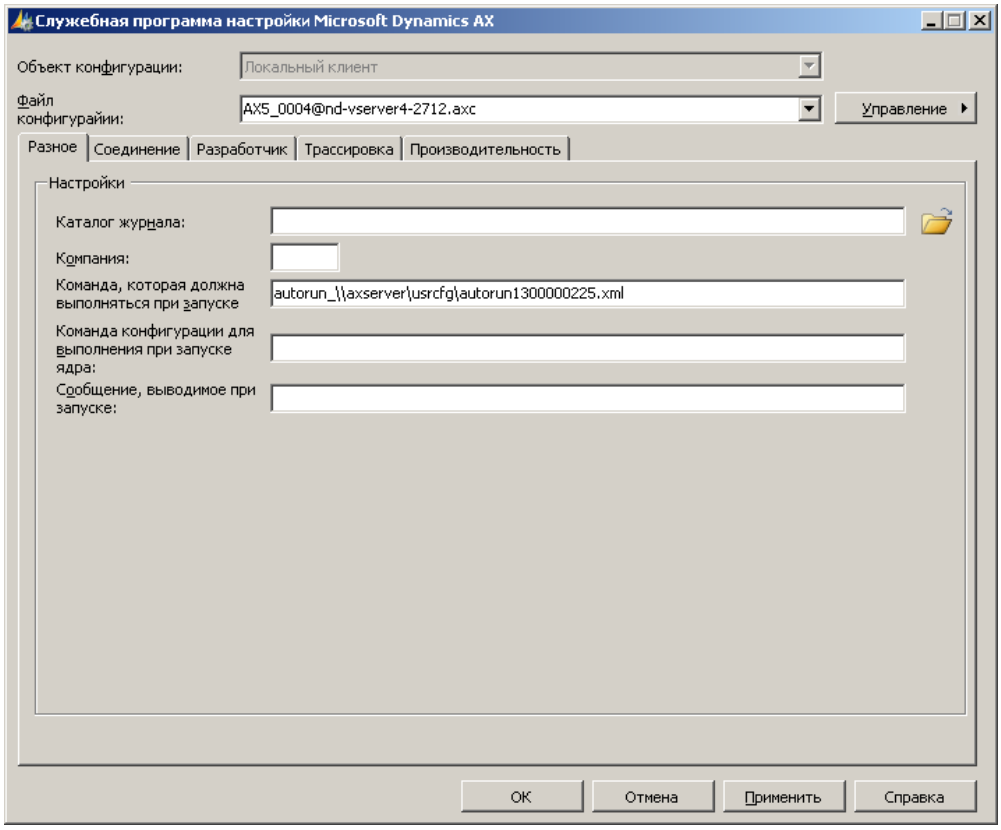

Рисунок 3.3 - Форма настройки служебных программ

Для автозапуска у пользователя должен быть доступ на просмотр соответствующего меню. Доступ к меню «InventBarcodeFormAutostart» настраивается в ветке «Контроль доступа: Управление запасами\ Разное\ Автозапуск формы выбора журналов для штрихкодирования».

3.2 Функциональность механизма штрихкодирование

Изменения в Заказе на покупку

Инженер формирует в ИУС П Axapta (Расчеты с поставщиками\ Обычные формы\ Закупка) заказы на покупку в статусе «Открыто».

На строках закупки признак «ШК» устанавливается в «Нет»в соответсвии с рисунком 3.4., если на номенклатуре нет присвоенного штрих-кода в статусе «Актуален».

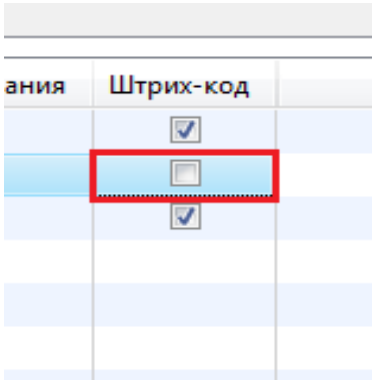

Рисунок 3.4 - Признак активности штрих-кода на строках закупки

Кладовщик не может обработать на закупке накладную (поля не доступны согласно рисунку 3.5), пока в ней есть строки с номенклатурами без актуального штрих-кода.

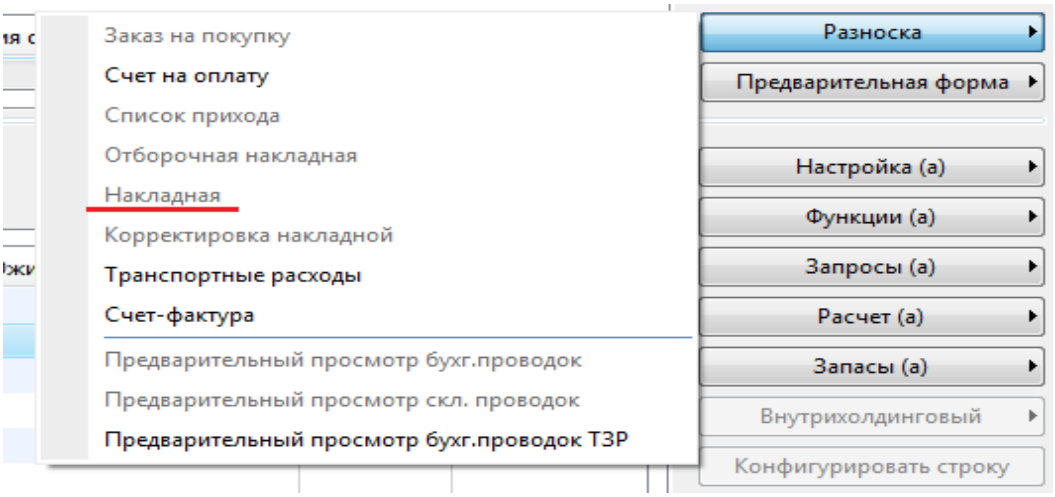

Рисунок 3.5 - Активность возможных операций в закупке

Генерация штрих-кода в ИУС П Axapta

Существует 2 способа генерации штрих-кода:

Из закупки для прихода МТР. В меню Расчеты с поставщиками\ Обычные формы\ Закупка\ выбрать закупку, в которой есть номенклатуры, с незаполненным штрих-кодом, приведено на рисунке 3.6.

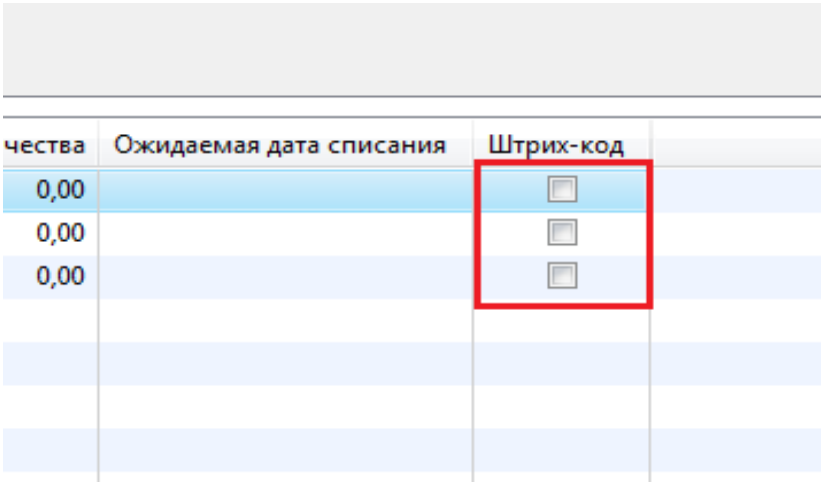

Рисунок 3.6 - Признак активности штрих-кода на строках закупки

Затем, на строках закупки, выбрать пункт Функции\ Создать штрих-код, согласно рисунка 3.7.

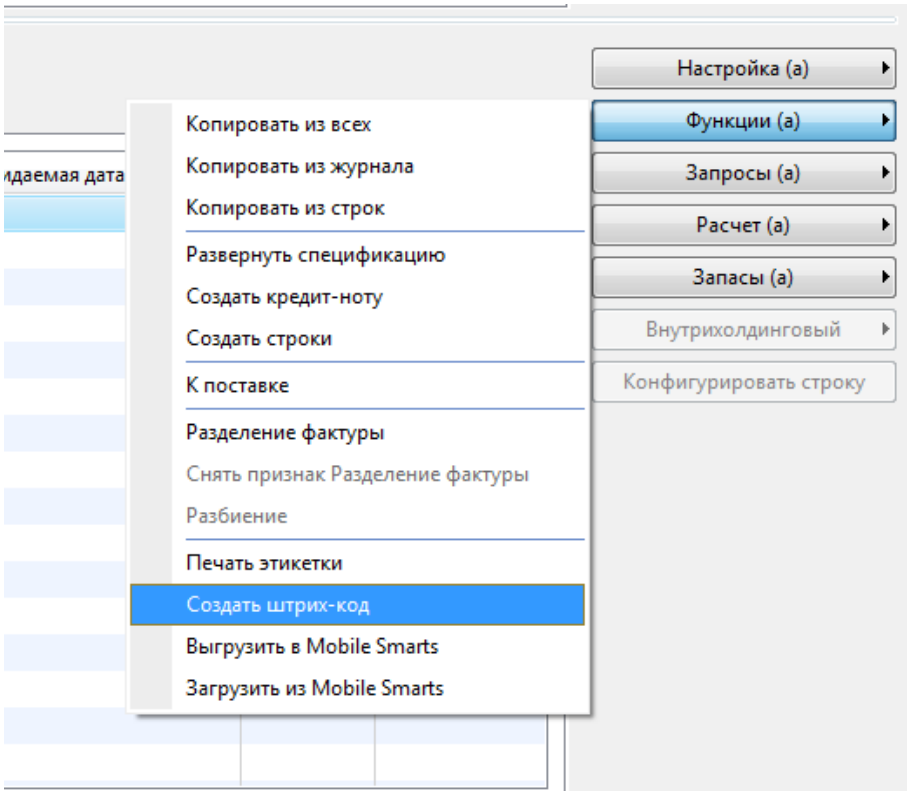

Рисунок 3.7 - Функции на строках закупки

В появившейся форме отображаются все номенклатуры без штрих-кода, согласно рисунку 3.8.

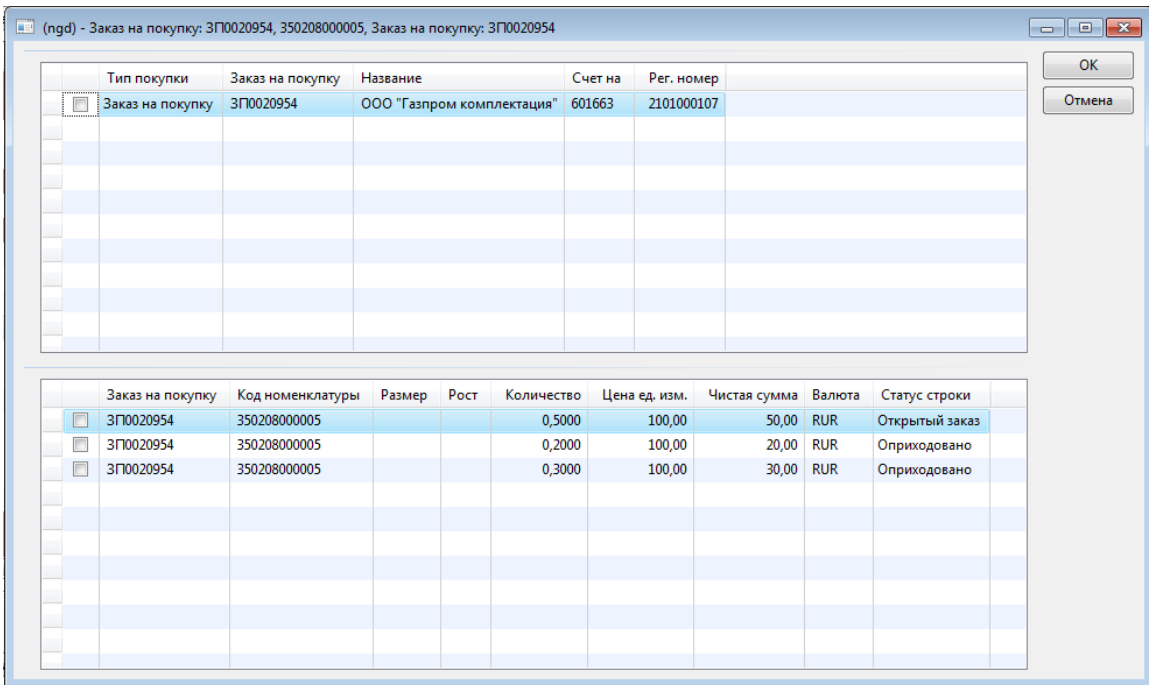

Рисунок 3.8 - Форма присвоения штрих-кодов

Необходимо отметить позиции, у которых нужно сгенерировать штрих-код и нажать кнопку ОК.

На строках закупки в поле «Штрих-код» появится отметка «Актуально».

Из номенклатурного справочника для остатков. Необходимо зайти в Управление запасами\ Обычные формы\ Сведения о номенклатуре\ Настройки\ Штрих-код согласно рисунку 3.9.

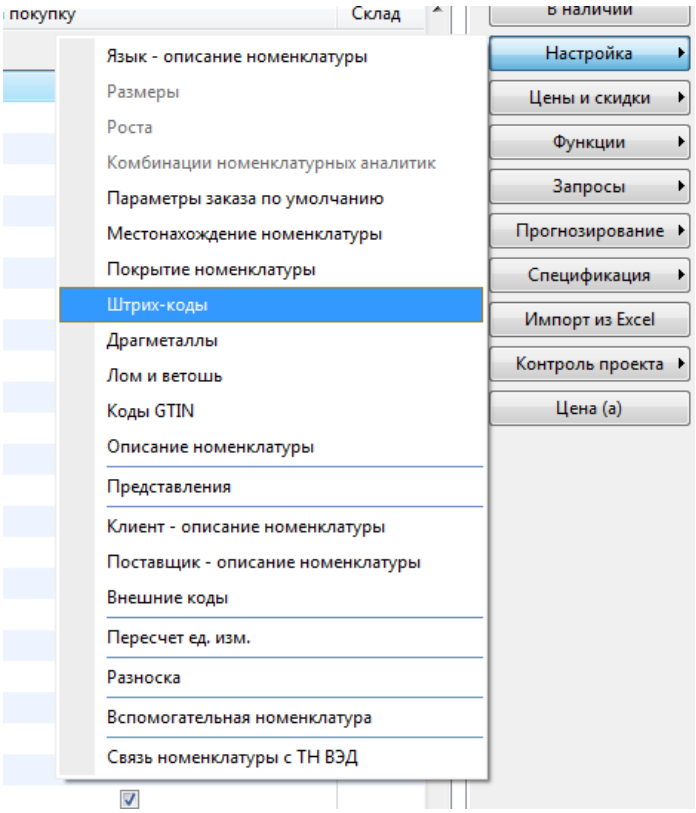

Рисунок 3.9 - Настройки в номенклатурном справочнике

Откроется форма для присвоения штрих-кода.

Для штрих-кодов, генерируемых вручную, необходимо выбирать тип C39 согласно рисунка 3.10.

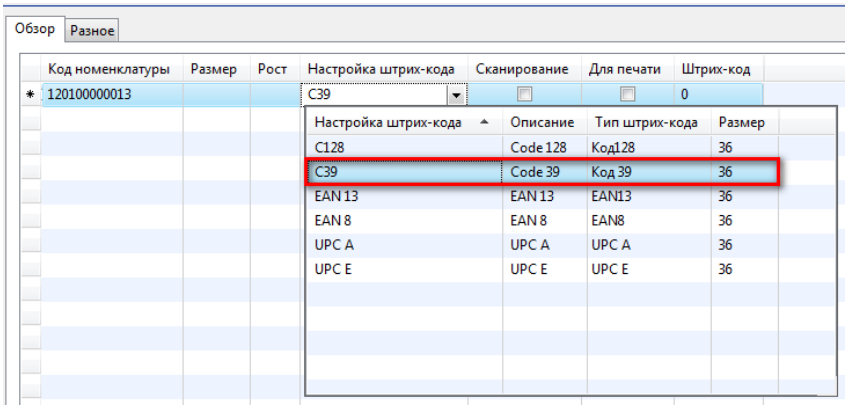

Рисунок 3.10 - Форма настройки штрих-кода

В поле «Штрих-код» автоматически сгенерируется номер.

Использование штрих-кода завода изготовителя в ИУС П Axapta

В меню Расчеты с поставщиками\ Обычные формы\ Закупка выбрать закупку, выбрать пункт Функции/ Выгрузить в Mobile Smarts, в соответствии с рисунком 3.11.

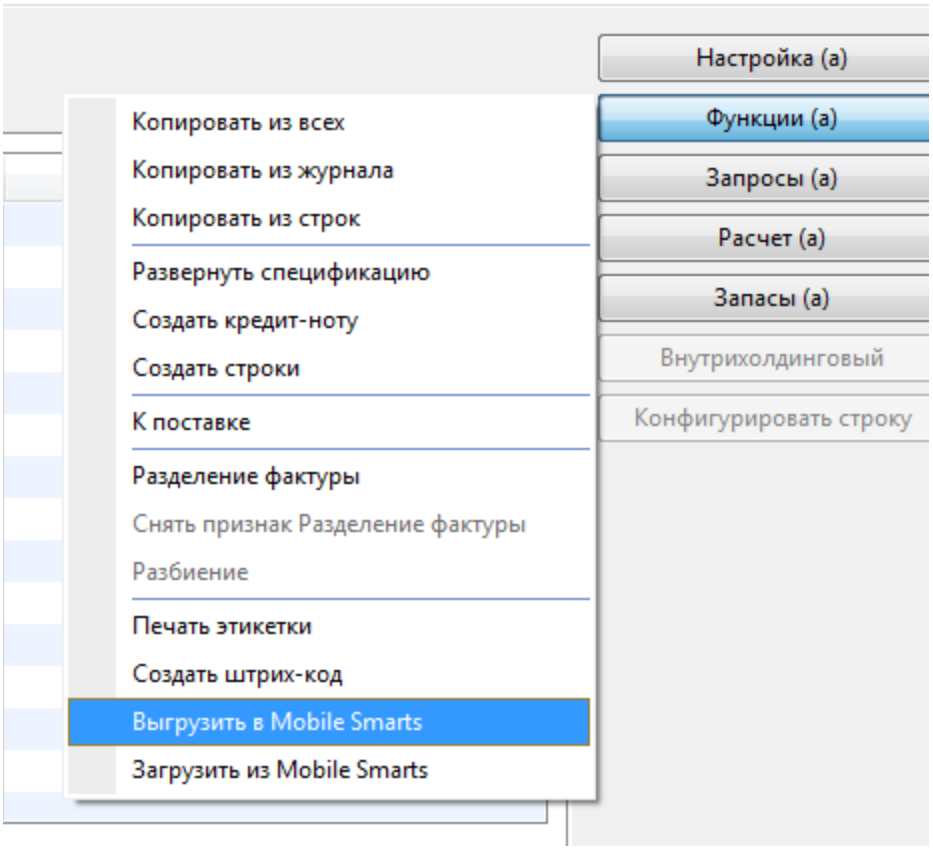

Рисунок 3.11 - Функции на строках закупки

Выйдет сообщение: «Выгрузка прошла успешно».

После необходимо передать данные на ТСД, для этого необходимо произвести синхронизацию ТСД с сервером.

На ТСД выбрать программу Mobile SMARTS, согласно рисунку 3.12.



Рисунок 3.12 - Меню на ТСД

На экране ТСД пользователю будет виден перечень доступных операций, необходимо выбрать Обмен с сервером в соответствии с рисунком 3.13.

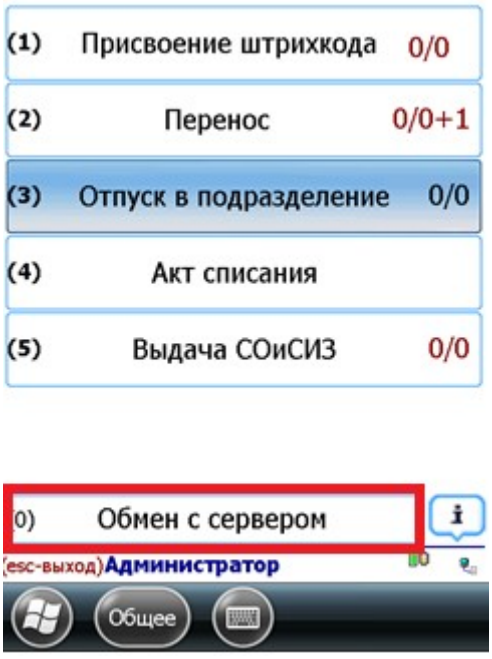

Рисунок 3.13 - Меню программы Mobile SMARTS на ТСД

После успешного обмена данными необходимо выбрать пункт Присвоение штрих-кода, затем выбрать закупку, из предложенных, как видно на рисунке 3.14.

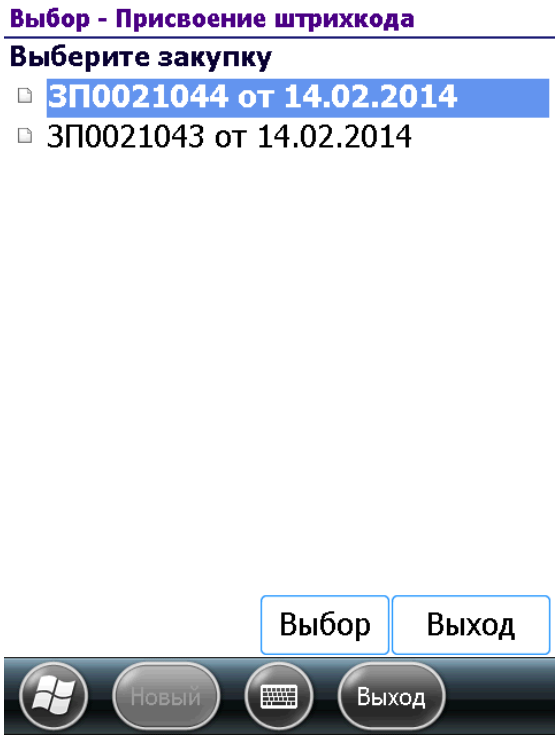

Рисунок 3.14 - Меню доступных (загруженных) закупок на ТСД

Откроется меню действий с закупкой, где доступно Сканирование, Просмотр отсканированных штрих-кодов, а так же:

- временно выйти - закупка остается не обработанной (при синхронизации ТСД с сервером, такие закупки остаются доступны на ТСД для дальнейшей работы);
- завершить - закупка перейдет в статус обработанной (при синхронизации ТСД с сервером, обработанные закупки выгружаются в ИУС П Axapta и удаляются из меню ТСД), согласно рисунку 3.15;

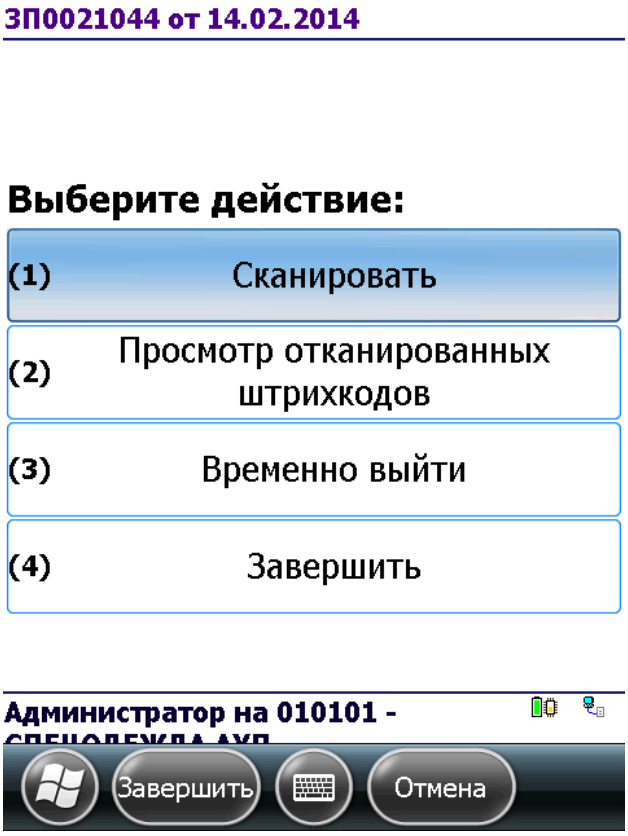

Рисунок 3.15 - Меню доступных операций для закупок на ТСД

- выбрать Сканировать, откроется список номенклатур из закупки согласно рисунка 3.16;

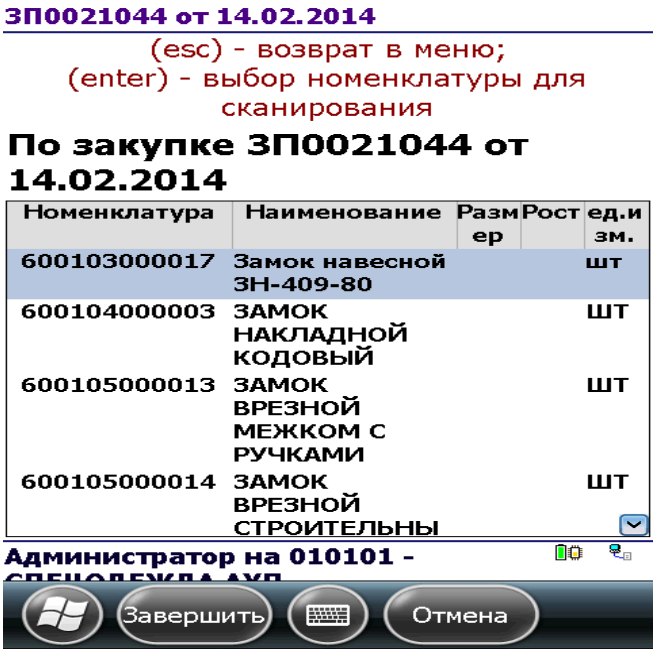

Рисунок 3.16 - Строки закупки на ТСД

- выбрать номенклатуру и нажать на клавиатуре ТСД кнопку ENT(Enter);
- появится приглашение Отсканировать штрих-код, как представлено на рисунке 3.17;

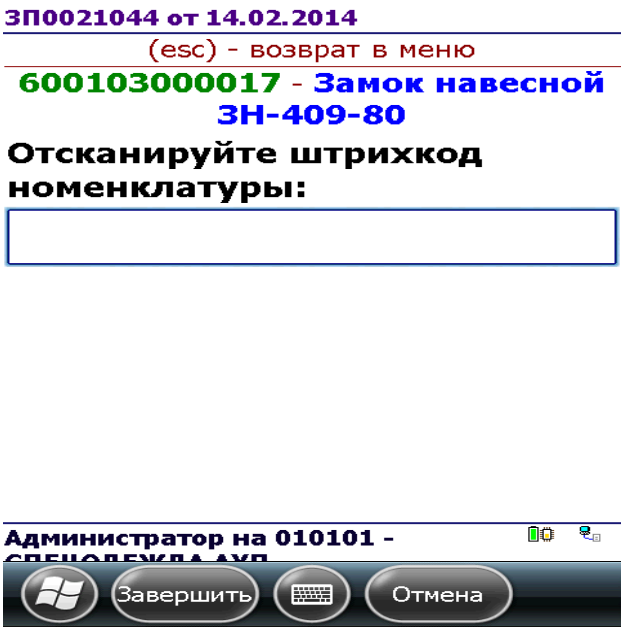

Рисунок 3.17 - Окно сканирования на ТСД

- необходимо луч ТСД навести на рисунок штрих-кода;
- на экране ТСД появится сообщение согласно рисунку 3.18;

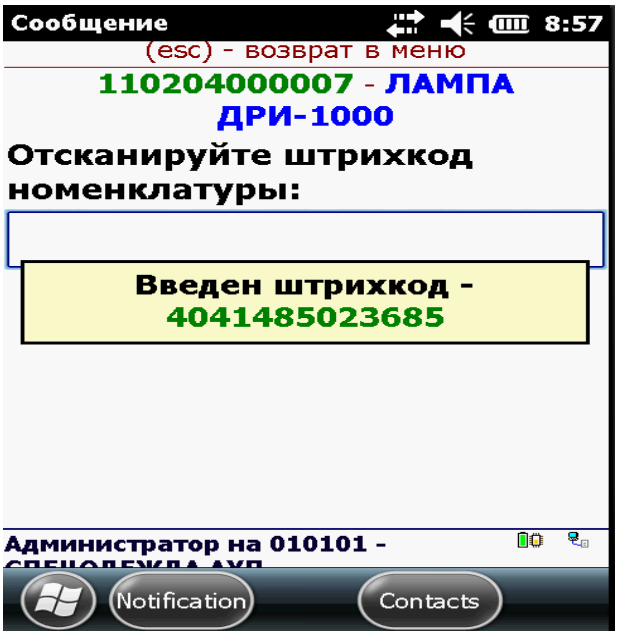

Рисунок 3.18 - Сообщение об успешном сканирование штрих-кода

- если работа с номенклатурой окончена, необходимо нажать ESC на клавиатуре ТСД. Система вернет в меню Действий с закупкой согласно рисунку 3.15, где доступен Просмотр отсканированных штрих-кодов.

Временно выйти - закупка остается не обработанной (при синхронизации ТСД с сервером, такие закупки остаются доступны на ТСД для дальнейшей работы).

Завершить - закупка перейдет в статус обработана (при синхронизации ТСД с сервером, обработанные закупки выгружаются в ИУС П Axapta и удаляются из меню ТСД).

- после того, как ко всем номенклатурам будут привязаны штрих-коды, данные можно передавать в ИУС П Axapta.

Для этого необходима Синхронизация с сервером Mobile SMARTS. (В результате после синхронизации из меню на ТСД удалятся все закупки, со статусом Завершено).

- далее в ИУС П Axapta из меню Расчеты с поставщиками\ Обычные формы\ Закупка выбрать закупку, находясь на строках выбрать пункт Функции\ Загрузить из Mobile SMARTS в соответствии с рисунком 3.19.

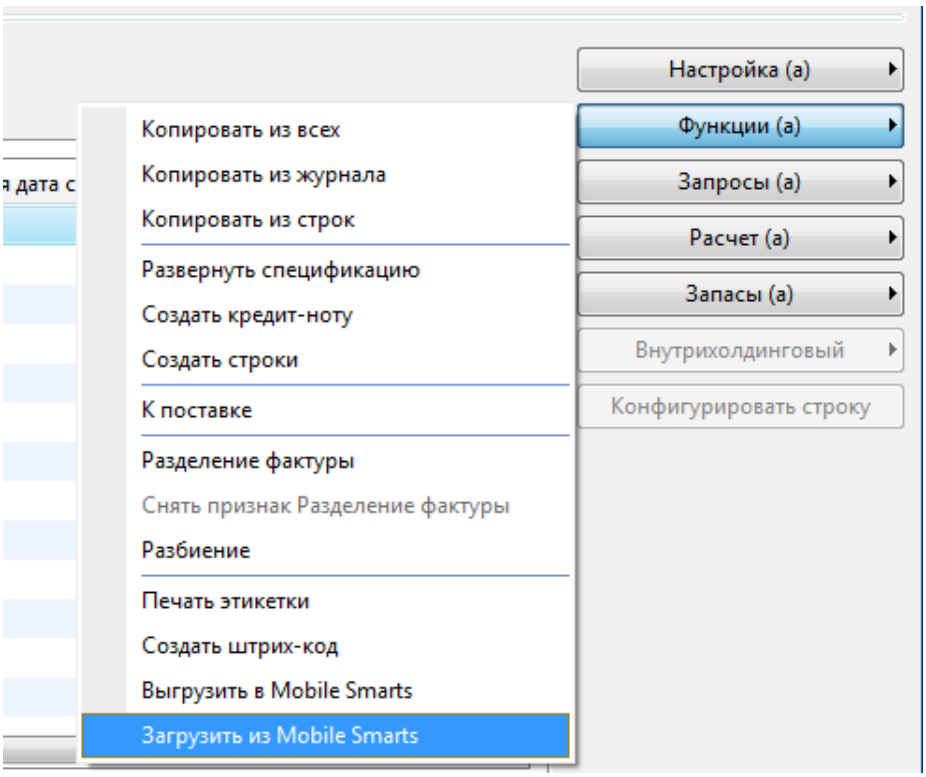

Рисунок 3.19 - Функции на строках закупки

- выйдет сообщение - Загрузка прошла успешно. Поле «Штрих-код» заполнится, как показано на рисунке 3.20.

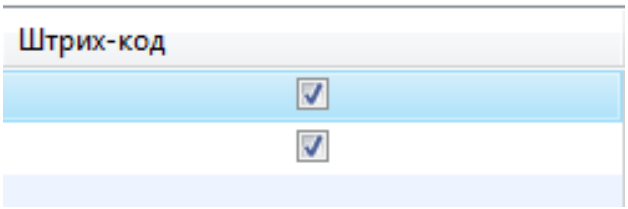

Рисунок 3.20 - Признак активности штрих-кода на строках закупки

Печать этикетки с штрих-кодом возможна двумя способами

В ИУС П Axapta из меню Расчеты с поставщиками\ Обычные формы\ Закупка выбрать закупку, выбрать пункт Функции\ Печать этикетки согласно рисунку 3.21.

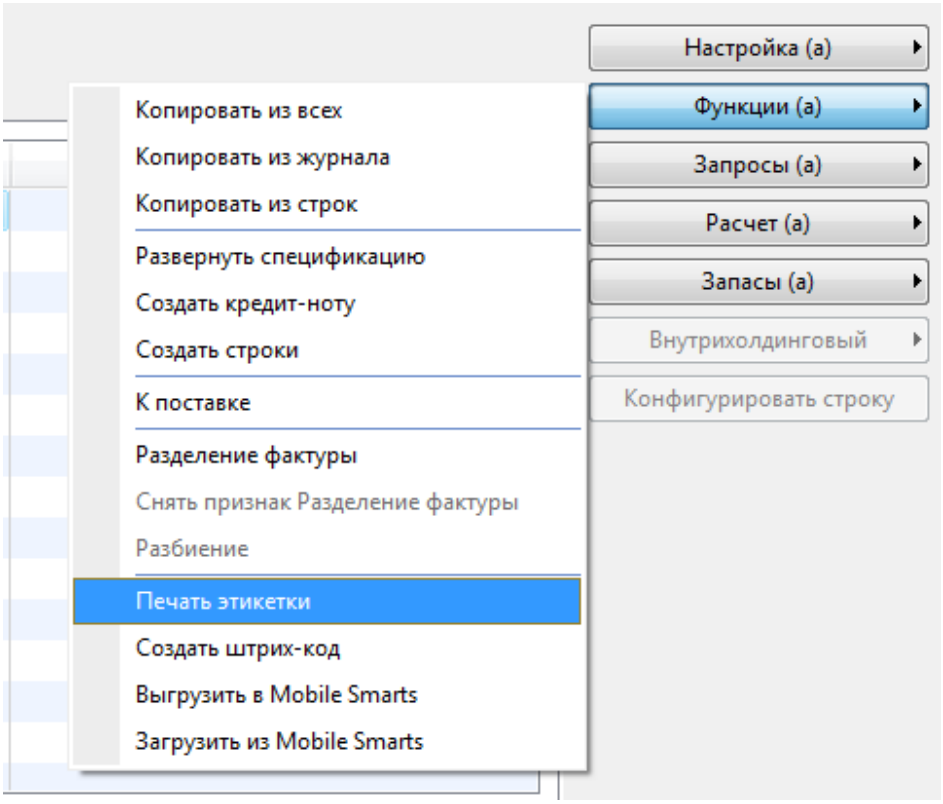

Рисунок 3.21 - Признак активности штрих-кода на строках закупки

Необходимо зайти в Управление запасами\ Обычные формы\ Сведения о номенклатуре\ Настройки\ Штрих-код, в соответствии с рисунком 3.22.

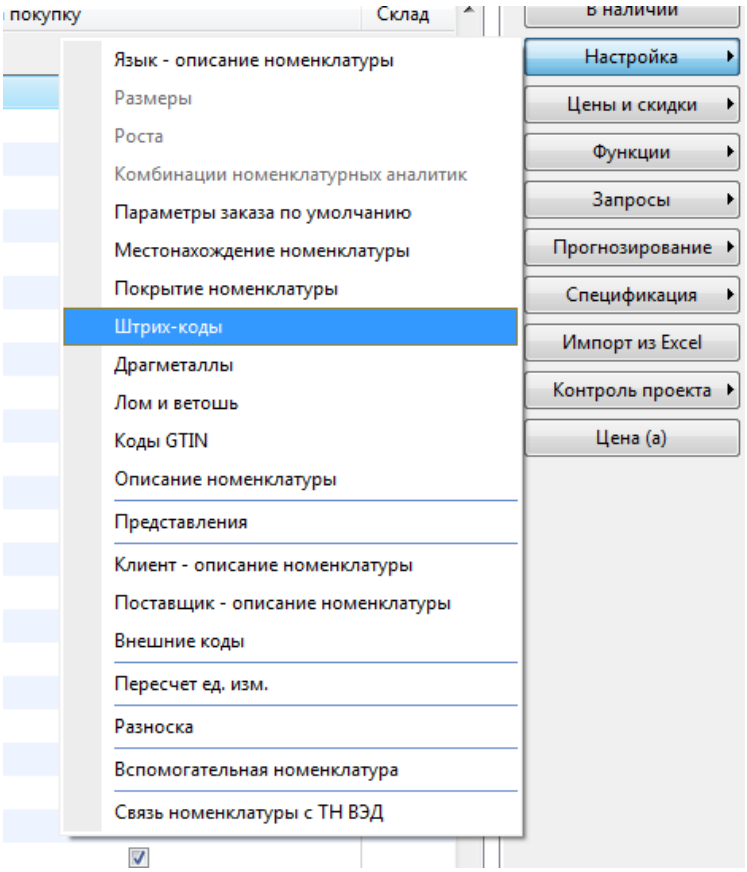

Рисунок 3.22 - Настройки в номенклатурном справочнике

Откроется форма для присвоения штрих-кода, где доступна кнопка Печать этикетки, как показано на рисунке 3.23.

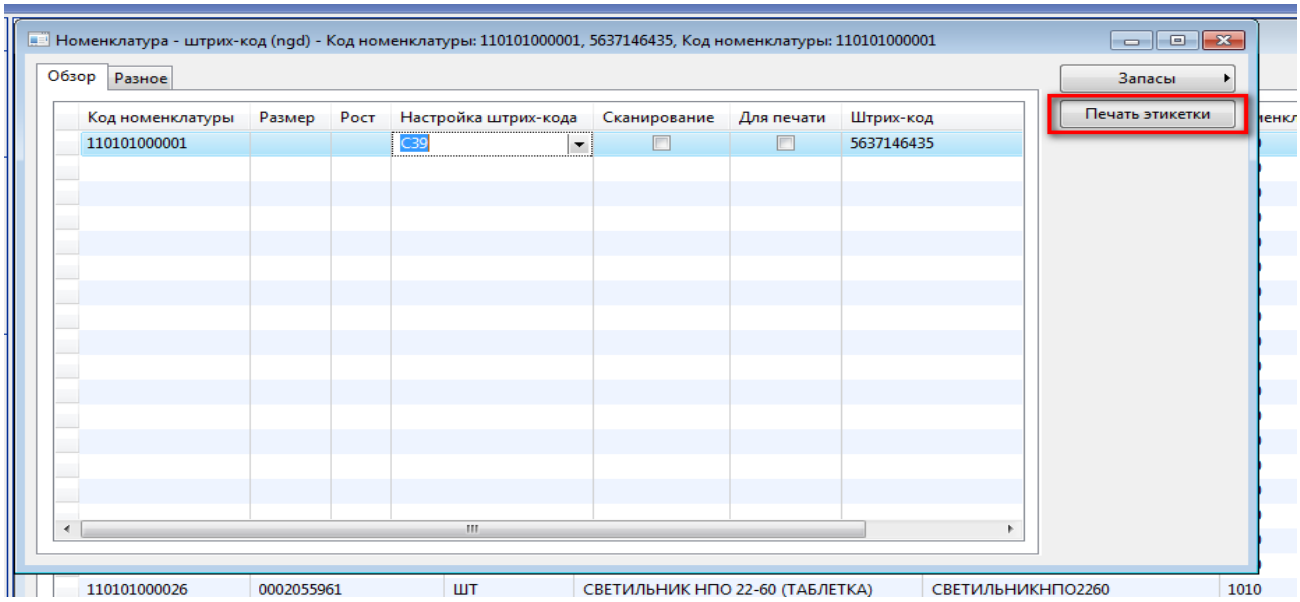

Рисунок 3.23 - Форма настройки штрих-кода

Формирование журналов на терминале сбора данных

Создание заголовка Журнала (общее, для всех журналов):

- на ТСД выбрать программу Mobile SMARTS, согласно рисунка 3.24;



Рисунок 3.24 - Меню на ТСД

- на экране ТСД пользователю будет виден список доступных для формирования Журналов, согласно рисунка 3.25;

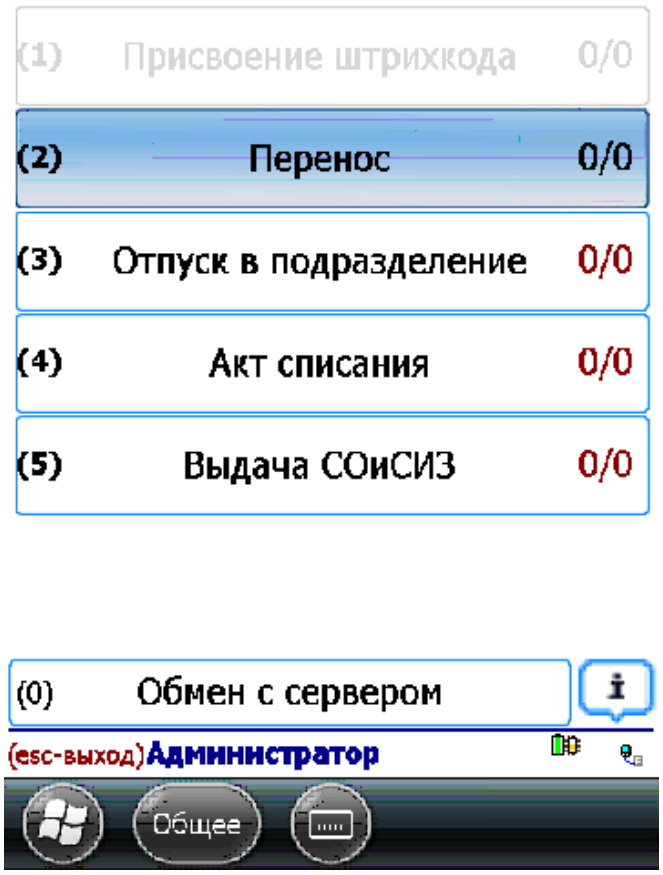

Рисунок 3.25 - Меню программы Mobile SMARTS на ТСД

- в меню на ТСД выбрать нужный вид журнала;
- открывшейся форме (после здесь будут отображаться созданные журналы) выбрать новый, в соответствии с рисунком 3.26;

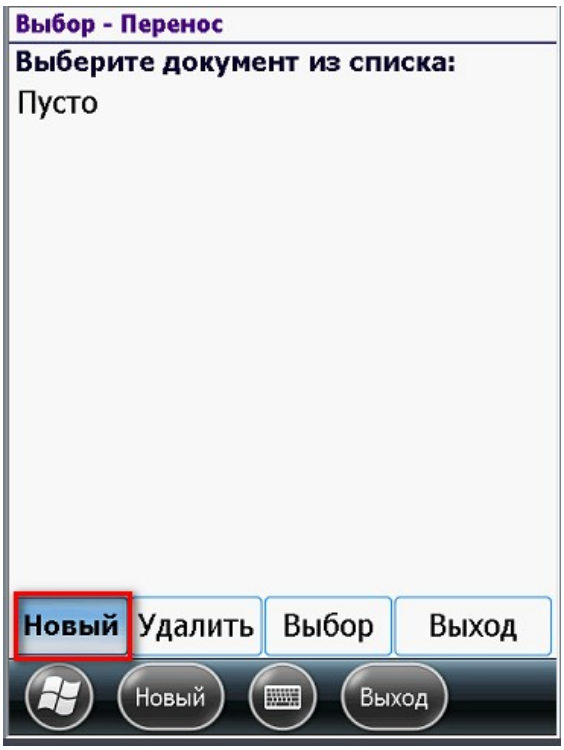

Рисунок 3.26 - Меню выбора журналов на ТСД

- система предложит ввести имя журнала (присваивается автоматически, при необходимости можно задать любое имя с помощью клавиатуры ТСД), далее рисунок 3.27;

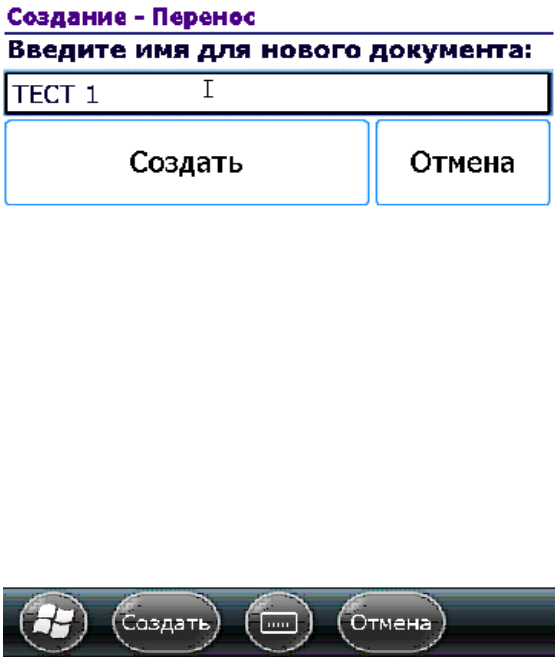

Рисунок 3.27 - Меню выбора журналов на ТСД

- нажать на экране Создать или ENT на клавиатуре ТСД;
- откроется меню Действий с журналом, где доступны Создание строк и Просмотр строк, а так же:
- временно выйти - журнал остается не обработан (при синхронизации ТСД с сервером, такие журналы остаются доступны на ТСД для дальнейшей работы);
- завершить - журнал перейдет в статус обработан (при синхронизации ТСД с сервером, обработанные журналы выгружаются в ИУС П Axapta и удаляются из меню ТСД), как показано на рисунке 3.28.

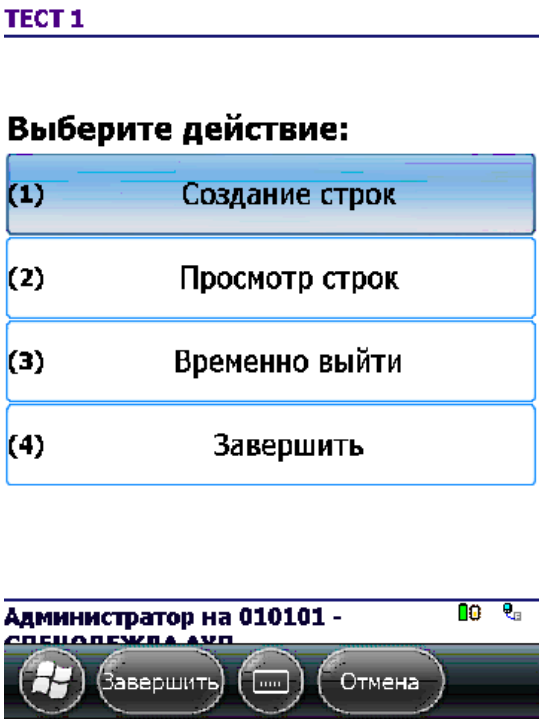

Рисунок 3.28 - Меню доступных операций для журналов на ТСД

Создание строк журналов (общее для журналов):

- на ТСД выбрать действие Создание строк. Необходимо выбрать ЦЗ, пользуясь прокруткой, согласно рисунку 3.29, и нажать ENT на клавиатуре ТСД;

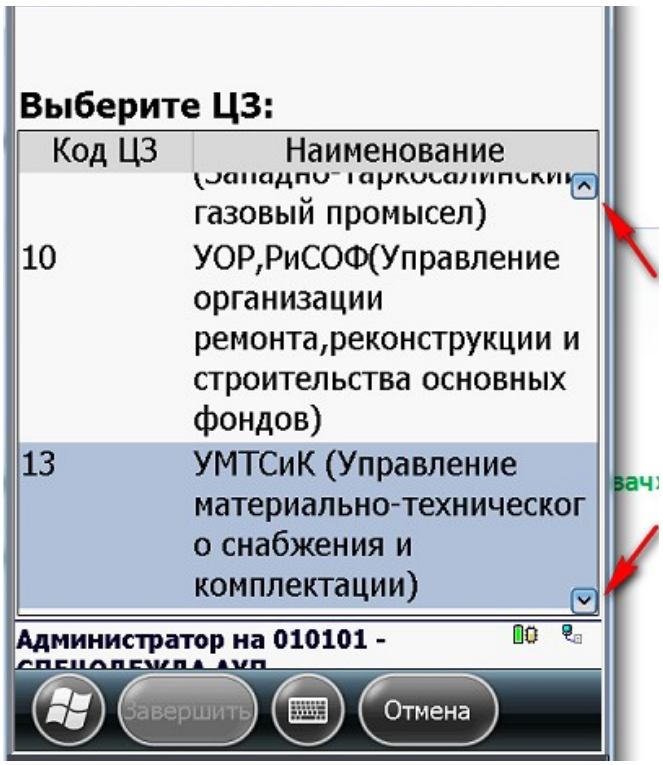

Рисунок 3.29 - Меню выбора ЦЗ на ТСД

- укажите Склад, пользуясь прокруткой ка на рисунке 3.30 или клавиатурой ТСД и нажмите ENT на клавиатуре ТСД.



Рисунок 3.30 - Меню выбора склада на ТСД

Журнал Перенос.

В следующем окне (пользуясь прокруткой, как на рисунке 3.29, или клавиатурой ТСД) и нажмите ENT на клавиатуре ТСД, необходимо указать Принимающий склад.

На экране ТСД отобразятся все склады, относящиеся к ЦЗ передающего склада в соответствиии с рисунком 3.31.



Рисунок 3.31 - Меню выбора склада на ТСД

Появится приглашение Отсканировать штрих-код.

Журнал Отпуск в подразделение.

На экране ТСД выбрать принимающий ЦЗ ка на рисунке 3.32.



Рисунок 3.32 - Меню выбора принимающего ЦЗ на ТСД

Затем выбрать принимающий склад, как на рисунке 3.33 и нажать ENT на клавиатуре ТСД.

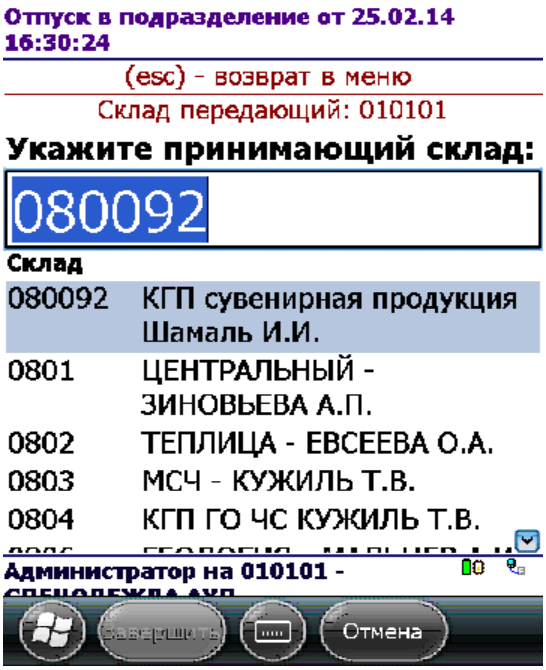

Рисунок 3.33 - Меню выбора принимающего склада на ТСД

Появится приглашение Отсканировать штрих-код.

Журнал Акт списания.

Необходимо указать МВЗ, как на рисунке 3.34, пользуясь прокруткой ,согласно рисунку 3.29 или клавиатурой ТСД.

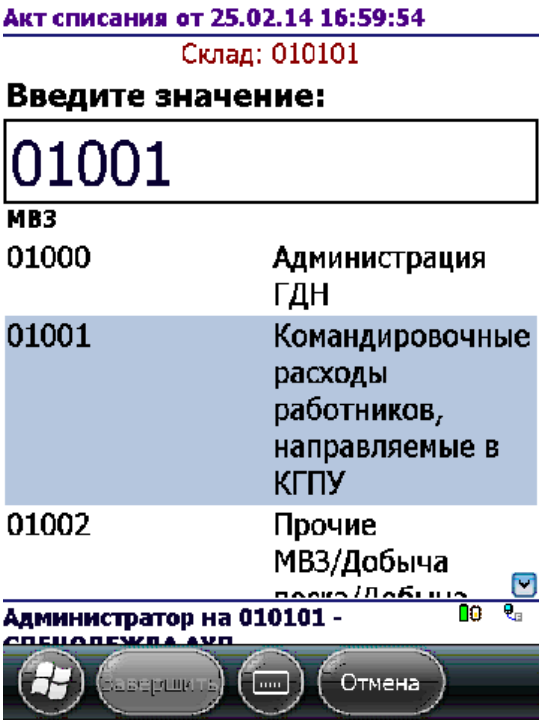

Рисунок 3.34 - Меню выбора МВЗ на ТСД

Появится приглашение Отсканировать штрих-код.

Журнал Выдача СиОС и Выдачи спецодежды.

Необходимо выбрать табельный номер сотрудника, на которого необходимо выдать СО или СиОС, как на рисунке 3.35, пользуясь прокруткой, согласно рисунку 3.29 или клавиатурой ТСД.

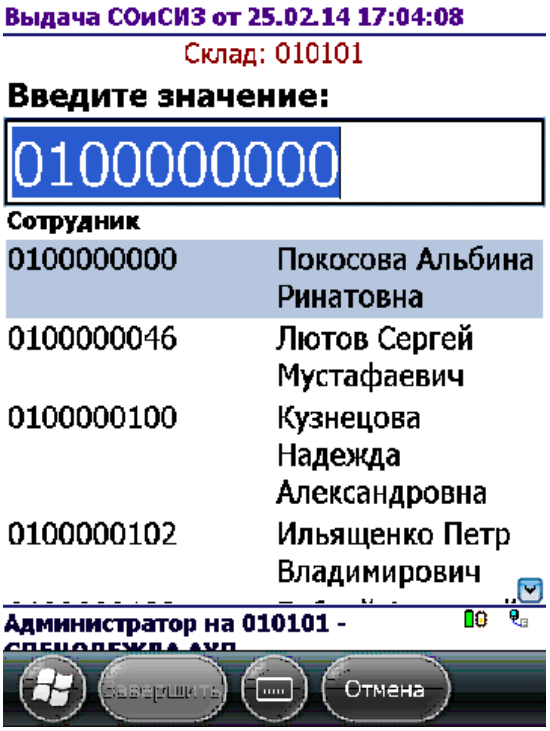

Рисунок 3.35 - Меню выбора табельного номера сотрудника на ТСД

Появится приглашение Отсканировать штрих-код.

Сканирование штрих-кода (общее для всех журналов)

Появится приглашение Отсканировать штрих-код как на рисунке 3.36

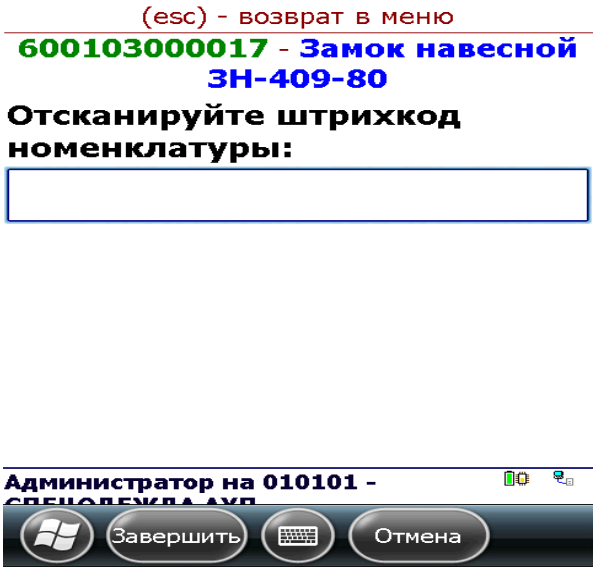

Рисунок 3.36 - Окно сканирования на ТСД

Необходимо луч ТСД навести на рисунок штрих-кода.

На экране ТСД появится сообщение в соответствии с рисунком 3.37.

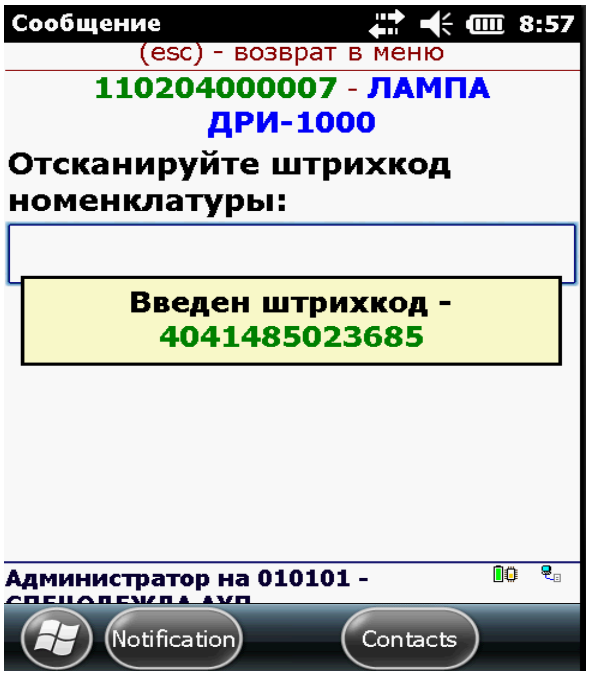

Рисунок 3.37 - Сообщение об успешном сканирование штрих-кода

Если работа с номенклатурой окончена, необходимо нажать ESC на клавиатуре ТСД.

Откроется меню Действий с закупкой, согласно рисунку 3.38, где доступны Создание строк и Просмотр строк, а так же:

- временно выйти - журнал остается в статусе «Не обработан» (при синхронизации ТСД с сервером, такие журналы остаются доступны на ТСД для дальнейшей работы).
- Завершить - журнал перейдет в статус «Обработан» (при синхронизации ТСД с сервером, обработанные журналы выгружаются в Аксапту и удаляются из меню ТСД).

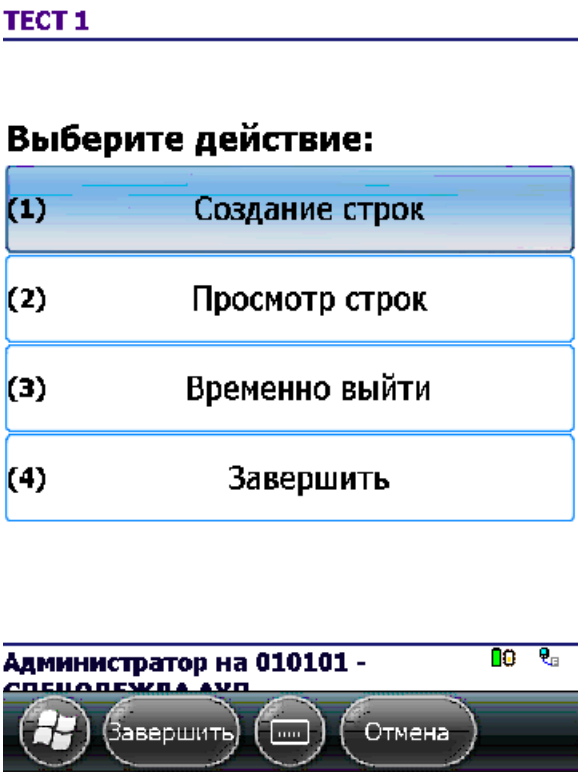

Рисунок 3.38 - Меню доступных операций для журнала на ТСД

Загрузка журналов в ИУСП Axapta с Mobile SMARTS

Необходима Синхронизация ТСД с сервером Mobile SMARTS (в результате после синхронизации из меню на ТСД удалятся все журналы, со статусом Завершено).

В ИУС П Axapta в модуле Управление запасами необходимо выбрать интересующий журнал, как на рисунке 3.39.

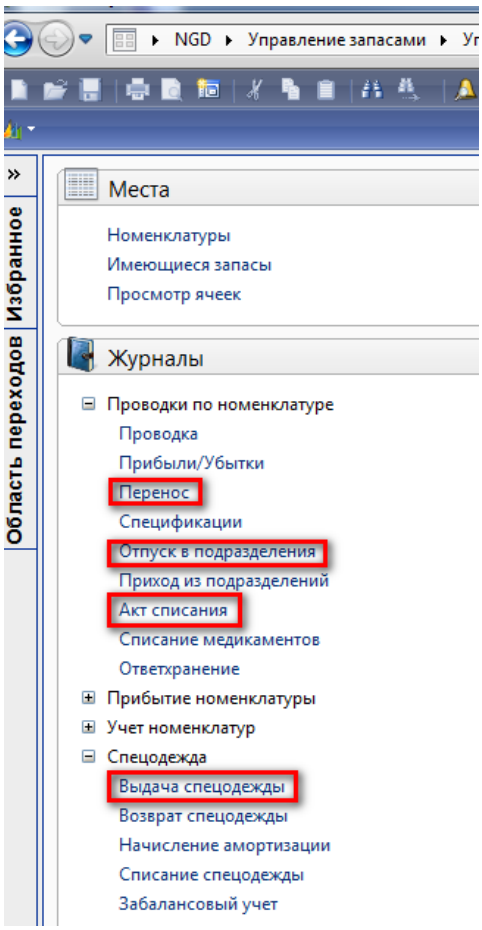

Рисунок 3.39 - Меню модуля Управление запасами в ИУСП Axapta

В меню журнала выбрать Функции\ Загрузка данных из Mobile Smarts, согласно рисунку 3.40.

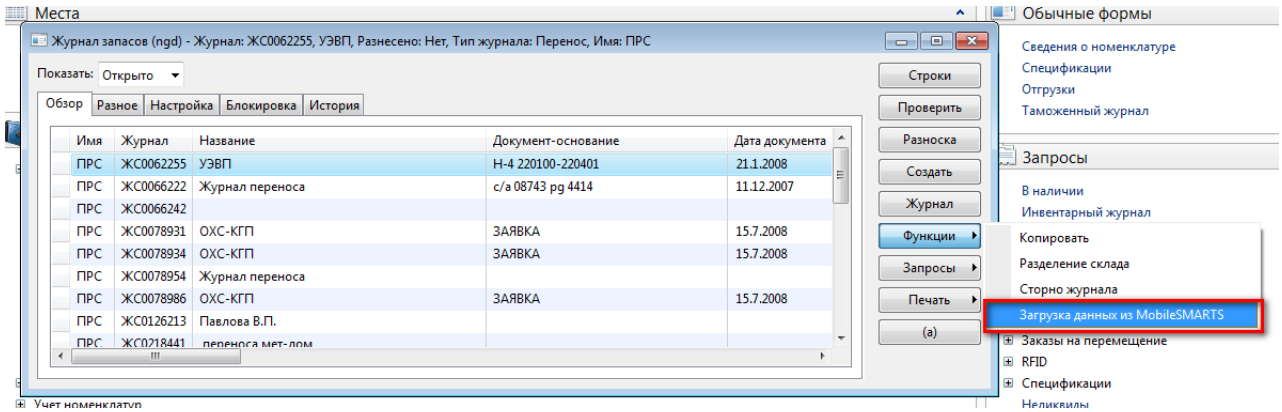

Рисунок 3.40 - Функции на строках журнала в ИУСП Axapta

В открывшейся форме необходимо нажать кнопку Синхронизировать с Mobile Smarts, как на рисунке 3.41.

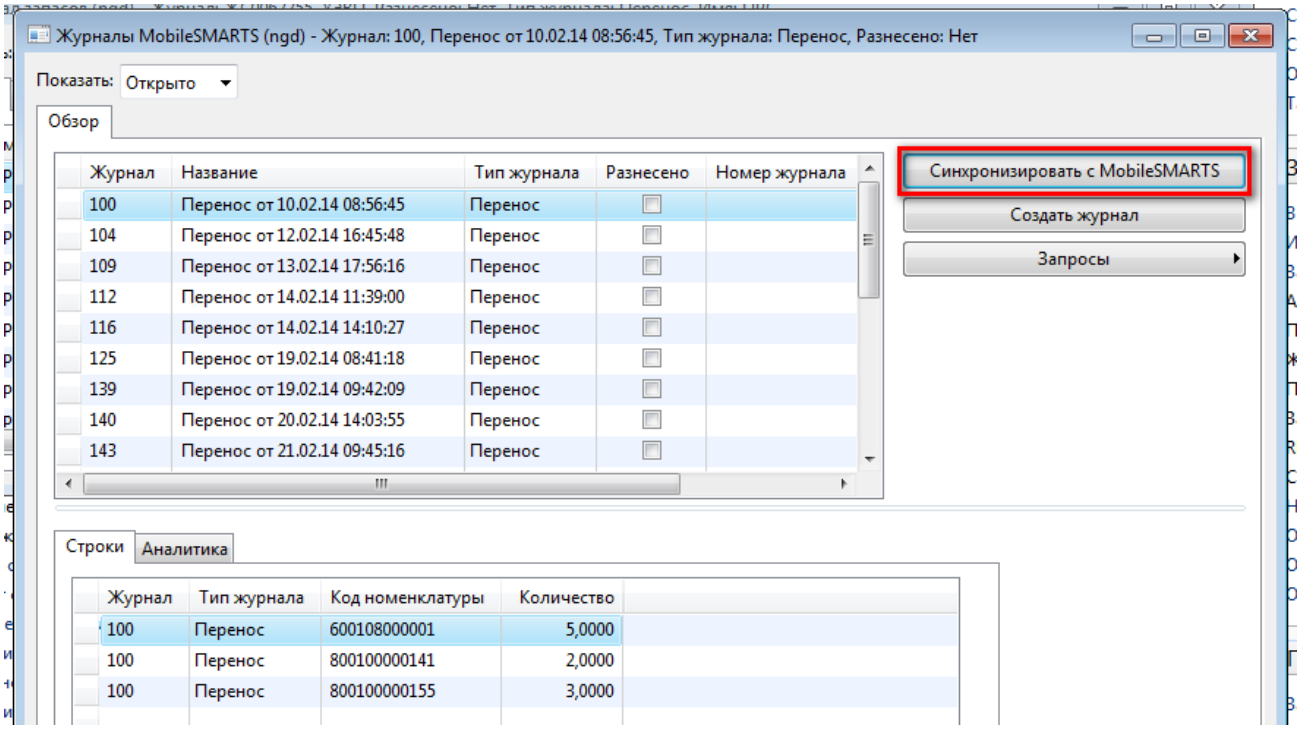

Рисунок 3.41 - Форма синхронизации ИУСП Axapta с Mobile SMARTS

В результате данные выгрузятся с сервера MS в ИУС П.

Необходимо выбрать интересующий журнал. В нижней части формы отображаются номенклатуры с аналитиками и количество по ним.

Нажать создать журнал и указать дату проводки, как на рисунке 3.42.

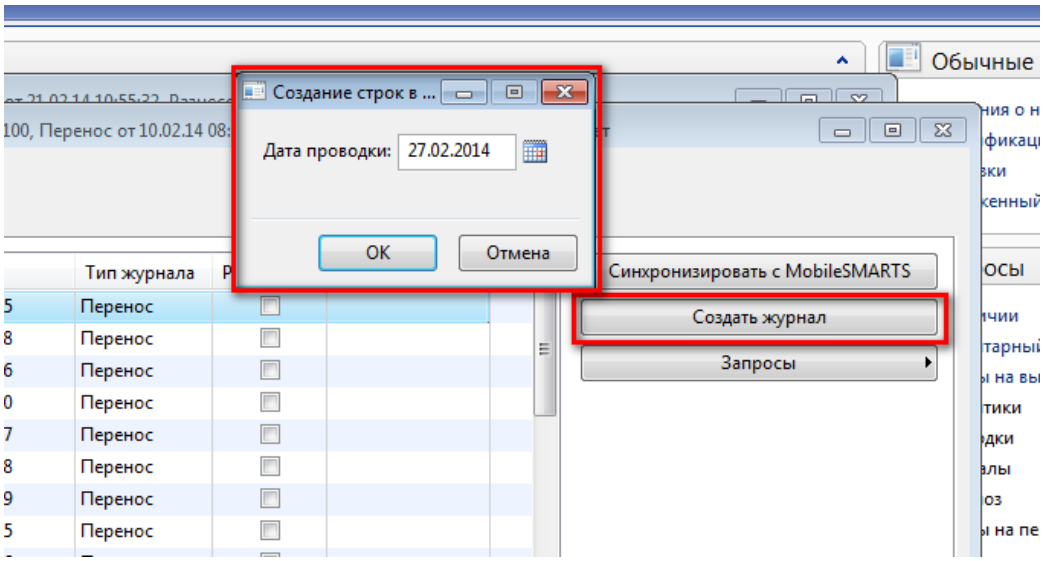

Рисунок 3.42 - Форма выбора даты в ИУСП Axapta

Система выдаст сообщение об успешной загрузке и в поле «Номер журнала» появится номер сформированный в ИУС П Axapta, согласно рисунку 3.43.

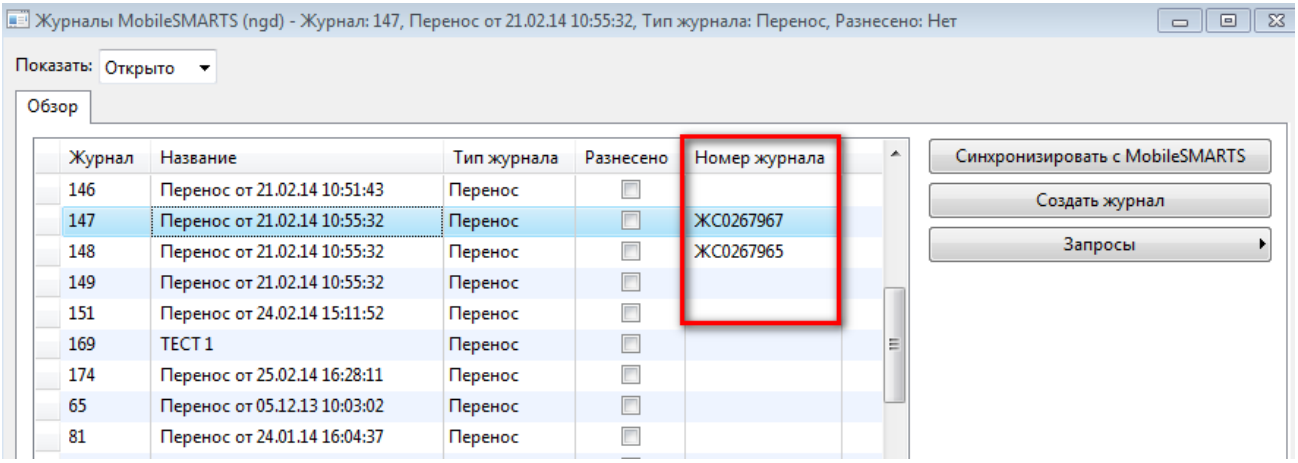

Рисунок 3.43 - Форма синхронизации ИУС П Axapta с Mobile SMARTS

ЗАКЛЮЧЕНИЕ

В результате внедрения проекта изменились процессы управления снабжением.

В процессе планирования МТР появилась дополнительная возможность проконтролировать заявленную потребность. К примеру, материал есть в наличии на складе и вновь попал в список заявок. В таких случаях с помощью ТСД на складе можно наглядно проверить предыдущую дату закупки материала, его заказчика и имеющееся на складе количество. В дальнейшем возможно принятие организационного решения.

В процессе поступления МТР, при наличии заводского штрих-кода, есть возможность сканировать его и использовать в дальнейшем. В случае отсутствия на них штрих-кода в учетной системе его можно сгенерировать, распечатать и наклеить на материал.

В ИУС П Axapta были выполнены доработки, позволяющие обеспечить связь между процессами формирования потребности в Заявочной кампании и поступления материалов в Общество.

После поступления МТР автоматически формируется разнарядка, которая учитывает период потребности, приоритет и предыдущую поставку. На основании нее формируются уведомления об отгрузке. Таким образом, система следит за процессом распределения материалов в подразделения и контролирует объём их отпуска.

Значительно изменился процесс комплектования МТР. Сейчас кладовщики, при помощи ТСД, выбирают необходимую операцию и указывают аналитику. После чего сканируют штрих-коды. Если номенклатура в разнарядке и на терминале совпали, указывается количество к отпуску.

После комплектации отгрузки, Терминал синхронизируется с сервером Mobile Smarts и данные загружаются в учетную систему. Автоматически формируется журналы и бухгалтерские проводки. Становится возможной печать документов. Использование штрих-кода позволило исключить ошибки, лишние операции и упорядочить процесс.

За счет применения штрих-кода оптимизирован процесс проведения инвентаризации. Сейчас подсчет МТР ведется при помощи ТСД. Все проектные работы проводились в соответствии с действующими стандартами. Разработка осуществлялась в среде системы MS Dynamics Ax, Mobile SMARTS. Поставленные цели и задачи достигнуты.

В ходе работы была разработана соответствующая техническая документация и инструкции. По результатам совещаний рабочей группы оформлялись протоколы.

Разработанная система удовлетворяет потребности руководства Общества и сотрудников Управления материально-технического обеспечения и комплектации.

ГП - газовый промысел

ГОиЧС – гражданской обороны и чрезвычайных ситуаций

ИУС П – информационно-управляющая система предприятия

ИУС П Axapta - информационно-управляющая система предприятия Axapta

КР – капитальный ремонт

КИПиА - контрольно измерительные приборы и автоматика

Mobile SMARTS - программная платформа

МТР – материально-технический ресурс

МВЗ – место возникновения затрат

НСИ – нормативно-справочная информация

ООО «Газпром добыча Ноябрьск» - общество с ограниченной ответственностью «Газпром добыча Ноябрьск»

ОАО «Газпром» - открытое акционерное общество «Газпром»

ОФИР – отдел фондов и реализации

ОНМ – оборудование не требующее монтажа

ПО – программное обеспечение

ПЭО – планово-экономический отдел

ПЭН - производственно-эксплуатационные нужды

РФ – Российская Федерация

СИУС – служба информационно-управляющих систем

СО, СиОС – спецодежда, смывающие и обезвреживающие средства

ТСД - терминал сбора данных

УМТС – управление материально-техническими ресурсами

ЦЗ – центр затрат

ШК - штрих-код

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

КНИГИ

Однотомные издания

1. Анализ в управлении финансовым состоянием коммерческой организации / Н.Н. Илышева, С.И. Крылов.- М.: Финансы и статистика; ИНФРА-М, 2012. С.240

2. Астринский Д., Наноян В. Экономический анализ финансового положения предприятия //Экономист. – 2012. - № 12. С. 55-57

3. Аюшева А.А. Финансы предприятия – Иркутск: Норма, 2014. С. 80

4. Баканов М. И. Теория экономического анализа / М. И. Баканов, А. Д. Шеремет. – М.: Финансы и статистика, 2012. С. 416

5. Бадмаева Д.Г. Платежеспособность коммерческой организации: финансовый анализ // Аудиторские ведомости. – 2011. - № 1. С.18-20

6. Басовский Л.Е. Теория экономического анализа: Учебное пособие. – М.: ИНФРА-М, 2009. С. 222

7. Бородина Е.И. Финансовый анализ в системе менеджмента организации// Экономический анализ: теория и практика. – 2009. -№22.С.12-16

8. Вендров, А. М. Проектирование программного обеспечения экономических информационных систем / А.М. Вендров. – М.: Финансы и статистика, 2013. С. 352

9. Смирнова, Г. Н. Проектирование экономических информационных систем / Г. Е. Смирнова, А. А. Сорокин, 30. Ф. Тельнов. - М.: Финансы и статистика, 2011.С. 512

10. Маклаков, С. В. BPwin и ERwin. CASE-средства разработки информационных систем. - М.: ДИАЛОГ-МИФИ, 2009. С. 256

11. Басовский, Л.Е. Финансовый менеджмент. Учебник. – М.: ИНФРА-М, 2012. С. 240

12. Конноли, Т., Бегг, К., Страчан А. Базы данных: проектирование, реализация, сопровождение. Пер. с англ. – М.: издательский дом «Вильямс», 2010. - С.1120

13. Бердникова Т.Б. Анализ и диагностика финансово-хозяйственной деятельности предприятия: Учебное пособие. – М.: Инфра –М, 2009. С. 215

14. Богомолов В.А., Богомолова А.В. Антикризисное регулирование экономики: Учеб. пособие / М.: ЮНИТИ, 2013. С. 160

15. Фаронов В.В., Шумаков П.В. Delphi. Руководство разработчика баз данных. – М.: «Нолидж», 2011. С. 640

16. Праг, Керри, Н., Ирвин, Майкл, Р. Access 2002. Библия пользователя.: Пер. с англ. – М.: Издательский дом «Вильямс», 2014. С. 1216

17. Жаров А. Железо IBM. – М: «МикроАрт», 2014. С. 336

18. Гавердовский А. Концепция построения систем автоматизации документооборота// Открытые системы. - 2009. - №1. С. 29 – 34

19. Афанасьев А. Частные реализации систем документооборота// Открытые системы. - 2010. - №1. С. 59 - 66

20. Конноли Т., Бегг К., Страчан А. Базы данных: проектирование, реализация, сопровождение. - 2009. С. 15 – 48

21. Бочаров В.В. Финансовый анализ: Учебное пособие. – СПб.: Питер, 2012. С. 240

22. Бочкарева Т.А. Анализ финансового состояния: Монография. – Хабаровск: ХГАЭП, 2013. С. 148

23. Бузырев В.В. Планирование на строительном предприятии: Учеб. пособие для студ. высш. учеб. заведений/ В.В. Бузырев, Ю. П. Панибратов, И.В. Федосеев. – М.: Академия, 2009. С. 435

Сборники официальных документов

24. САНПИН, 2.2.2/2.4.1340-03 «Гигиенические требования к персональным электронно-вычислительным машинам и организации работы».

Сборники научных статей

25. Сборник научных трудов. Вып.3. – Волгоград: РПК «Политехник», 2010. С. 58

Многотомные издания

26. Аникина И.Д. Корпоративное управление на российских предприятиях // Стратегия и тактика управления предприятием в переходной экономике.

27. Артеменко В.Г., Беллендир М.В. Финансовый анализ: Учебное пособие. – 2 изд., перераб. и доп. – М.: Дело и Сервис, 2009. С. 160

28. Балабанов И.Т. Финансовый анализ и планирование хозяйствующего субъекта. – 2-е изд.; доп. – М.: Финансы и статистика, 2009. С. 208

29. Абрютина М.С., Грачев А.В. Анализ финансово-экономической деятельности предприятия: учебно-практическое пособие. – 2-е изд.; испр. – М.: Дело и сервис, 2013. С. 256

ПРИЛОЖЕНИЕ А

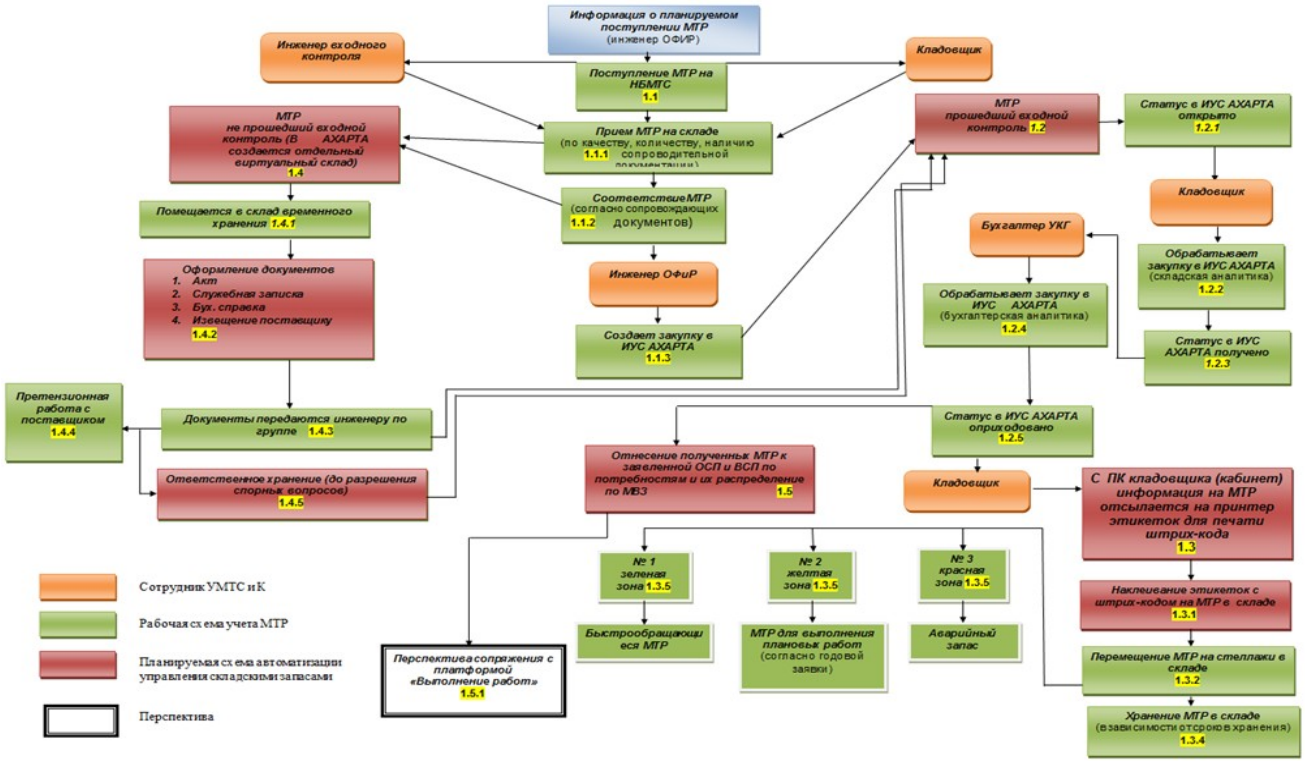

Рисунок А.1 - Обращение материальных потоков УМТСиК

ПРИЛОЖЕНИЕ Д

Таблица Д.1 - Сводная таблица сравнения программных продуктов

Функционал	**Mobile SMARTS**	**RDP или Citrix ICA (удаленный доступ к рабочему столу)**	**Веб-браузер**
Аппаратно не позволять пользователю ничего сканировать там, где сканирование не ожидается или запрещено	✔	✖	✖
Не позволять пользователю сканировать штрихкоды быстрее, чем их обрабатывает учетная система (чтобы не допускать потерянных сканирований)	✔	✖	✖
«Пулеметная» скорость сканирования и распознавания штрихкодов (с использованием локального справочника товаров)	✔	✖	?*
Звуковое оповещение при ошибках (привлечь внимание пользователя к экрану ТСД)	✔	✖	?**
Работа без стилуса (стилусы теряются, шариковые ручки портят экран устройства)	✔	?***	?****
Мобильная печать на Bluetooth принтеры	✔	✖	✔
Автоматическое подстраивание под разные размеры экрана	✔	✖	✔
Блокирование доступа пользователей к другим программам на ТСД (чтобы не играли, не ходили в интернет, не ломали настройки)	✔	только в платных клиентах	часто только в платных браузерах
Оповещения о скором разряде батареи	✔	✖	✖
Программистам не нужно учить ничего нового	✖	✔	✔
Администам не нужно учить ничего нового	?*****	✖	✖

* - локальный javascript с выгрузкой локальных баз mysql для Windows CE и т.п.

** - закачка и проигрывание звуковых файлов при помощи javascript, но в большинстве случаев в IE под Windows CE работать не будет.

*** - много часов отладки форм, чтобы вычистить все случаи потери фокуса ввода и кривого порядка перевода фокуса.

**** - много кода javascript и много часов отладки javascript, чтобы настроить правильную навигацию внутри страниц и между ними.

***** - администрирование Mobile SMARTS снимает с сисадминов задачу настройки терминалов сбора данных, т.к. установщик Mobile SMARTS сам настраивает устройство для автовосстановления и т.п., а клиент Mobile SMARTS при запуске сам настраивает сканер ТСД для оптимальной работы на складе. В то же время использование удаленного доступа или браузера заставит сисадминов погрузиться в изучение настроек сканера, сохранения и восстановления настроек сканера, изучение узлов системного реестра Windows CE, поиск способов закрыть пользователям доступ к играм и интернету и т.п.

ПРИЛОЖЕНИЕ Г

Архитектура базы данных

Таблица Г. 1 - Сущность RInventMTRJournalTable

№	Наименование	Описание	Тип	Значение	Ключ
1	JournalId	Идентификатор журнала	Int	Not null	Первичный
2	SystemBlocked	Блокировка журнала	NoYes	Not null	
3	Approved	Утверждено	NoYes	Not null	
4	JournalNameId	Идентификатор наименования журнала	String	Not null	
5	Description	Краткое описание проводки	String	Not null	
6	BlockUserId	Блокировка Журнала	NoYes	Not null	
7	JournalType	Тип журнала		Not null	

Таблица Г.2 – Сущность RInventMTRJournalTable

№	Наименование	Описание	Тип	Значение	Ключ
1	CostsPlace	МВЗ	Int	Not null	
2	CostsCenter	ЦЗ	Int	Not null	
3	ItemId	Номенклатурный номер	String	Not null	
4	InventSubGroupId	Подгруппа номенклатуры	Int	Not null	
5	ItemName	Наименование номенклатуры	String	Not null	
7	UnitID	Единица измерения	String	Not null	
8	JournalId	Идентификатор журнала	nvarchar	Not null	
9	LineNum	Номер строки	NoYes	Not null	

10	TransDate	Дата проводки	String	Not null	
11	PeriodCode	Период выполнения проводки	String	Not null	
12	PeriodDate	Дата проводки	NoYes	Not null	
13	CreationType	Тип создания	NoYes	Not null	
14	QtyMin	Количество минимальное	String	Not null	
15	QtyMax	Количество максимальное	String	Not null	
16	CostsPrice	Цена	NoYes	Not null	
17	Priority	Приоритет	NoYes	Not null	
18	CostsMin	Стоимость минимум	String	Not null	
19	CostsMax	Стоимость максимум	String	Not null	
20	QtyApproved	Количество утвержденное	NoYes	Not null	
21	CostsApproved	Сумма	NoYes	Not null	
22	QtyEmergency Stock	Аварийный запас	String	Not null	
23	QtyGOCHS	ГОиЧС	String	Not null	
24	QtyReserve	Резервный запас	NoYes	Not null	
25	CostsReserveAll	Стоимость , min с учетом общего резерва	NoYes	Not null	
26	CostsEmergencyStock	Стоимость аварийный запас	String	Not null	
27	CostsGOCHS	Стоимость ГОиЧС	String	Not null	
28	CostsAgencyCompensationMin	Стоимость, агентское вознаграждение	NoYes	Not null	

29	CostsTransportationMin	Стоимость, ТЗР	NoYes		
30	CostsQualityFactorMin	Стоимость, коэффициент качества	String		
31	CostsAgencyCompensationMax	Стоимость, агентское вознаграждение	String		
32	CostsTransportationMax	Стоимость, ТЗР	NoYes		
33	CostsQualityFactorMax	Стоимость, коэффициент качества	NoYes	Not null	
34	CostsAgencyCompensationGOCHS	Стоимость, агентское вознаграждение	String	Not null	
35	CostsTransportationGOCHS	Стоимость, ТЗР	String	Not null	
36	CostsQualityFactorGOCHS	Стоимость, коэффициент качества	NoYes	Not null	
37	CostsAgencyCompensationEmergencyStock	Стоимость , агентское вознаграждение	NoYes	Not null	
38	CostsTransportationEmergencyStock	Стоимость, ТЗР	String		
39	CostsQualityFactorEmergencyStock	Стоимость , коэффициент качества	String		
40	CostsAgencyCo	Стоимость агентское	Int	Not null	

	mpensationApproved	вознаграждения			
41	CostsTransportationApproved	Стоимость, ТЗР	Int	Not null	
42	CostsQualityFactorApproved	Стоимость, коэффициент качества	Int	Not null	
43	QtyRemains	Прогнозируемый остаток	string	Not null	
44	QtyPurch	К закупу	string	Not null	
45	CostsPurch	Сумма к закупу	Int		
46	CostsAgencyCompensationPurch	Стоимость, агентское вознаграждение	Int		
47	CostsTransportationPurch	Стоимость, ТЗР	Int	Not null	
48	CostsQualityFactorPurch	Стоимость, коэффициент качества	string	Not null	
49	QtyFreeRemains	Свободный остаток	string	Not null	
50	OtdelCode	Код отдела ГКИ	nvarchar	Not null	
51	TechnicParamMTR	Технические характеристики МТР	Int		
52	MarkMTR	Марка МТР	Int	Not null	
53	ProducerCode	Код производителя	string	Not null	
54	ProducerName	Наименование производителя	string	Not null	
55	modifiedDateTime	Дата модификации	nvarchar	Not null	

56	modifiedBy	Время модификации	Int	Not null	
57	createdDateTime	Дата создания	Int	Not null	
58	createdBy	Время создания	string	Not null	
59	dataAreaId	Код кампании	string	Not null	
60	recVersion	Версия	nvarchar	Not null	

Таблица Г.3 – Сущность RInventMTRDemandTrans

№	Наименование	Описание	Тип	Значение	Ключ
1	CostsCenterCode	Центр затрат	Int	Not null	
2	InventLocation	Код склада	Int	Not null	
3	ItemId	Номенклатурный номер	Int	Not null	
4	ItemName	Наименование номенклатуры	string	Not null	
5	UnitID	Единица измерения	string	Not null	
6	Qty	Остаток	nvarchar	Not null	
7	PeriodCode	Номер журнала	Int	Not null	
8	JournalId	Идентификатор журнала	Int	Not null	
9	LineNum	Номер строки	string	Not null	
10	TransDate	Дата проводки	string	Not null	
11	QtyReceipt	Планируемый период	nvarchar	Not null	
12	QtyIssue	Планируемый расход	Int	Not null	
13	NotUse	Признак нелеквидности	Int	Not null	
14	InventDimId	Номер складских аналитик	string	Not null	

15	QtyReserv	Резерв	string	Not null	
16	InventSubGroupId	Код группы	nvarchar	Not null	
17	OtdelCode	Код отдела ГКИ	string	Not null	
18	modifiedDateTime	Дата модификации	string	Not null	
19	modifiedBy	Время модификации	nvarchar	Not null	
20	createdDateTime	Дата создания	string	Not null	
21	createdBy	Время создания	string	Not null	
22	dataAreaId	Код кампании	nvarchar	Not null	
23	recVersion	Версия	string	Not null	
24	RecId	Идентификатор проводки	string	Not null	

Таблица Г.4 – Сущность RInventMTRInventRemainsTrans

№	Наименование	Описание	Тип	Значение	Ключ
1	ItemId	Номер номенклатуры	nvarchar		
2	MarkMTR	Марка МТР	Int		
3	TechnicParamMTR	Техническая характеристика МТР	Int		
4	CostPriceLast	Цена последней закупки	string		
5	CostPriceLastUpdate	Дата последнего обновления цены	string		
6	Deflator	Дефлятор	nvarchar		
7	DeflateCostPrice	Цена с учетом дефлятора	Int		

8	CalculationPrice	Цена в расчет	Int		
9	ApprovedGKPrice	Цена утвержденная в ГК 1 этап	string		
10	SpecificationsPrice	Цена спецификации	nvarchar		
11	JournalId	Идентификатор журнала	Int		
12	LineNum	Идентификатор строки	Int		
13	TransDate	Дата проводки	string		
14	PeriodCode	Номер журнала	string		
15	PeriodDate	Дата проводки	nvarchar		
16	CalculatedPrice	В расчет	Int		
17	CreationType	Тип создания	Int		
18	InventSubGroupId	Группа номенклатур	string		
19	ItemName	Наименование номенклатуры	nvarchar		
20	UnitID	Единица измерения	Int		
21	ReceiptDateLast	Дата последнего прихода	Int		
22	ApprovedGKPrice20	Цена утвержденная в ГК 2 этап	string		
23	dataAreaId	Код кампании	string		
24	recVersion	Версия	nvarchar		

25	RecId	Идентификатор проводки	Int		

Таблица Г.5 – Сущность RInventMTRCorrectPriceTrans

№	Наименование	Описание	Тип	Значение	Ключ
1	ItemGroupId	Номенклатурная группа	nvarchar	Not null	
2	ItemId	Номенклатурный номер	Int	Not null	
3	ItemName	Наименование номенклатуры	Int	Not null	
4	ItemType	Тип номенклатуры	string	Not null	
5	PurchModel	Последняя цена покупки	string	Not null	
6	Height	Количество товарооборота	nvarchar	Not null	
7	Width	Ширина	Int	Not null	
8	SalesModel	Цена продажи	Int		
9	CostGroupId	Группа надбавок	string		
10	ReqGroupId	Группа покрытия	nvarchar	Not null	
11	EPCManager	Диспетчер EPS	Int	Not null	
12	PrimaryVendorId	Поставщик	Int	Not null	
13	NetWeight	Вес нетто	string	Not null	
14	Depth	Единица измерения для аналитики	string	Not null	
15	UnitVolume	Объём	nvarchar	Not null	
16	BOMUnitId	Единица измерения	Int	Not null	

		спецификации			
17	Density	Плотность	Int	Not null	
18	Dimension	ЦЗ	string		
19	Dimension2	Цель	nvarchar	Not null	
20	Dimension3	МВЗ	Int	Not null	
21	Dimension4	Ст. Затрат	Int	Not null	
22	Dimension5	Эл. Затрат	string	Not null	
23	Dimension6	Обьем	string		
24	Dimension7	Контрагент	nvarchar	Not null	
25	Dimension8	Стройка	Int	Not null	
26	Dimension9	Эл. Затрат2	Int	Not null	
27	Dimension10	Ст. Затрат2	string	Not null	
28	CostModel	Последняя себестоимость	nvarchar		
29	UseAltItemId	Использовать	Int		
30	AltItemId	Альтернативный код номенклатуры	Int	Not null	
31	PBAItemAutoGenerated	Сгенерированный автоматически	string	Not null	
32	WMSArrivalHandlingTime	Время обработки прибытия	string	Not null	
33	StopExplode	Остановка развертывания	nvarchar	Not null	
34	Phantom	Искуственный	Int		
35	BOMLevel	Уровень	Int		
36	BatchNumGroupId	Стоимость агентское вознаграждения	string	Not null	
37	AutoReportFinished	Стоимость, ТЗР	nvarchar	Not null	

38	ABCTieUp	Стоимость, коэффициент качества	Int	Not null	
39	ABCRevenue	Прогнозируемый остаток	Int	Not null	
40	ABCValue	К закупу	string		
41	ABCContributionMargin	Сумма к закупу	string		
42	SalesPercentMarkup	Стоимость, агентское вознаграждение	nvarchar	Not null	
43	SalesContributionRatio	Стоимость, ТЗР	Int	Not null	
44	SalesPriceModelBasic	Стоимость, коэффициент качества	Int	Not null	
45	NameAlias	Свободный остаток	string	Not null	
46	grossDepth	Код отдела ГКИ	nvarchar		
47	grossWidth	Технические характеристики МТР	Int	Not null	
48	grossHeight	Марка МТР	Int	Not null	
49	sortCode	Код производителя	string	Not null	
50	SerialNumGroupId	Наименование производителя	string	Not null	
51	DimGroupId	Дата модификации	nvarchar		
52	ModelGroupId	Время модификации	Int		
53	ItemBuyerGroupId	Дата создания	Int		
54	TaxPackagingQty	Время создания	string		

55	TaraWeight	Стоимость агентское вознаграждения	nvarchar		
56	ScrapVar	Стоимость, ТЗР	Int		
57	ScrapConst	Код склада	Int		
58	StandardInventColorId	Номенклатурный номер	string		
59	StandardInventSizeId	Наименование номенклатуры	string	Not null	
60	ItemDimCombinationAutoCreate	Единица измерения	nvarchar	Not null	
61	ItemDimCostPrice	Остаток	Int	Not null	
62	ItemIdCompany	Номер журнала	Int	Not null	
63	AltInventSizeId	Идентификатор журнала	string	Not null	
64	AltInventColorId	Номер строки	nvarchar	Not null	
65	BOMCalcGroupId	Дата проводки	Int	Not null	
66	MarkupCode_RU	Планируемый период	Int	Not null	
67	Packing_RU	Планируемый расход	string	Not null	
68	AssetGroupId_RU	Признак нелеквидности	string	Not null	
69	AssetId_RU	Номер складских аналитик	nvarchar	Not null	
70	IntrastatWeight_CZ	Код склада	Int	Not null	
71	SubGroupId	Номенклатурный номер	Int	Not null	
72	OKOF	Наименование номенклатуры	string	Not null	

73	TehParms	Единица измерения	nvarchar	Not null	
74	ItemCategory	Остаток	Int	Not null	
75	OverallsCategoryId	Номер журнала	Int	Not null	
76	PhoneService	Идентификатор журнала	string	Not null	
77	UseDashItemId	Номер строки	string	Not null	
78	UseDashAddress	Дата проводки	nvarchar	Not null	
79	ROffBalanceMBPMark	Планируемый период	Int	Not null	
80	RInventMTR	Планируемый расход	Int	Not null	
81	RInventMTROtdelCode	Признак нелеквидности	string	Not null	
82	RInventMTRTimeItem	Номер складских аналитик	nvarchar	Not null	
83	RInventMTRItemCodeGK	Код номенклатуры ГК	Int	Not null	
84	modifiedDateTime	Дата модификации	Int	Not null	
85	dEL_ModifiedTime	Время модификации	string	Not null	
86	modifiedTransactionId	Дата создания	string	Not null	
87	createdDateTime	Время создания	nvarchar	Not null	
88	dEL_CreatedTime	Код кампании	Int	Not null	
89	createdBy	Версия	Int	Not null	
90	dataAreaId	Дата модификации	string	Not null	
91	recVersion	Время модификации	data	Not null	
92	RecId	Дата создания	data	Not null	

Таблица Г.6 – Сущность InventTable

№	Наименование	Описание	Тип	Значение	Ключ
1	SubGroupId	Код подгруппы	string	Not null	
2	Name	Наименование подгруппы	nvarchar	Not null	
3	GroupId	Код подгруппы	Int	Not null	
4	InventSubGroupType	Тип подгруппы	Int	Not null	
5	ROffBalanceMBPMark	Забаланс ИХП	string	Not null	
6	ROffBalanceMBPUsefulDate	Срок полезного использования	string	Not null	
7	ROffBalanceAccTireMark	Признак АКШ	nvarchar	Not null	
8	God_0	Дефлятор	Int	Not null	
9	God_1	Дефлятор	Int	Not null	
10	God_2	Дефлятор	string	Not null	
11	God_3	Дефлятор	data	Not null	
12	God_4	Дефлятор	data	Not null	
13	AgencyCompensation	Агентское вознаграждение	string	Not null	
14	CostTransportatio	Транспортные	nvarchar	Not	

	n	расходы		null	
15	QualityFactor	Коэффициент качества	Int	Not null	
16	RInventMTR	Включить в заявочную кампанию	Int	Not null	
17	RInventMTROtdelCode	Код отдела ГКИ	string	Not null	
18	NoCheckOfCreate	Не проверять при создании номенклатуры	string	Not null	
19	dataAreaId	Дата модификации	nvarchar	Not null	
20	recVersion	Время модификации	Int	Not null	
21	RecId	Дата создания	Int	Not null	

Таблица Г.7 – Сущность InventSubGroup

№	Наименование	Описание	Тип	Значение	Ключ
1	Code	Код центра затрат	string	Not null	
2	Name	Наименование ЦЗ	nvarchar	Not null	
3	Blocked	Влокированно	Int	Not null	
4	modifiedDateTime	Дата модификации	Int	Not null	
5	dEL_ModifiedTi	Время	string	Not	

	me	модификации		null	
6	modifiedBy	Дата создания	string	Not null	
7	createdDateTime	Время создания	nvarchar	Not null	
8	dEL_CreatedTime	Код кампании	Int	Not null	
9	createdBy	Версия	Int	Not null	
10	dataAreaId	Дата модификации	string	Not null	
11	recVersion	Время модификации	data	Not null	
12	RecId	Дата создания	data	Not null	

Таблица Г.8 – Сущность CostsPlace

№	Наименование	Описание	Тип	Значение	Ключ
1	Code	Код МВЗ	string	Not null	
2	Dimension	Центр затрат	nvarchar	Not null	
3	Dimension2	Цель	Int	Not null	
4	Dimension3	МВЗ	Int	Not null	
5	Dimension4	Статья затрат	string	Not null	
6	Dimension5	Элемент затрат	string	Not null	
7	Dimension6	Объект	nvarchar	Not null	
8	Dimension7	Контрагент	Int	Not null	
9	Dimension8	Стройка	Int	Not null	
10	Dimension9	Элемент затрат2	string	Not null	
11	Dimension10	Статья затрат2	data	Not null	

12	LedgerAccount	Счет ГК	data	Not null	
13	Name	Наименование	string	Not null	
14	CostsItemCode	Код статьи затрат	nvarchar	Not null	
15	CostsProcessCode	Код процесса	Int	Not null	
16	CostsElementCode	Код элемента затрат	Int	Not null	
17	UnitID	Единица измерения	string	Not null	
18	DimensionNum	Номер	string	Not null	
19	Blocked	Блокировано	nvarchar	Not null	
20	ParentCode	Родительский код	Int	Not null	
21	CostsPurposeCode	Код процесса	Int	Not null	
22	CostsPlaceTypeCode	Код вида места возникновения затрат	string	Not null	
23	CostsCenterCode	Код центра затрат	data	Not null	
24	ProcessName	Наименование	data	Not null	
25	CostsN	Очередь	string	Not null	
26	ParentCostsCenterCode_RU	Код родительского центра затрат	nvarchar	Not null	
27	Form121LineCode	Код строки формы	Int	Not null	
28	CostsPlaceGroupCode	Код группы МВЗ	Int	Not null	
29	VaxtaPoselok	Вахтовый поселок	string	Not null	

30	DepartamentsCode	Код курирующего отдела	string	Not null	
31	DimensionGroup	Аналитика для распределения затрат НУ	nvarchar	Not null	
32	DimensionGroup11	Центр затрат	Int	Not null	
33	DimensionGroup12	Цель	Int	Not null	
34	DimensionGroup13	МВЗ	string	Not null	
35	DimensionGroup14	Статья затрат	data	Not null	
36	DimensionGroup15	Элемент затрат	data	Not null	
37	DimensionGroup16	Объект	string	Not null	
38	DimensionGroup17	Контрагент	nvarchar	Not null	
39	DimensionGroup18	Стройка	Int	Not null	
40	DimensionGroup19	Элемент затрат2	Int	Not null	
41	DimensionGroup17	Статья затрат2	string	Not null	
42	DimensionGroup18	Обьект2	string	Not null	
43	DimensionGroup19	Контрагент2	nvarchar	Not null	
44	CostsProductionTypeCode	Код вида продукции	Int	Not null	
45	BlockingConsumer	Блокировать в качестве поставщика услуг	Int	Not null	
46	BlockingRecipient	Блокировать в качестве потребителя услуг	string	Not null	
47	DateEnd	Дата окончания	data	Not null	

48	DateBegin	Дата начала	data	Not null	
49	modifiedDateTime	Дата модификации	string	Not null	
50	dEL_ModifiedTime	Время модификации	nvarchar	Not null	
51	modifiedBy	Дата создания	Int	Not null	
52	createdDateTime	Время создания	Int	Not null	
53	dEL_CreatedTime	Код кампании	string	Not null	
54	createdBy	Версия	string	Not null	
55	dataAreaId	Дата модификации	nvarchar	Not null	
56	recVersion	Время модификации	Int	Not null	

ПРИЛОЖЕНИЕ Б

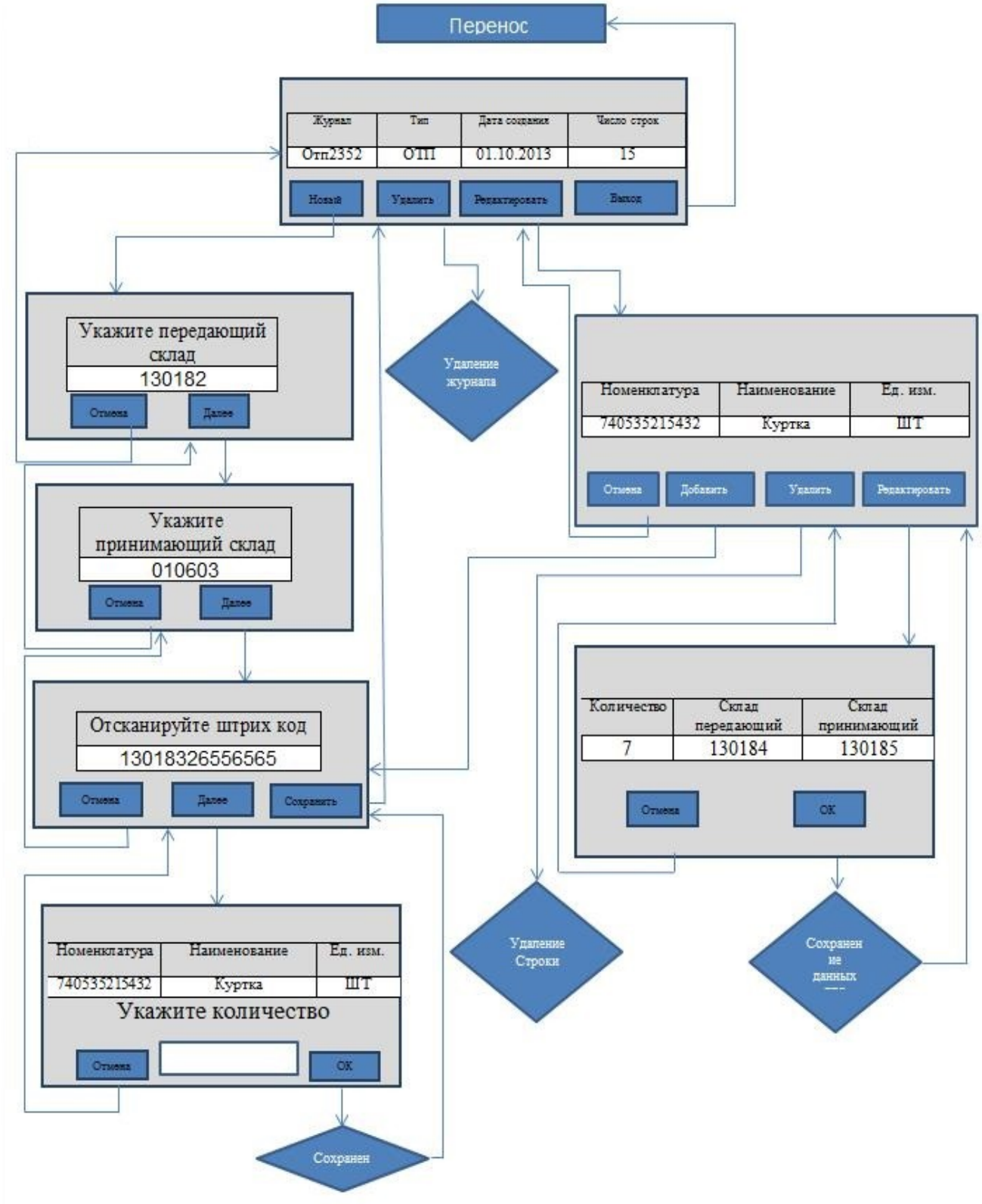

Рисунок Б.1 - Функция Перенос

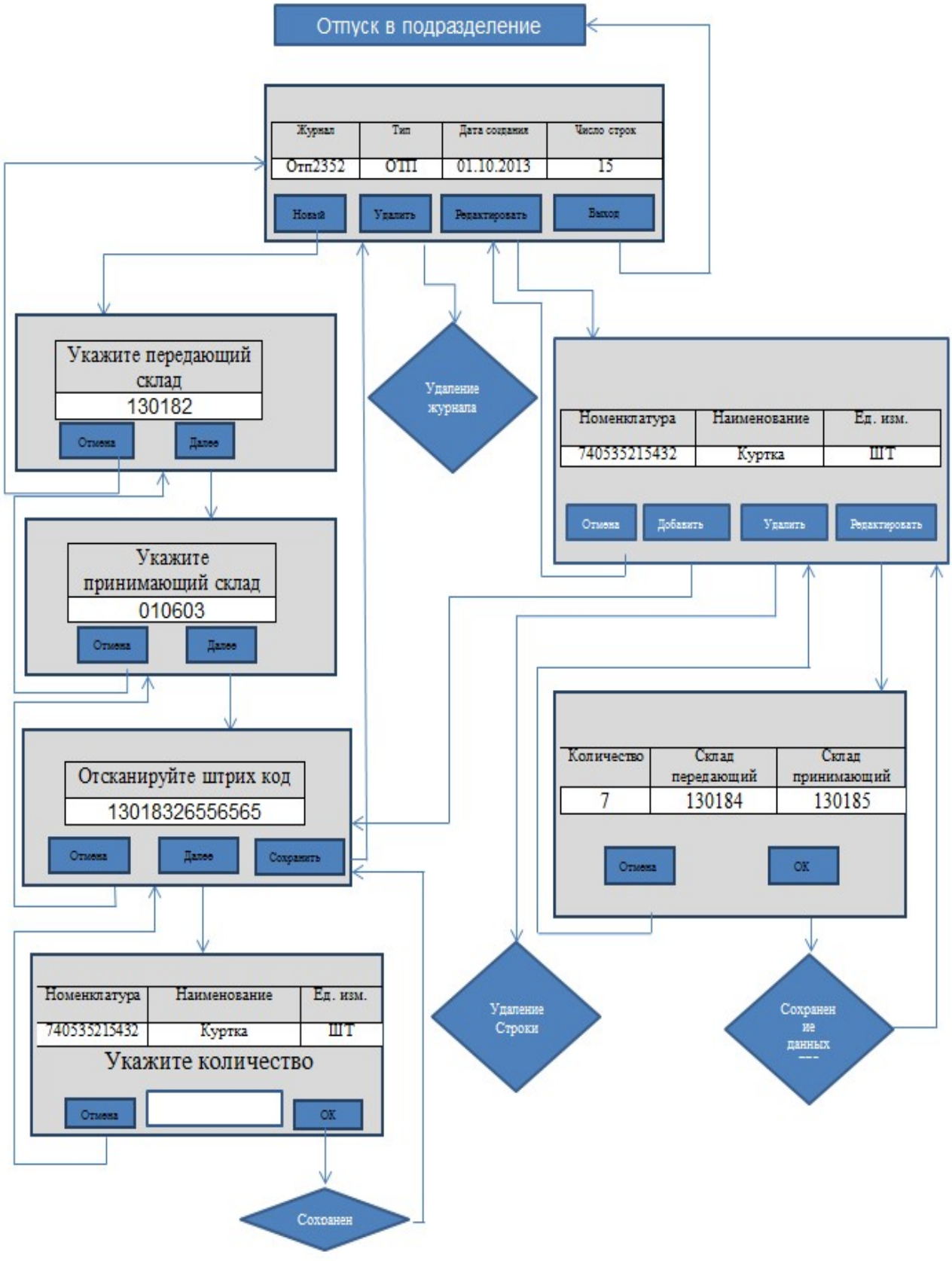

Рисунок Б.2 - Функция Отпуск в подразделени

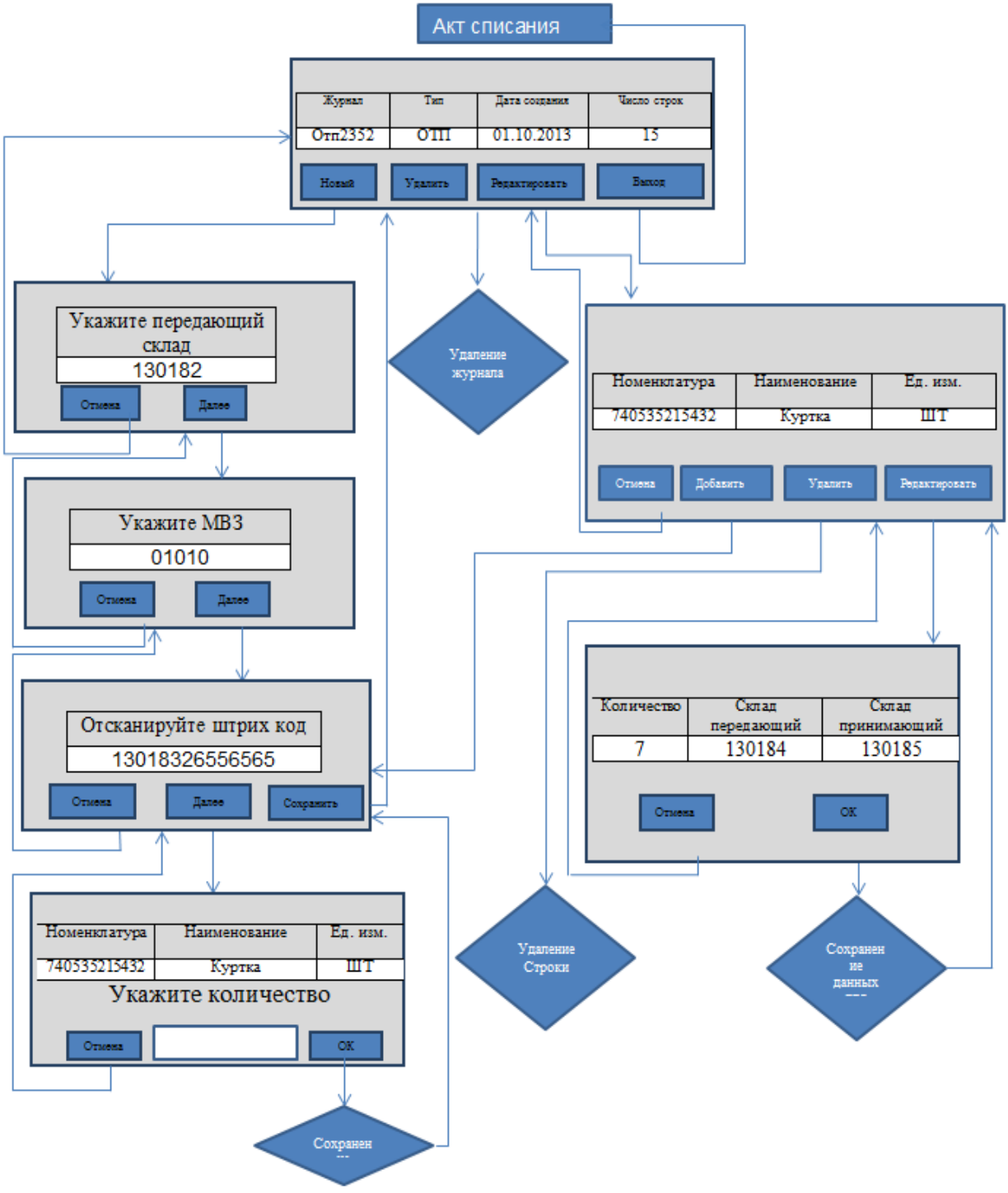

Рисунок Б.3 - Функция Акт списания

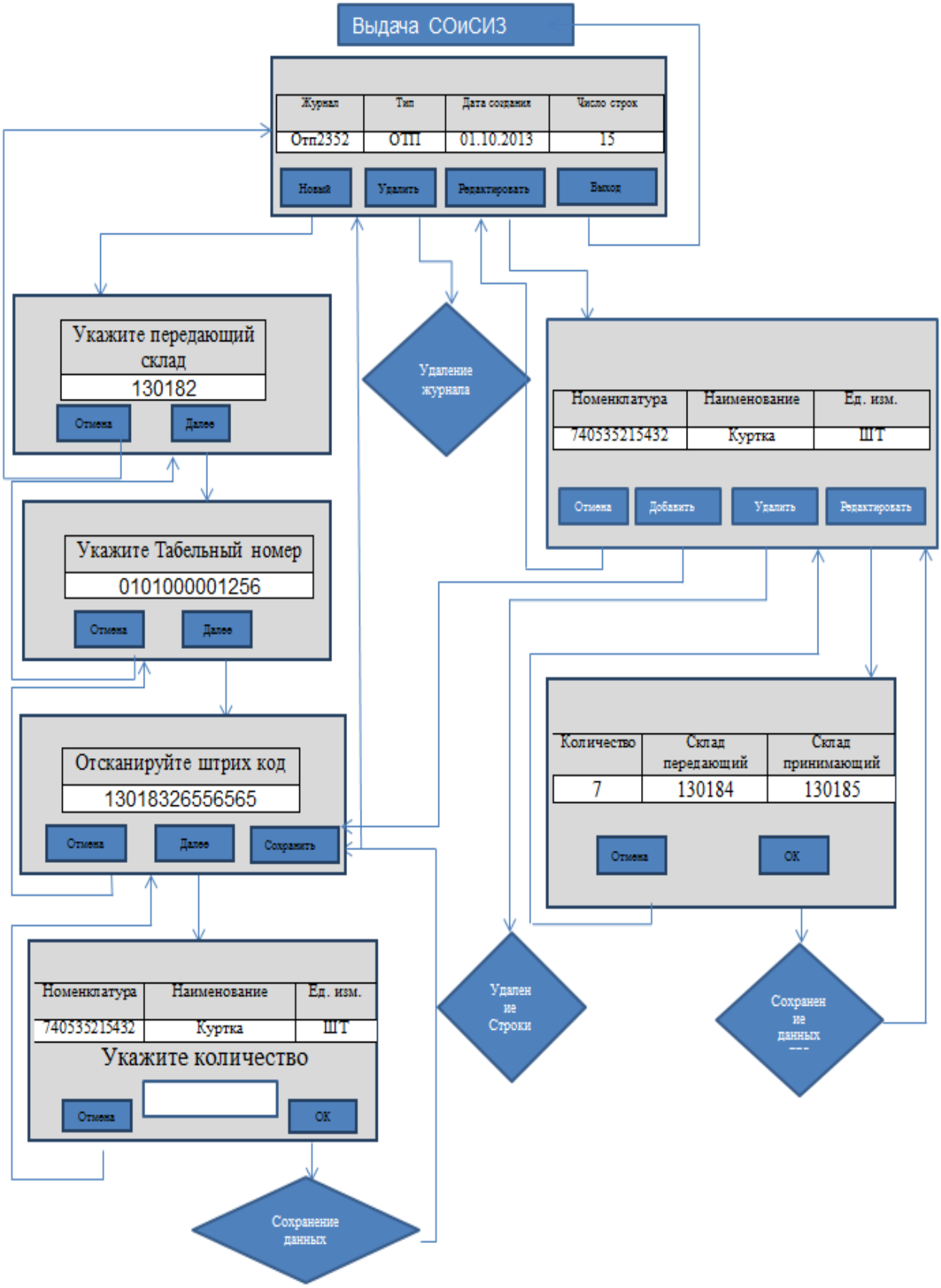

Рисунок Б.4 - Функция Выдача СОиСИЗ

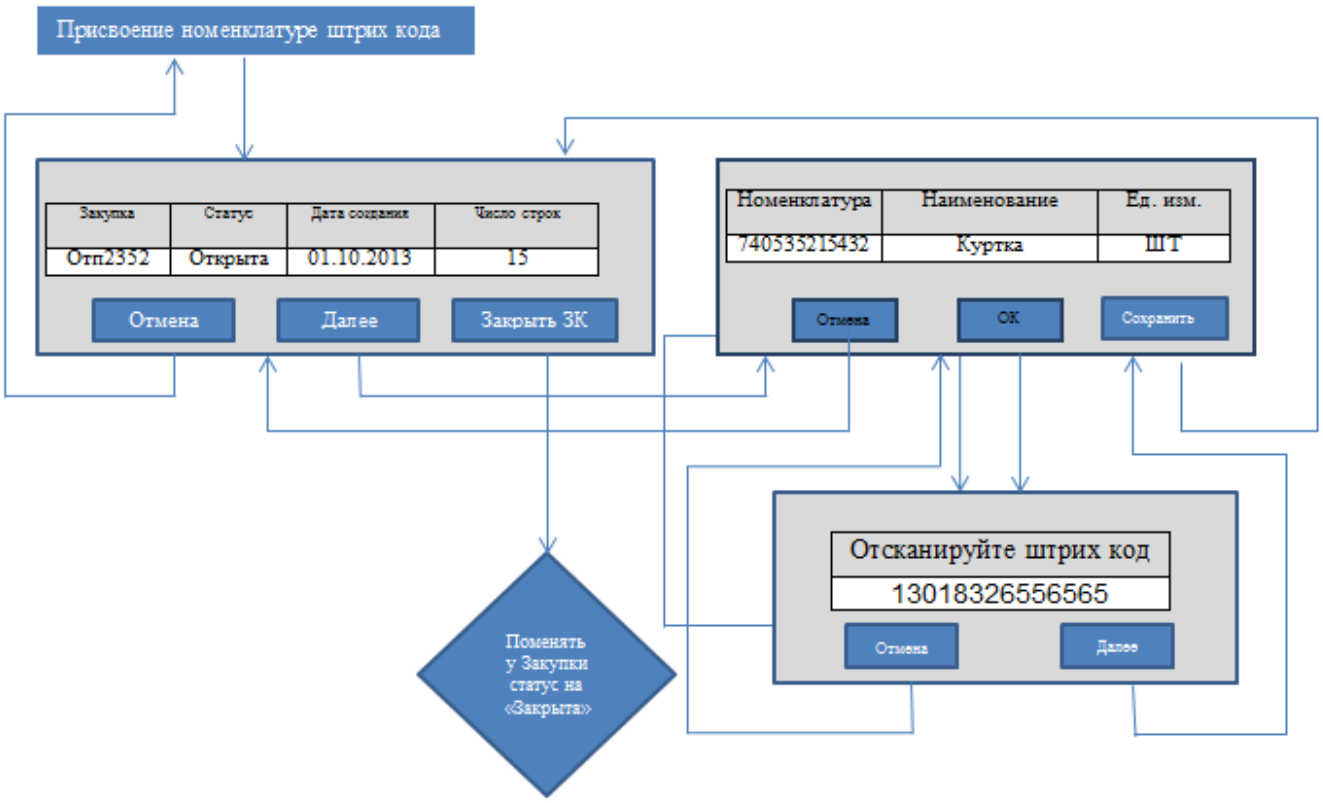

Рисунок Б.5 - Присвоение номенклатуре штрих кода завода-изготовителя

ПРИЛОЖЕНИЕ В

Листинг программы

```
JournalTableData            journalTableData;
        JournalTransData            journalTransData;
        RInventMTRInventRemainsTrans  rInventMTRInventRemainsTrans;
        InventTable                 inventTable;
        InventLocation              inventLocation;
        InventSum                   inventSum;
      //   RInventMTRCostsPlace         rInventMTRCostsPlace;

        Query                       query;
        QueryRun                    queryRun;
        QueryBuildDataSource        qbdsInventLocation;
        QueryBuildDataSource        qbdsInventDim;
        QueryBuildDataSource        qbdsInventSum;
        QueryBuildDataSource        qbdsInventTable;

        RecordSortedList            recordSortedList = new
RecordSortedList(tableNum(RInventMTRInventRemainsTrans));
        boolean                     loop;

        Qty                         postedQty;
        CostsPlaceGroupCode         costsCenterCode = '!';
        CostsPlaceGroupCode         costsCenterNew;

        Int                 totalCount;
        int                 counter;
        ;

        query = new Query();

        // InventLocation
        qbdsInventLocation = query.addDataSource(tableNum(InventLocation));

        qbdsInventLocation.addRange(fieldNum(InventLocation,
RInventMTRLocationType)).value(queryValue(RInventMTRLocationType::Central
CostsCenter));
        qbdsInventLocation.addRange(fieldNum(InventLocation,
RInventMTRLocationType)).value(queryValue(RInventMTRLocationType::CostsCe
nter));
```

```
        //   qbdsInventLocation.addRange(fieldNum(InventLocation,
RInventMTRLocationType)).value(queryValue(RInventMTRLocationType::Emerge
ncyStock));
        //   qbdsInventLocation.addRange(fieldNum(InventLocation,
RInventMTRLocationType)).value(queryValue(RInventMTRLocationType::GOCHS
));
        //////   qbdsInventLocation.addRange(fieldNum(InventLocation,
DimCenterIdSubordinated)).value(queryValue('35'));
            qbdsInventLocation.addRange(fieldNum(InventLocation,
UsedLocation)).value(QueryValue(NoYes::No));
            qbdsInventLocation.addRange(fieldNum(InventLocation,
Blocked)).value(QueryValue(NoYes::No));
            qbdsInventLocation.addSortField(fieldNum(InventLocation, DimCenterId),
SortOrder::Ascending);
            qbdsInventLocation.addSortField(fieldNum(InventLocation,
DimCenterIdSubordinated), SortOrder::Ascending);
            qbdsInventLocation.orderMode(OrderMode::GroupBy);

            // InventDim
            qbdsInventDim =
qbdsInventLocation.addDataSource(tableNum(InventDim));
            qbdsInventDim.relations(false);
            qbdsInventDim.addLink(fieldNum(InventLocation, InventLocationId),
fieldNum(InventDim, InventLocationId));
            qbdsInventDim.addRange(fieldNum(InventDim,
InventSerialId)).value(SysQuery::valueEmptyString());
            qbdsInventDim.orderMode(OrderMode::GroupBy);

           // InventSum
            qbdsInventSum = qbdsInventDim.addDataSource(tableNum(InventSum));
            qbdsInventSum.relations(false);
            qbdsInventSum.addLink(fieldNum(InventDim, InventDimId),
fieldNum(InventSum, InventDimId));
            qbdsInventSum.addSelectionField(fieldnum(InventSum, PostedQty),
SelectionField::SUM);
            qbdsInventSum.orderMode(OrderMode::GroupBy);

            // InventTable
            qbdsInventTable = qbdsInventSum.addDataSource(tableNum(InventTable));
            qbdsInventTable.relations(false);
            qbdsInventTable.addLink(fieldNum(InventSum, ItemId),
fieldNum(InventTable, ItemId));
            qbdsInventTable.addRange(fieldNum(InventTable,
RInventMTR)).value(SysQuery::value(NoYes::Yes));
```

```
        //    qbdsInventTable.addRange(fieldNum(InventTable,
ItemId)).value('110101000027');
        //    qbdsInventTable.addRange(fieldNum(InventTable,
ItemId)).value('110101000001');
            qbdsInventTable.addSortField(fieldNum(InventTable, ItemId));
            qbdsInventTable.addSortField(fieldNum(InventTable, ItemName));
            qbdsInventTable.addSortField(fieldNum(InventTable, SubGroupId));
            qbdsInventTable.addSortField(fieldNum(InventTable,
RInventMTROtdelCode));
            qbdsInventTable.orderMode(OrderMode::GroupBy);

            totalCount = SysQuery::countLoops(new queryRun(query));
            this.progressInit("Создание строк", totalCount,  AviUpdate);

            queryRun = new QueryRun(query);

            ttsbegin;

            recordSortedList.sortOrder(fieldNum(RInventMTRInventRemainsTrans,
CostsCenterCode),
                                       fieldNum(RInventMTRInventRemainsTrans, ItemId));

            while(queryRun.next())
            {
              inventTable    = queryRun.get(tableNum(InventTable));
              inventLocation = queryRun.get(tableNum(InventLocation));
              inventSum      = queryRun.get(tableNum(InventSum));

              counter ++;
              progress.setText(strfmt("%1 из %2", counter, totalCount));
              progress.incCount();

              costsCenterNew = inventLocation.DimCenterIdSubordinated != "" ?
inventLocation.DimCenterIdSubordinated : inventLocation.DimCenterId;

              postedQty = this.postedQtyByDate(inventTable.ItemId,
inventLocation.DimCenterIdSubordinated, inventLocation.DimCenterId,
RInventMTRPeriodTable::find(lastPeriod).TransDate, inventSum.PostedQty);

              if (!postedQty)
                continue;

              if (costsCenterCode != costsCenterNew)
              {
```

```
            costsCenterCode = costsCenterNew;
            rInventMTRJournalTable.JournalType    =
InventJournalType::InventRemains;

rInventMTRJournalTable.initJournalName(InventJournalType::InventRemains);
            journalTableData =
JournalTableData::newTable(rInventMTRJournalTable);
            rInventMTRJournalTable.JournalId = journalTableData.nextJournalId();
            rInventMTRJournalTable.PeriodCode = lastPeriod;
            rInventMTRJournalTable.Approved = NoYes::No;
            rInventMTRJournalTable.CostsCenterCode = costsCenterCode;
            rInventMTRJournalTable.EmplId =
EmplTable::findByUserId_W(curUserId()).EmplId;
            rInventMTRJournalTable.insert();
            journalTableData.journalTable().update();
           }

           rInventMTRInventRemainsTrans.clear();
           journalTransData =
journalTableData.journalStatic().newJournalTransData(rInventMTRInventRemainsT
rans, journalTableData);
           journalTransData.initFromJournalTable();

           rInventMTRInventRemainsTrans.PeriodCode       = lastPeriod;
           rInventMTRInventRemainsTrans.CostsCenterCode  = costsCenterCode;
           rInventMTRInventRemainsTrans.InventSubGroupId =
inventTable.SubGroupId;
           rInventMTRInventRemainsTrans.ItemId          = inventTable.ItemId;
           rInventMTRInventRemainsTrans.ItemName        = inventTable.ItemName;
           rInventMTRInventRemainsTrans.UnitID          =
InventTableModule::find(inventTable.ItemId,
ModuleInventPurchSales::Purch).UnitId;
           rInventMTRInventRemainsTrans.OtdelCode       =
inventTable.RInventMTROtdelCode;
           rInventMTRInventRemainsTrans.TransDate       = systemDateGet();
           rInventMTRInventRemainsTrans.Qty             = postedQty;
           rInventMTRInventRemainsTrans.NotUse          = NoYes::No;

           recordSortedList.ins(rInventMTRInventRemainsTrans);
          }

          //Добавление записей в таблицу с учетом сортировки
          loop = recordSortedList.first(rInventMTRInventRemainsTrans);
          while(loop)
```

```
        {
          rInventMTRInventRemainsTrans.insert();
          loop = recordSortedList.next(rInventMTRInventRemainsTrans);
        }

        while select forupdate rInventMTRJournalTable
            where rInventMTRJournalTable.PeriodCode == lastPeriod &&
rInventMTRJournalTable.JournalType == InventJournalType::InventRemains
        {
          journalTableData =
JournalTableData::newTable(rInventMTRJournalTable);
          journalTableData.initTotal();
          journalTableData.journalTable().update();
        }
        ttscommit;
      }
    ENDSOURCE
    SOURCE  createJournal_New
      public void createJournal_New()
      {
        JournalTableData            journalTableData;
        JournalTransData            journalTransData;
        RInventMTRInventRemainsTrans  rInventMTRInventRemainsTrans;
        InventTable                 inventTable;
        InventLocation               inventLocation;
        InventSum                   inventSum;

        Query                     query;
        QueryRun                    queryRun;

        RecordSortedList            recordSortedList = new
RecordSortedList(tableNum(RInventMTRInventRemainsTrans));
        boolean                     loop;

        Qty                       postedQty;

        Int                   totalCount;
        int                   counter;
        ;

        query  = this.initQueryCentralLocation();

        totalCount = SysQuery::countLoops(new queryRun(query));
        this.progressInit("Создание строк", totalCount,  AviUpdate);
```

```
        queryRun = new QueryRun(query);

        ttsbegin;

        recordSortedList.sortOrder(fieldNum(RInventMTRInventRemainsTrans,
ItemId));

        rInventMTRJournalTable.JournalType    =
InventJournalType::InventRemains;

rInventMTRJournalTable.initJournalName(InventJournalType::InventRemains);
        journalTableData = JournalTableData::newTable(rInventMTRJournalTable);
        rInventMTRJournalTable.Description =
rInventMTRJournalTable.Description + ' Общества';
        rInventMTRJournalTable.JournalId = journalTableData.nextJournalId();
        rInventMTRJournalTable.PeriodCode = lastPeriod;
        rInventMTRJournalTable.Approved = NoYes::No;
        rInventMTRJournalTable.CostsCenterCode = '!';
        rInventMTRJournalTable.EmplId =
EmplTable::findByUserId_W(curUserId()).EmplId;
        rInventMTRJournalTable.insert();
        journalTableData.journalTable().update();

        while(queryRun.next())
        {
          inventTable    = queryRun.get(tableNum(InventTable));
          inventLocation = queryRun.get(tableNum(InventLocation));
          inventSum      = queryRun.get(tableNum(InventSum));

          counter ++;
          progress.setText(strfmt("%1 из %2", counter, totalCount));
          progress.incCount();

          postedQty = this.postedQtyByDate_New(inventTable.ItemId,
RInventMTRPeriodTable::find(lastPeriod).TransDate, inventSum.PostedQty);

          if (!postedQty)
            continue;

          rInventMTRInventRemainsTrans.clear();
          journalTransData =
journalTableData.journalStatic().newJournalTransData(rInventMTRInventRemainsT
rans, journalTableData);
```

```
            journalTransData.initFromJournalTable();

            rInventMTRInventRemainsTrans.PeriodCode       = lastPeriod;
            rInventMTRInventRemainsTrans.CostsCenterCode  = '';
            rInventMTRInventRemainsTrans.InventSubGroupId =
inventTable.SubGroupId;
            rInventMTRInventRemainsTrans.ItemId           = inventTable.ItemId;
            rInventMTRInventRemainsTrans.ItemName         = inventTable.ItemName;
            rInventMTRInventRemainsTrans.UnitID           =
InventTableModule::find(inventTable.ItemId,
ModuleInventPurchSales::Purch).UnitId;
            rInventMTRInventRemainsTrans.OtdelCode        =
inventTable.RInventMTROtdelCode;
            rInventMTRInventRemainsTrans.TransDate        = systemDateGet();
            rInventMTRInventRemainsTrans.Qty              = postedQty;
            rInventMTRInventRemainsTrans.NotUse           = NoYes::No;

            recordSortedList.ins(rInventMTRInventRemainsTrans);
          }

          //Добавление записей в таблицу с учетом сортировки
          loop = recordSortedList.first(rInventMTRInventRemainsTrans);
          while(loop)
          {
            rInventMTRInventRemainsTrans.insert();
            loop = recordSortedList.next(rInventMTRInventRemainsTrans);
          }
          ttscommit;
        }
      ENDSOURCE
      SOURCE  dialog
        protected Object dialog(DialogRunbase _dialog, boolean _forceOnClient)
        {
          DialogRunBase   dialog = super(_dialog, _forceOnClient);
          ;

          dialog.caption("Расчет остатков на складах");
          dialog.addGroup("Параметры");
          dlgPeriod = dialog.addFieldValue(typeid(RInventMTRPeriodJournalId),
period, "Период", "Период", 'PeriodControl');

          return dialog;
        }
      ENDSOURCE
```

```
    SOURCE  dialogPostRun
      public void dialogPostRun(DialogRunbase dialog)
      {
         super(dialog);

         dialog.dialogForm().formRun().controlMethodOverload(true);
         dialog.dialogForm().formRun().controlMethodOverloadObject(this);
         PeriodControl = dialog.formRun().design().controlName('PeriodControl');

      }
    ENDSOURCE
    SOURCE  getFromDialog
      public boolean getFromDialog()
      {
         boolean ret;

         ret = super();

         period = dlgPeriod.value();
         if(!period || !lastPeriod)
            ret = checkFailed('Не задан период');

         return ret;
      }
    ENDSOURCE
    SOURCE  initQueryCentralLocation
      private Query initQueryCentralLocation()
      {
         Query                 query = new Query();
         QueryBuildDataSource    qbds;
         ;

         // InventLocation
         qbds = query.addDataSource(tableNum(InventLocation));
         qbds.addRange(fieldNum(InventLocation,
RInventMTRLocationType)).value(queryValue(RInventMTRLocationType::Central)
);

         qbds.addRange(fieldNum(InventLocation,
UsedLocation)).value(QueryValue(NoYes::No));
         qbds.addRange(fieldNum(InventLocation,
Blocked)).value(QueryValue(NoYes::No));
         qbds.orderMode(OrderMode::GroupBy);
```

```
            // InventDim
            qbds = qbds.addDataSource(tableNum(InventDim));
            qbds.orderMode(OrderMode::GroupBy);
            qbds.relations(false);
            qbds.addLink(fieldNum(InventLocation, InventLocationId),
fieldNum(InventDim, InventLocationId));
            qbds.addRange(fieldNum(InventDim,
InventSerialId)).value(SysQuery::valueEmptyString());

           // InventSum
            qbds = qbds.addDataSource(tableNum(InventSum));
            qbds.orderMode(OrderMode::GroupBy);
            qbds.relations(false);
            qbds.addLink(fieldNum(InventDim, InventDimId), fieldNum(InventSum,
InventDimId));
            qbds.addSelectionField(fieldnum(InventSum, PostedQty),
SelectionField::SUM);

            // InventTable
            qbds = qbds.addDataSource(tableNum(InventTable));
            qbds.orderMode(OrderMode::GroupBy);
            qbds.relations(false);
            qbds.addLink(fieldNum(InventSum, ItemId), fieldNum(InventTable,
ItemId));
            qbds.addRange(fieldNum(InventTable,
RInventMTR)).value(SysQuery::value(NoYes::Yes));
            qbds.addSortField(fieldNum(InventTable, ItemId));
            qbds.addSortField(fieldNum(InventTable, ItemName));
            qbds.addSortField(fieldNum(InventTable, SubGroupId));
            qbds.addSortField(fieldNum(InventTable, RInventMTROtdelCode));

            return query;
        }
    ENDSOURCE
    SOURCE  pack
      public container pack()
      {
            container ret;

            ret = [ CurrentVersion,  CurrentList];

            return ret;
      }
    ENDSOURCE
```

```
    SOURCE  parmRInventMTRJournalTable
      RInventMTRJournalTable
parmRInventMTRJournalTable(RInventMTRJournalTable
_rInventMTRJournalTable = rInventMTRJournalTable)
      {
        ;
        rInventMTRJournalTable = _rInventMTRJournalTable;
        return rInventMTRJournalTable;

      }
    ENDSOURCE
    SOURCE  PeriodControl_LookUp
      void PeriodControl_LookUp()
      {
        Query              query = new Query();
        QueryBuildDataSource qbds;
        SysTableLookUp      sysTableLookUp;
        ;

        sysTableLookUp =
SysTableLookUp::newParameters(tableNum(RInventMTRPeriodTable),
PeriodControl);

        sysTableLookUp.addLookupfield(fieldNum(RInventMTRPeriodTable,
JournalId), true);
        sysTableLookUp.addLookupfield(fieldNum(RInventMTRPeriodTable,
Description));
        sysTableLookUp.addLookupfield(fieldNum(RInventMTRPeriodTable,
TransDate));
        sysTableLookUp.addLookupfield(fieldNum(RInventMTRPeriodTable,
Year));

        qbds = query.addDataSource(tableNum(RInventMTRPeriodTable));
        qbds.addRange(fieldNum(RInventMTRPeriodTable,
PeriodClosed)).value(queryValue(OpenClosed_RU::Open));

        sysTableLookUp.parmQuery(query);
        sysTableLookUp.performFormLookup();
      }
    ENDSOURCE
    SOURCE  periodControl_modified
      //сделано для отображения в поле период даты периода вместо кода
периода
      //с одновременным сохранением кода периода
```

```
        boolean PeriodControl_modified()
        {
            RInventMTRPeriodTable   rInventMTRPeriodTable;
            boolean                 ret;
            ;
        // ND-Ural 02 Дек 2010 10:26:34 kpg  Begin
            PeriodControl.modified();
        // ND-Ural 02 Дек 2010 10:26:34 kpg  End
            lastPeriod = dlgPeriod.value();
            lastPeriod = PeriodControl.text();

            rInventMTRPeriodTable = RInventMTRPeriodTable::find(lastPeriod);
            if(rInventMTRPeriodTable)
            {

dlgPeriod.value(date2str(rInventMTRPeriodTable.TransDate,123,2,2,2,2,2));
                ret = true;
            }
            if(!ret)
                checkFailed('Выбранный период не существует');
            return ret;

        }

    ENDSOURCE
    SOURCE  postedQtyByDate
      private Qty  postedQtyByDate(ItemId _itemId, CostsPlaceGroupCode
_costsCenter, CostsPlaceGroupCode  _costsCenter2, TransDate _perDate, Qty
_postedQty)
        {
            InventSum             inventSum;
            InventTransPosting      inventTransPostingFinancial;
            InventTrans           inventTrans;
            InventDim             inventDim;
            InventLocation          inventLocation;
            ;

            select TableId from inventLocation
            where inventLocation.DimCenterId == _costsCenter2 &&
inventLocation.DimCenterIdSubordinated == _costsCenter &&
                (inventLocation.RInventMTRLocationType ==
RInventMTRLocationType::CentralCostsCenter ||
inventLocation.RInventMTRLocationType ==
RInventMTRLocationType::CostsCenter /*||
```

```
                inventLocation.RInventMTRLocationType ==
RInventMTRLocationType::EmergencyStock ||
inventLocation.RInventMTRLocationType ==
RInventMTRLocationType::GOCHS*/)

        join TableId from inventDim
        where inventDim.InventLocationId == inventLocation.InventLocationId

        join TableId from inventTransPostingFinancial
        where inventTransPostingFinancial.ItemId              == _itemId
&&
              inventTransPostingFinancial.InventTransPostingType ==
InventTransPostingType::Financial &&
              inventTransPostingFinancial.Dimension[2]        ==
inventDim.DimCenterId          &&
      //      inventTransPostingFinancial.IsPosted            == NoYes::Yes
&&
              inventTransPostingFinancial.TransDate           >=  _perDate

        join sum(Qty) from inventTrans
        where inventTrans.Voucher      == inventTransPostingFinancial.Voucher
&&
              inventTrans.DateFinancial == inventTransPostingFinancial.TransDate
&&
              inventTrans.inventDimId   == inventDim.inventDimId                &&
              inventTrans.InventTransId ==
inventTransPostingFinancial.InventTransId &&
              inventTrans.PackingSlipReturned == NoYes::No &&
              (inventTrans.StatusIssue == StatusIssue::Sold ||
inventTrans.StatusReceipt == StatusReceipt::Purchased);

        _postedQty -=inventTRans.Qty;

        return _postedQty;
      }
    ENDSOURCE
    SOURCE  postedQtyByDate_New
      private Qty  postedQtyByDate_New(ItemId _itemId, TransDate _perDate, Qty
_postedQty)
      {
        InventSum           inventSum;
        InventTransPosting     inventTransPostingFinancial;
        InventTrans         inventTrans;
        InventDim           inventDim;
```

```
        InventLocation          inventLocation;
        ;

        select TableId from inventLocation

        join TableId from inventDim
        where inventDim.InventLocationId == inventLocation.InventLocationId &&
inventLocation.RInventMTRLocationType == RInventMTRLocationType::Central

       join TableId from inventTransPostingFinancial
        where inventTransPostingFinancial.ItemId              == _itemId
&&
            inventTransPostingFinancial.InventTransPostingType ==
InventTransPostingType::Financial &&
         //     inventTransPostingFinancial.IsPosted            == NoYes::Yes
&&
            inventTransPostingFinancial.TransDate           >=  _perDate

        join sum(Qty) from inventTrans
        where inventTrans.Voucher       == inventTransPostingFinancial.Voucher
&&
              inventTrans.DateFinancial == inventTransPostingFinancial.TransDate
   &&
            inventTrans.inventDimId   == inventDim.inventDimId                &&
            inventTrans.InventTransId ==
inventTransPostingFinancial.InventTransId &&
            inventTrans.PackingSlipReturned == NoYes::No &&
            (inventTrans.StatusIssue == StatusIssue::Sold ||
inventTrans.StatusReceipt == StatusReceipt::Purchased);

        _postedQty -= inventTRans.Qty;

        return _postedQty;
      }
    ENDSOURCE
    SOURCE  run
      void run()
      {
        super();

        info(global::time2StrHMS(timenow()));

       if (this.checkJournal())
        {
```

```
            this.createJournal_New();
            this.createJournal();
          }

          info(global::time2StrHMS(timenow()));
        }
      ENDSOURCE
      SOURCE  unpack
        public boolean unpack(container packedClass)
        {
          Version version     = runbase::getVersion(packedClass);
          ;

          switch (version)
          {
        /*        case  CurrentVersion:
                  [version, CurrentList] = packedClass;
                  break;
        */
              default:
                  return false;
          }

          return true;

        }
      ENDSOURCE
      SOURCE  main
        static void main(Args _args)
        {
          RInventMTRCreateInventRemainsJournal createInventRemainsJournal =
new RInventMTRCreateInventRemainsJournal();
          ;

          //Если вызывается из журналов
          if(_args.dataset() == tableNum(RInventMTRJournalTable) &&
_args.record())
          {

createInventRemainsJournal.parmRInventMTRJournalTable(_args.record());
          }

          try
          {
```

```
            if(createInventRemainsJournal.prompt())
            {
               createInventRemainsJournal.run();
            }
         }
         catch(Exception::Error)
         {
            throw error("Работа процедуры была прервана");
         }
      }
CLASS  RInventMTRCreateSummaryRequestJournal
   Id 42859
   PROPERTIES
     Name              RInventMTRCreateSummaryRequestJournal
     Extends           RunBaseBatch
     RunOn             Called from
   ENDPROPERTIES

   METHODS
     Version: 3
     SOURCE  checkJournal
       private boolean checkJournal()
       {
          RInventMTRJournalTable  journalTable;
          RInventMTRCostsPlace    rInventMTRCostsPlace;
          container               conCenterCode;
          RInventMTRCostsCenter   costsCenterCode = '';
          str                     strCenterCode='';
          str                     strCenterCodeN='';
          int                     i;
          boolean                 ret = true;
          ;

          select firstonly journalTable where journalTable.JournalType ==
InventJournalType::InventSummaryRequest &&
                                      journalTable.PeriodCode == lastPeriod;
          if (journalTable)
          {
            throw error("Журнал \"Сводная заявка\" за данный период уже
существует");
          }

          while select rInventMTRCostsPlace
              order by CostsCenter
```

```
        {
          if (rInventMTRCostsPlace.CostsCenter != costsCenterCode)
          {
            conCenterCode = conins(conCenterCode, conLen(conCenterCode)+1,
rInventMTRCostsPlace.CostsCenter);
            costsCenterCode = rInventMTRCostsPlace.CostsCenter;
          }
        }

        for (i=1; i<=conLen(conCenterCode); i+=1)
        {
          costsCenterCode = conpeek(conCenterCode, i);
          select journalTable
          where journalTable.JournalType ==
InventJournalType::RInventMTRDemandCostsCenter &&
                journalTable.PeriodCode == lastPeriod &&
                journalTable.CostsCenterCode == costsCenterCode &&
                journalTable.NumOfLines !=0;
          if (!journalTable)
          {
            strCenterCode = strCenterCode + costsCenterCode + ', ';
          }
          else
          {
            if (journalTable.Approved == NoYes::No)
            {
              strCenterCodeN = strCenterCodeN + costsCenterCode+', ';
            }
          }
        }

        if (strCenterCode)
        {
          strCenterCode = substr(strCenterCode, 1, strLen(strCenterCode)-2);
          throw error(StrFmt("Журнал \"Расчета потребности в МТР по ЦЗ\" %1
не создан.", strCenterCode));
        }

        if (strCenterCodeN)
        {
          strCenterCodeN = substr(strCenterCodeN, 1, strLen(strCenterCodeN)-2);
          throw error(StrFmt("Журнал \"Расчета потребности в МТР по ЦЗ\" %1
не утвержден.", strCenterCodeN));
        }
```

```
        return ret;
      }
    ENDSOURCE
    SOURCE  classDeclaration
      // Rivis 25 Ноя 2008 10:28:28 kpg 0799_20081107_add_СводЗаявСпец
      class RInventMTRCreateSummaryRequestJournal extends RunBaseBatch
      {
        DialogField               dlgPeriod;

        RInventMTRPeriodJournalId   period, lastPeriod;
        RInventMTRJournalTable      rInventMTRJournalTable;

        FormStringControl         periodControl;

         define.CurrentVersion(1)
         LocalMacro.CurrentList
           period,
           lastPeriod
         EndMacro



      }
    ENDSOURCE
    SOURCE  dialog
      protected Object dialog(DialogRunbase _dialog, boolean _forceOnClient)
      {
        DialogRunBase      dialog = super(_dialog, _forceOnClient);
        ;

        dialog.caption("Сводная заявка");
        dialog.addGroup("Параметры");
        dlgPeriod = dialog.addFieldValue(typeId(RInventMTRPeriodJournalId),
period, "Период", "Период", 'periodControl');

        return dialog;
      }
    ENDSOURCE
    SOURCE  dialogPostRun
      public void dialogPostRun(DialogRunbase dialog)
      {
        ;
```

```
        super(dialog);

        dialog.dialogForm().formRun().controlMethodOverload(true);
        dialog.dialogForm().formRun().controlMethodOverloadObject(this);
        periodControl = dialog.formRun().design().controlName('periodControl');
      }
    ENDSOURCE
    SOURCE  getFromDialog
      public boolean getFromDialog()
      {
        boolean ret;

        ret = super();

        period         = dlgPeriod.value();

        if(!period || !lastPeriod)
          ret = checkFailed('Не задан период');

        return ret;
      }
    ENDSOURCE
    SOURCE  initQuery
      private Query initQuery()
      {
        Query                 query = new Query();
        QueryBuildDataSource    qbds;
        ;

        qbds = query.addDataSource(tableNum(RInventMTRJournalTable));
        qbds.addRange(fieldNum(RInventMTRJournalTable,
periodCode)).value(queryValue(lastPeriod));
        qbds.addRange(fieldNum(RInventMTRJournalTable,
JournalType)).value(queryValue(InventJournalType::RInventMTRDemandCostsCent
er));
        qbds.addRange(fieldNum(RInventMTRJournalTable,
Approved)).value(queryValue(NoYes::Yes));                    // выбираются
только утвержденные журналы

        qbds = qbds.addDataSource(tableNum(RInventMTRDemandTrans));
        qbds.orderMode(OrderMode::GroupBy);
        qbds.relations(false);
        qbds.addLink(fieldNum(RInventMTRJournalTable, JournalId),
fieldNum(RInventMTRDemandTrans, journalId));
```

```
            qbds.addSortField(fieldNum(RInventMTRDemandTrans, OtdelCode));      //
            qbds.addSortField(fieldNum(RInventMTRDemandTrans,
InventSubGroupId));
            qbds.addSortField(fieldNum(RInventMTRDemandTrans, ItemId));
          //   qbds.addSortField(fieldNum(RInventMTRDemandTrans, ItemName));    ///
          //    qbds.addSortField(fieldNum(RInventMTRDemandTrans, UnitId));      //
            qbds.addSortField(fieldNum(RInventMTRDemandTrans, PeriodCode));    ///
            qbds.addSortField(fieldNum(RInventMTRDemandTrans, PeriodDate));   ///
            qbds.addSortField(fieldNum(RInventMTRDemandTrans, CostsPrice));   ///
            qbds.addSortField(fieldNum(RInventMTRDemandTrans, Priority));

            // количество
             qbds.addSelectionField(fieldNum(RInventMTRDemandTrans, QtyPurch),
SelectionField::Sum);

            // стоимость
             qbds.addSelectionField(fieldNum(RInventMTRDemandTrans, CostsPurch),
SelectionField::Sum);

            // Агентское вознаграждение
            qbds.addSelectionField(fieldNum(RInventMTRDemandTrans,
CostsAgencyCompensationPurch), SelectionField::Sum);

            // ТЗР
            qbds.addSelectionField(fieldNum(RInventMTRDemandTrans,
CostsTransportationPurch), SelectionField::Sum);

            // Клэффициент качества
            qbds.addSelectionField(fieldNum(RInventMTRDemandTrans,
CostsQualityFactorPurch), SelectionField::Sum);

            qbds.addSelectionField(fieldNum(RInventMTRDemandTrans,
QtyFreeRemains), SelectionField::Sum);

            qbds.orderMode(OrderMode::GroupBy);


            return query;
        }
    ENDSOURCE
    SOURCE  pack
      public container pack()
      {
```

```
            container ret;

            ret = [ CurrentVersion,  CurrentList];

            return ret;
        }
    ENDSOURCE
    SOURCE  parmRInventMTRJournalTable
      RInventMTRJournalTable
parmRInventMTRJournalTable(RInventMTRJournalTable
_rInventMTRJournalTable = rInventMTRJournalTable)
        {
            ;
            rInventMTRJournalTable = _rInventMTRJournalTable;
            return rInventMTRJournalTable;

        }
    ENDSOURCE
    SOURCE  periodControl_LookUp
      void periodControl_LookUp()
      {
            Query               query = new Query();
            SysTableLookUp      sysTableLookUp;
            ;

            sysTableLookUp =
SysTableLookUp::newParameters(tableNum(RInventMTRPeriodTable),
periodControl);

query.addDataSource(tableNum(RInventMTRPeriodTable)).addRange(fieldNum(RI
nventMTRPeriodTable, PeriodClosed)).value(queryValue(OpenClosed_RU::Open));

            sysTableLookUp.parmQuery(query);
            sysTableLookUp.addLookupfield(fieldNum(RInventMTRPeriodTable,
JournalId), true);
            sysTableLookUp.addLookupfield(fieldNum(RInventMTRPeriodTable,
Description));
            sysTableLookUp.addLookupfield(fieldNum(RInventMTRPeriodTable,
TransDate));
            sysTableLookUp.addLookupfield(fieldNum(RInventMTRPeriodTable,
Year));
            sysTableLookUp.performFormLookup();
        }
    ENDSOURCE
```

```
    SOURCE  periodControl_modified
      //сделано для отображения в поле период даты периода вместо кода
периода
      //с одновременным сохранением кода периода
      boolean periodControl_modified()
      {
        RInventMTRPeriodTable   rInventMTRPeriodTable;
        boolean                 ret;
        ;
        periodControl.modified();
        lastPeriod = dlgPeriod.value();
        rInventMTRPeriodTable = RInventMTRPeriodTable::find(lastPeriod);
        if(rInventMTRPeriodTable)
        {

dlgPeriod.value(date2str(rInventMTRPeriodTable.TransDate,123,2,2,2,2,2));
           ret = true;
        }
        if(!ret)
            checkFailed('Выбранный период не существует');
        return ret;

      }

    ENDSOURCE
    SOURCE  qtyGroup
      void qtyGroup(RInventMTRSummaryRequestTrans
_rInventMTRSummaryRequestTrans)
      {
        RInventMTRTransQty          rInventMTRTransQty,
rInventMTRTransQtyLocal;
        RInventMTRJournalTable      rInventMTRJournalTableLocal;
        RInventMTRDemandTrans       rInventMTRDemandTransLocal;
        RInventMTRSummaryRequestTrans  rInventMTRSummaryRequestTrans;
        ;

        while select rInventMTRJournalTableLocal
            where rInventMTRJournalTableLocal.JournalType ==
InventJournalType::RInventMTRDemandCostsCenter &&
                rInventMTRJournalTableLocal.PeriodCode == lastPeriod &&
                rInventMTRJournalTableLocal.Approved == NoYes::Yes

            join rInventMTRDemandTransLocal
```

```
            where rInventMTRDemandTransLocal.JournalId ==
rInventMTRJournalTableLocal.JournalId &&
                  rInventMTRDemandTransLocal.PeriodCode ==
rInventMTRJournalTableLocal.PeriodCode &&
                  rInventMTRDemandTransLocal.CostsCenter ==
rInventMTRJournalTableLocal.CostsCenterCode &&
                  rInventMTRDemandTransLocal.ItemId ==
_rInventMTRSummaryRequestTrans.ItemId
        {
          while select PeriodCode, QtyMin, QtyMax, QtyGOCHS,
QtyEmergencyStock from rInventMTRTransQty
              where rInventMTRTransQty.RefTableId ==
rInventMTRDemandTransLocal.TableId &&
                  rInventMTRTransQty.RefRecId ==
rInventMTRDemandTransLocal.RecId
          {
            rInventMTRTransQtyLocal =
RInventMTRTransQty::find(_rInventMTRSummaryRequestTrans.TableId,
_rInventMTRSummaryRequestTrans.RecId, rInventMTRTransQty.PeriodCode,
true);
            ttsbegin;
            if (rInventMTRTransQtyLocal)
            {
              rInventMTRTransQtyLocal.QtyMin =
rInventMTRTransQtyLocal.QtyMin + rInventMTRTransQty.QtyMin;
              rInventMTRTransQtyLocal.QtyMax =
rInventMTRTransQtyLocal.QtyMax + rInventMTRTransQty.QtyMax;
              rInventMTRTransQtyLocal.QtyGOCHS =
rInventMTRTransQtyLocal.QtyGOCHS + rInventMTRTransQty.QtyGOCHS;
              rInventMTRTransQtyLocal.QtyEmergencyStock =
rInventMTRTransQtyLocal.QtyEmergencyStock +
rInventMTRTransQty.QtyEmergencyStock;
              rInventMTRTransQtyLocal.doUpdate();
            }
            ttscommit;
          }
        }
      }
    ENDSOURCE
    SOURCE  run
      public void run()
      {
        JournalTableData            journalTableData;
        JournalTransData            journalTransData;
```

```
        RInventMTRDemandTrans          rInventMTRDemandTrans;
        RInventMTRSummaryRequestTrans  rInventMTRSummaryRequestTrans;
        RInventMTRJournalTable         journalTable;

        Query                  query;
        QueryRun                 queryRun;
        QueryBuildDataSource         qbds;

        Int                totalCount;
        int                counter;
        ;

        super();

        this.checkJournal();

        query = this.initQuery();

        totalCount = SysQuery::countTotal(new queryRun(query));
        this.progressInit("Создание строк", totalCount,  AviUpdate);

        queryRun = new QueryRun(query);

        ttsbegin;
        journalTableData =  JournalTableData::newTable(rInventMTRJournalTable);

        while(queryRun.next())
        {
          counter ++;
          progress.setText(strfmt("%1 из %2", counter, totalCount));
          progress.incCount();

          rInventMTRDemandTrans =
queryRun.get(tableNum(RInventMTRDemandTrans));

          rInventMTRSummaryRequestTrans.clear();

          journalTransData =
journalTableData.journalStatic().newJournalTransData(rInventMTRSummaryReques
tTrans, journalTableData);
          journalTransData.initFromJournalTable();

          rInventMTRSummaryRequestTrans.PeriodCode       = lastPeriod;
```

```
        rInventMTRSummaryRequestTrans.PeriodDate        =
rInventMTRDemandTrans.PeriodDate;
        rInventMTRSummaryRequestTrans.ItemId            =
rInventMTRDemandTrans.ItemId;
        rInventMTRSummaryRequestTrans.InventSubGroupId   =
rInventMTRDemandTrans.InventSubGroupId;
        rInventMTRSummaryRequestTrans.ItemName          =
rInventMTRDemandTrans.ItemName;
        rInventMTRSummaryRequestTrans.UnitID            =
rInventMTRDemandTrans.UnitID;
        rInventMTRSummaryRequestTrans.OtdelCode         =
rInventMTRDemandTrans.OtdelCode;
        rInventMTRSummaryRequestTrans.CostsPrice        =
rInventMTRDemandTrans.CostsPrice;

        rInventMTRSummaryRequestTrans.QtyDeclared                =
rInventMTRDemandTrans.QtyPurch;
        rInventMTRSummaryRequestTrans.CostsDeclared              =
rInventMTRDemandTrans.CostsPurch;
        rInventMTRSummaryRequestTrans.CostsAgencyCompensationDeclared =
rInventMTRDemandTrans.CostsAgencyCompensationPurch;
        rInventMTRSummaryRequestTrans.CostsTransportationDeclared    =
rInventMTRDemandTrans.CostsTransportationPurch;
        rInventMTRSummaryRequestTrans.CostsQualityFactorPurch        =
rInventMTRDemandTrans.CostsQualityFactorPurch;

        rInventMTRSummaryRequestTrans.QtyRemains        =
RInventMTRInventRemainsTrans::find(lastPeriod,
",rInventMTRDemandTrans.ItemId).QtyRemains();
        rInventMTRSummaryRequestTrans.QtyPurch          =
(rInventMTRSummaryRequestTrans.QtyDeclared -
rInventMTRSummaryRequestTrans.QtyRemains -
rInventMTRSummaryRequestTrans.QtyFreeRemains()) > 0 ?

(rInventMTRSummaryRequestTrans.QtyDeclared -
rInventMTRSummaryRequestTrans.QtyRemains  -
rInventMTRSummaryRequestTrans.QtyFreeRemains()) : 0;

        rInventMTRSummaryRequestTrans.QtyDeclaredUMTS    =
rInventMTRSummaryRequestTrans.QtyPurch;
        rInventMTRSummaryRequestTrans.QtyDeclaredUMTSOne =
rInventMTRSummaryRequestTrans.QtyPurch;
        rInventMTRSummaryRequestTrans.QtyDeclaredUMTSTwo = 0;
```

```
            rInventMTRSummaryRequestTrans.TransDate = systemDateGet();

            rInventMTRSummaryRequestTrans.insert();

        //      this.qtyGroup(rInventMTRSummaryRequestTrans);           //
Заполняется таблица RInventMTRTransQty

RInventMTRTransQty::updateSumTransQty_Summary(rInventMTRSummaryReque
stTrans);  // Заполняется таблица RInventMTRTransQty

RInventMTRTransQty::updateTransQtyPurch_Summary(rInventMTRSummaryRequ
estTrans);

            journalTransData.addTotal(false,false);
          }

          JournalTable = journalTableData.journalTable();
          journalTableData.initTotal();
          JournalTable.PeriodCode = lastPeriod;
          journalTable.Approved = NoYes::No;
          journalTable.EmplId = EmplTable::findByUserId_W(curUserId()).EmplId;

          journalTable.update();
          journalTableData.journalTable().update();

          ttscommit;
        }

      ENDSOURCE
      SOURCE  runAll
        public void runAll()
        {
          JournalTableData            journalTableData;
          JournalTransData            journalTransData;

          RInventMTRDemandTrans         rInventMTRDemandTrans;
          RInventMTRSummaryRequestTrans  rInventMTRSummaryRequestTrans;
          RInventMTRJournalTable        journalTable;
          InventTable                 inventTable;

          Query                     query;
          QueryRun                    queryRun;
          QueryBuildDataSource          qbds;

          Int                   totalCount;
```

```
        int                 counter;
        ;

      //  super();

        this.checkJournal();

        query = this.initQuery();

        totalCount = SysQuery::countTotal(new queryRun(query));
        this.progressInit("Создание строк", totalCount,  AviUpdate);

        queryRun = new QueryRun(query);

        ttsbegin;
     //   journalTableData =
JournalTableData::newTable(rInventMTRJournalTable);

        rInventMTRJournalTable.JournalType    =
InventJournalType::InventSummaryRequest;

rInventMTRJournalTable.initJournalName(InventJournalType::InventSummaryRequ
est);
        journalTableData = JournalTableData::newTable(rInventMTRJournalTable);
        rInventMTRJournalTable.Description =
rInventMTRJournalTable.Description + ' по Обществу';
        rInventMTRJournalTable.JournalId = journalTableData.nextJournalId();
        rInventMTRJournalTable.PeriodCode = lastPeriod;
        rInventMTRJournalTable.Approved = NoYes::No;
        rInventMTRJournalTable.CostsCenterCode = '';
        rInventMTRJournalTable.OtdelCode = '';
        rInventMTRJournalTable.EmplId =
EmplTable::findByUserId_W(curUserId()).EmplId;
        rInventMTRJournalTable.insert();
        journalTableData.journalTable().update();

        while(queryRun.next())
        {
          counter ++;
          progress.setText(strfmt("%1 из %2", counter, totalCount));
          progress.incCount();

          rInventMTRDemandTrans =
queryRun.get(tableNum(RInventMTRDemandTrans));
```

```
            if (rInventMTRDemandTrans.QtyPurch == 0)
              continue;

            rInventMTRSummaryRequestTrans.clear();

            journalTransData =
journalTableData.journalStatic().newJournalTransData(rInventMTRSummaryReques
tTrans, journalTableData);
            journalTransData.initFromJournalTable();

            inventTable = InventTable::find(rInventMTRDemandTrans.ItemId);

            rInventMTRSummaryRequestTrans.PeriodCode         = lastPeriod;
            rInventMTRSummaryRequestTrans.PeriodDate         =
rInventMTRDemandTrans.PeriodDate;
            rInventMTRSummaryRequestTrans.SummaryRequestJournalType =
RInventMTRSummaryRequestJournalType::All;
            rInventMTRSummaryRequestTrans.ItemId             =
rInventMTRDemandTrans.ItemId;
            rInventMTRSummaryRequestTrans.InventSubGroupId   =
rInventMTRDemandTrans.InventSubGroupId;
            rInventMTRSummaryRequestTrans.ItemName           =
inventTable.ItemName;
            rInventMTRSummaryRequestTrans.UnitID             =
InventTableModule::find(inventTable.ItemId,
ModuleInventPurchSales::Purch).UnitId;
            rInventMTRSummaryRequestTrans.OtdelCode          =
rInventMTRDemandTrans.OtdelCode;
            rInventMTRSummaryRequestTrans.CostsPrice         =
rInventMTRDemandTrans.CostsPrice;

            rInventMTRSummaryRequestTrans.QtyDeclared                  =
rInventMTRDemandTrans.QtyPurch;
            rInventMTRSummaryRequestTrans.CostsDeclared                =
rInventMTRDemandTrans.CostsPurch;
            rInventMTRSummaryRequestTrans.CostsAgencyCompensationDeclared =
rInventMTRDemandTrans.CostsAgencyCompensationPurch;
            rInventMTRSummaryRequestTrans.CostsTransportationDeclared    =
rInventMTRDemandTrans.CostsTransportationPurch;
            rInventMTRSummaryRequestTrans.CostsQualityFactorPurch        =
rInventMTRDemandTrans.CostsQualityFactorPurch;
```

```
            rInventMTRSummaryRequestTrans.QtyRemains        =
RInventMTRInventRemainsTrans::find(lastPeriod,
",rInventMTRDemandTrans.ItemId).QtyRemains();
            rInventMTRSummaryRequestTrans.QtyPurch          =
(rInventMTRSummaryRequestTrans.QtyDeclared -
rInventMTRSummaryRequestTrans.QtyRemains -
rInventMTRSummaryRequestTrans.QtyFreeRemains()) > 0 ?

(rInventMTRSummaryRequestTrans.QtyDeclared -
rInventMTRSummaryRequestTrans.QtyRemains  -
rInventMTRSummaryRequestTrans.QtyFreeRemains()) : 0;

            rInventMTRSummaryRequestTrans.QtyDeclaredUMTS    =
rInventMTRSummaryRequestTrans.QtyPurch;
            rInventMTRSummaryRequestTrans.QtyDeclaredUMTSOne =
rInventMTRSummaryRequestTrans.QtyPurch;
            rInventMTRSummaryRequestTrans.QtyDeclaredUMTSTwo = 0;

            rInventMTRSummaryRequestTrans.TransDate = systemDateGet();

            rInventMTRSummaryRequestTrans.insert();

        //      this.qtyGroup(rInventMTRSummaryRequestTrans);           //
Заполняется таблица RInventMTRTransQty

RInventMTRTransQty::updateSumTransQty_Summary(rInventMTRSummaryReque
stTrans);  // Заполняется таблица RInventMTRTransQty

RInventMTRTransQty::updateTransQtyPurch_Summary(rInventMTRSummaryRequ
estTrans);

         ////     journalTransData.addTotal(false,false);
           }

         /*   JournalTable = journalTableData.journalTable();
           journalTableData.initTotal();
           JournalTable.PeriodCode = lastPeriod;
           journalTable.Approved = NoYes::No;
           journalTable.EmplId = EmplTable::findByUserId_W(curUserId()).EmplId;

           journalTable.update();
           journalTableData.journalTable().update();
         */
           ttscommit;
        }
```

```
    ENDSOURCE
    SOURCE  runOtdel
      public void runOtdel()
      {
        JournalTableData              journalTableData;
        JournalTransData              journalTransData;

        RInventMTRDemandTrans         rInventMTRDemandTrans;
        RInventMTRSummaryRequestTrans rInventMTRSummaryRequestTrans;
       //   RInventMTRJournalTable        journalTable;
        InventTable                   inventTable;

        Query                         query;
        QueryRun                      queryRun;
        QueryBuildDataSource          qbds;

        RInventMTROtdelCode           otdelCode = '!';

        Int                   totalCount;
        int                   counter;
        ;

        this.checkJournal();

        query = this.initQuery();

        totalCount = SysQuery::countTotal(new queryRun(query));
        this.progressInit("Создание строк", totalCount,  AviUpdate);

        queryRun = new QueryRun(query);

        ttsbegin;

        while(queryRun.next())
        {
          counter ++;
          progress.setText(strfmt("%1 из %2", counter, totalCount));
          progress.incCount();

          rInventMTRDemandTrans =
queryRun.get(tableNum(RInventMTRDemandTrans));
          if (rInventMTRDemandTrans.QtyPurch == 0)
            continue;
```

```
            if (otdelCode != rInventMTRDemandTrans.OtdelCode)
            {
              otdelCode = rInventMTRDemandTrans.OtdelCode;
              rInventMTRJournalTable.JournalType    =
InventJournalType::InventSummaryRequest;

rInventMTRJournalTable.initJournalName(InventJournalType::InventSummaryRequ
est);
              journalTableData =
JournalTableData::newTable(rInventMTRJournalTable);
              rInventMTRJournalTable.Description =
rInventMTRJournalTable.Description + ' по ' +
rInventMTRDemandTrans.OtdelCode + ' отделу';
              rInventMTRJournalTable.JournalId = journalTableData.nextJournalId();
              rInventMTRJournalTable.PeriodCode = lastPeriod;
              rInventMTRJournalTable.Approved = NoYes::No;
              rInventMTRJournalTable.CostsCenterCode = '';
              rInventMTRJournalTable.OtdelCode =
rInventMTRDemandTrans.OtdelCode;
              rInventMTRJournalTable.EmplId =
EmplTable::findByUserId_W(curUserId()).EmplId;
              rInventMTRJournalTable.insert();
              journalTableData.journalTable().update();
            }

            rInventMTRSummaryRequestTrans.clear();

            journalTransData =
journalTableData.journalStatic().newJournalTransData(rInventMTRSummaryReques
tTrans, journalTableData);
            journalTransData.initFromJournalTable();

            inventTable = InventTable::find(rInventMTRDemandTrans.ItemId);

            rInventMTRSummaryRequestTrans.PeriodCode        = lastPeriod;
            rInventMTRSummaryRequestTrans.PeriodDate        =
rInventMTRDemandTrans.PeriodDate;
            rInventMTRSummaryRequestTrans.SummaryRequestJournalType =
RInventMTRSummaryRequestJournalType::Otdel;
            rInventMTRSummaryRequestTrans.ItemId            =
rInventMTRDemandTrans.ItemId;
            rInventMTRSummaryRequestTrans.InventSubGroupId   =
rInventMTRDemandTrans.InventSubGroupId;
```

```
            rInventMTRSummaryRequestTrans.ItemName          =
inventTable.ItemName;
            rInventMTRSummaryRequestTrans.UnitID            =
InventTableModule::find(inventTable.ItemId,
ModuleInventPurchSales::Purch).UnitId;
            rInventMTRSummaryRequestTrans.OtdelCode         =
rInventMTRDemandTrans.OtdelCode;
            rInventMTRSummaryRequestTrans.CostsPrice        =
rInventMTRDemandTrans.CostsPrice;

            rInventMTRSummaryRequestTrans.QtyDeclared                   =
rInventMTRDemandTrans.QtyPurch;
            rInventMTRSummaryRequestTrans.CostsDeclared                 =
rInventMTRDemandTrans.CostsPurch;
            rInventMTRSummaryRequestTrans.CostsAgencyCompensationDeclared =
rInventMTRDemandTrans.CostsAgencyCompensationPurch;
            rInventMTRSummaryRequestTrans.CostsTransportationDeclared     =
rInventMTRDemandTrans.CostsTransportationPurch;
            rInventMTRSummaryRequestTrans.CostsQualityFactorPurch         =
rInventMTRDemandTrans.CostsQualityFactorPurch;

            rInventMTRSummaryRequestTrans.QtyRemains        =
RInventMTRInventRemainsTrans::find(lastPeriod,
",rInventMTRDemandTrans.ItemId).QtyRemains();
            rInventMTRSummaryRequestTrans.QtyPurch          =
(rInventMTRSummaryRequestTrans.QtyDeclared -
rInventMTRSummaryRequestTrans.QtyRemains -
rInventMTRSummaryRequestTrans.QtyFreeRemains()) > 0 ?

(rInventMTRSummaryRequestTrans.QtyDeclared -
rInventMTRSummaryRequestTrans.QtyRemains  -
rInventMTRSummaryRequestTrans.QtyFreeRemains()) : 0;

            rInventMTRSummaryRequestTrans.QtyDeclaredUMTS    =
rInventMTRSummaryRequestTrans.QtyPurch;
            rInventMTRSummaryRequestTrans.QtyDeclaredUMTSOne =
rInventMTRSummaryRequestTrans.QtyPurch;
            rInventMTRSummaryRequestTrans.QtyDeclaredUMTSTwo = 0;

            rInventMTRSummaryRequestTrans.TransDate = systemDateGet();

            rInventMTRSummaryRequestTrans.insert();
```

```
RInventMTRTransQty::updateSumTransQty_Summary(rInventMTRSummaryReque
stTrans);  // Заполняется таблица RInventMTRTransQty

RInventMTRTransQty::updateTransQtyPurch_Summary(rInventMTRSummaryRequ
estTrans);
          }

          while select forupdate rInventMTRJournalTable
               where rInventMTRJournalTable.PeriodCode == lastPeriod &&
rInventMTRJournalTable.JournalType ==
InventJournalType::InventSummaryRequest
          {
            journalTableData =
JournalTableData::newTable(rInventMTRJournalTable);
            journalTableData.initTotal();
            journalTableData.journalTable().update();
          }

          ttscommit;
       }

    ENDSOURCE
    SOURCE  unpack
      public boolean unpack(container packedClass)
      {
          Version version     = runbase::getVersion(packedClass);
          ;

          switch (version)
          {
      /*        case  CurrentVersion:
                 [version, CurrentList] = packedClass;
                 break;
      */        default:

                 return false;
          }

          return true;

      }
    ENDSOURCE
    SOURCE  main
      static void main(Args _args)
```

```
    {
        RInventMTRCreateSummaryRequestJournal
rInventMTRCreateSummaryRequestJournal = new
RInventMTRCreateSummaryRequestJournal();

        ;

        if(_args.dataset() == tableNum(RInventMTRJournalTable) &&
_args.record())
        {

rInventMTRCreateSummaryRequestJournal.parmRInventMTRJournalTable(_args.re
cord());
        }
        else
        {
            throw error("Не выбран журнал Сводная заявка");
        }

        try
        {
            if(rInventMTRCreateSummaryRequestJournal.prompt())
            {
    //            rInventMTRCreateSummaryRequestJournal.run();
              if (rInventMTRCreateSummaryRequestJournal.checkJournal())
              {
                rInventMTRCreateSummaryRequestJournal.runAll();
                rInventMTRCreateSummaryRequestJournal.runOtdel();
              }
            }
        }
        catch(Exception::Error)
        {
            if(!
rInventMTRCreateSummaryRequestJournal.parmRInventMTRJournalTable().Period
Code)
            {
                ttsbegin;

rInventMTRCreateSummaryRequestJournal.parmRInventMTRJournalTable().delete(
);
                ttscommit;
            }
            infolog.add(Exception::Error,"Работа процедуры была прервана");
        }
```