Geekbrains

**Разработка прототипа мобильного приложения «Инвентаризация»**

Программа:

Разработчик – Android-разработка

Перминов Сергей Олегович

Санкт-Петербург

2024

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 3](#_Toc44368368)

[1. ПОСТАНОВКА ЗАДАЧИ 4](#_Toc44368369)

[1.1. Функциональные требования 4](#_Toc44368371)

[1.2. Нефункциональные требования 9](#_Toc44368372)

[2. ПРОЕКТИРОВАНИЕ АРХИТЕКТУРЫ 10](#_Toc44368373)

[2.1. Микросервисная архитектура 10](#_Toc44368374)

[2.2. Брокер сообщений 11](#_Toc44368375)

[2.3. Архитектура микросервиса 12](#_Toc44368375)

[2.3.1. Инструменты для программной реализации 13](#_Toc44368375)

[2.3.1.1. Слой обеспечения безопасности 14](#_Toc44368375)

[2.3.1.1. Слой REST контроллера 16](#_Toc44368375)

[2.3.1.1. Слой бизнес логики 18](#_Toc44368375)

[2.3.1.1. Слой взаимодействия с базой данных 18](#_Toc44368375)

[3. ПРОГРАММНАЯ РЕАЛИЗАЦИЯ 20](#_Toc44368373)

[3.1. Определение сущностей 20](#_Toc44368375)

[3.2. Определение атрибутов сущностей со связями 20](#_Toc44368375)

[3.3. Определение операций 28](#_Toc44368375)

[3.4. Проектирование базы данных 36](#_Toc44368375)

[3.5. Среда разработки и система сборки проекта 37](#_Toc44368375)

[3.6. Создание проекта 38](#_Toc44368375)

[3.7. Подключение базы данных 40](#_Toc44368375)

[3.8. Системы контроля версий 42](#_Toc44368375)

[3.9. Структура папок проекта 45](#_Toc44368375)

[3.10. Использование Lombok 46](#_Toc44368375)

[3.11. Сущности проекта и интерфейсы репозитория 46](#_Toc44368375)

[3.12. Реализация бизнес логики 47](#_Toc44368375)

[3.13. Реализация REST контроллера 48](#_Toc44368375)

[3.14. Реализация системы безопасности 48](#_Toc44368375)

[3.15. Реализация слоя брокера сообщений 51](#_Toc44368375)

[3.16. Реализация системы служебных сообщений 51](#_Toc44368375)

[3.17. Описание ресурсов 52](#_Toc44368375)

[4. ТЕСТИРОВАНИЕ 71](#_Toc44368382)

[ЗАКЛЮЧЕНИЕ 72](#_Toc44368385)

[Приложение 1 73](#_Toc44368387)

[Приложение 2 74](#_Toc44368387)

[Приложение 3 75](#_Toc44368387)

**ВВЕДЕНИЕ**

Данная работа нацелена на разработку прототипа мобильного приложения для обеспечения автоматизации бизнес процессов по инвентаризации запасов, в соответствии с функциональными требованиями АО «Балтийский завод».

Разрабатываемое мобильное приложение является частью системы управления складскими запасами на базе ERP Infor LN и представляет собой интерфейс для сбора текущего состояния запасов на складах АО «Балтийский завод». Приложение реализуется на базе операционной системы Android для терминалов сбора данных Honeywell EDA50k-1.

Основные потребители приложения – работники склада и представители инвентаризационной комиссии.

1. **ПОСТАНОВКА ЗАДАЧИ**

Цель данной работы — разработать прототип мобильного приложение на базе операционной системы Android для терминала сбора данных Honeywell EDA50k-1, реализующее задачи инвентаризации складских запасов АО «Балтийский завод».

Разработка мобильного приложения производится в соответствии с функциональными требованиями, спецификой бизнес процессов, технического оснащения АО «Балтийский завод».

Для выполнения работы необходимо решить следующие подзадачи:

- получить и оформить список функциональных требований;

- получить и оформить список нефункциональных требований;

- спроектировать архитектуру;

- подобрать необходимые инструменты;

- выполнить программную реализацию;

- провести функциональное тестирование;

* 1. **Функциональные требования.**
     1. **Введение**
* Требования разработаны на основании запроса от бизнес подразделения на разработку мобильного приложения;
* Описаны бизнес-процессы по просмотру, учету и поиску информации о складских запасах с использованием мобильного приложения;
* Разрабатываемое мобильное приложение будет взаимодействовать с системой управления предприятием на базе ERP Infor LN в части обмена данными, через импорт/экспорт csv файла.
  + 1. **Основной бизнес процесс**

1. В ERP Infor LN формируются заказы на циклическую инвентаризацию.
2. Заказы на циклическую инвентаризацию со складскими остатками экспортируются в csv файл.
3. Файл csv импортируется в мобильное приложение терминала сбора данных.
4. Терминал сбора данных готов для фиксации фактических запасов.
5. Сканируется DataMatrix код склада.
6. Сканируется DataMatrix код места.
7. Сканируется совмещенный DataMatrix код с ТМЦ и партией.
8. По необходимости устанавливаются опции: печать этикетки, брак, неликвид.
9. В диалоговом окне вводится количество фактического запаса.
10. Позиция сохраняется в базе данных мобильного приложения.
11. По окончании инвентаризации данные экспортируются в файл csv для последующего импорта в ERP Infor LN



Рисунок 1.1.Общая схема основного процесса

* + - 1. **Функциональные требования основного бизнес процесса**

1. Файл импорта в мобильное приложение должен иметь следующие поля:

* Номер заказа на циклическую инвентаризацию
* Строка заказа на циклическую инвентаризацию
* Номер склада
* Место хранения запаса
* Партия
* Код запаса
* Количество запаса по данным ERP Infor LN
* Единица измерения

1. Файл экспорта из мобильного приложения должен иметь следующие поля:

* Номер заказа на циклическую инвентаризацию
* Строка заказа на циклическую инвентаризацию
* Номер склада
* Место хранения запаса
* Партия
* Код запаса
* Количество фактического запаса
* Дата фиксации фактического запаса
* Признак необходимости печати этикетки запаса
* Признак брака
* Признак неликвида

1. Импорт/экспорт файла должен осуществляться только сотрудником дирекции информационных технологий.
2. Перед импортом данных в мобильное приложение данные предыдущей инвентаризации должны автоматически удаляться.
3. Экспорт данных инвентаризации из мобильного устройство должен быть неограничен по количеству.
4. Главный экран приложения должен содержать следующие поля:

* Склад
* Место
* Код запаса
* Партия
* Опция печати этикетки
* Опция признака брака
* Опция признака неликвида

1. Заполнение полей склад, место, код запаса, партия необходимо производить сканированием соответствующих этикеток с DataMatrix кодом значений полей, используя штатный сканер терминала сбора данных.
2. При заполнении полей посредством сканера необходимо распознавание отсканированного значения и автоматическая подстановка значения в соответствующее поле.
3. Ручное заполнение, изменение полей склад, место, код запаса, партия.
4. При заполнении поля склад необходимо автоматически очищать поля место, код запаса, партия.
5. При заполнении поля место необходимо автоматически очищать поля код запаса, партия.
6. При ручном заполнении поля код запаса необходимо автоматически очищать поле партия.
7. Валидация полей склад, место, код запаса, партия
8. Автоматическое формирование диалогового окна для ввода количества фактического запаса после заполнения полей склад, место, код запаса, партия.
9. Поля диалогового окна:

* Количество по ИНВ-3
* Единица измерения по ИНВ-3
* Место
* Поле для ввода фактического количества

1. Если позиция была учтена ранее, то в поле ввода фактического количества должно отображаться фактическое количество, введенное ранее.
2. Автоматическая очистка полей код запаса и партия после ввода количества фактического запаса.
3. Сохранение данных фактического запаса в базу данных терминала сбора данных.
   * 1. **Вспомогательный бизнес процесс импорта / экспорта файла**
4. В мобильном приложении выбирается раздел Настройки.
5. Вводится пароль.
6. В случае неправильного пароля приложение возвращается на главный экран.
7. В случае правильного пароля, становятся доступны команды для импорта и экспорта данных.
8. Импорт данных производится путем выбора файла с данными и заполнения ими базы данных приложения.
9. Экспорт данных производится путем формирования файла экспорта с данными итогов инвентаризации.



Рисунок 1.2.Общая схема вспомогательного бизнес процесса импорта данных



Рисунок 1.3.Общая схема вспомогательного бизнес процесса экспорта данных

* + - 1. **Функциональные требования вспомогательного процесса импорта / экспорта данных**

1. Доступ к функциональности импорта / экспорта только через пароль.
2. В текущей реализации мобильного приложения пароль постоянный, не подлежит изменению.
3. Файлы импорта и экспорта имеют ограниченный набор полей описанных в разделе 1.1.2.1. Функциональные требования основного бизнес процесса.
4. Перед импортом данных предыдущие итоги инвентаризации должны автоматически очищаться.
5. Экспорт данных должен проводиться неограниченное количество раз.
6. Поля файла импорта:

* заказ на циклическую инвентаризацию
* строка заказа на циклическую инвентаризацию
* склад
* место
* партия
* изделие (запас, ТМЦ)
* количество по ЕНВ-3
* единица измерения
* заказ на закупку

1. Поля файла экспорта:

* заказ на циклическую инвентаризацию
* строка заказа на циклическую инвентаризацию
* склад
* место
* партия
* изделие (запас, ТМЦ)
* фактическое количество
* дата учета фактического количества
* признак печати этикетки
* признак брака
* признак неликвида
  + 1. **Вспомогательный бизнес процесс ввода данных инвентаризации вручную**

Данный бизнес процесс предназначен для ввода данных с этикеток старого образца не имеющих штрих кодировки, а также поврежденных этикеток.



Рисунок 1.4.Общая схема вспомогательного бизнес процесса ввода данных вручную.

* + - 1. **Функциональные требования вспомогательного процесса ввода данных инвентаризации вручную**

1. Отдельная команда на главном экране для режима работы приложения ввода данных вручную.
2. Поля место, код запаса, партия доступны для ручного изменения, только после активации команды ввода данных вручную.
3. Команда для ввода фактического количества доступна по окончании заполнения всех полей.
   * 1. **Вспомогательный бизнес процесс для выбора партии из списка партий.**

Данный процесс предназначен для заполнения поля партия путем выбора партии из списка партий текущего ТМЦ по номеру заказа на закупку



Рисунок 1.5.Общая схема вспомогательного бизнес процесса для выбора партий из списка партий.

* + - 1. **Функциональные требования вспомогательного процесса для выбора партий из списка партий**

1. Команда для выбора партии из списка партий.
2. Команда доступна при ручном заполнении поля партия.
3. Команда формирует список партий по предварительно заполненному значению ТМЦ.
4. Список партий должен отображаться на отдельном экране в виде карточек с полями:

* Заказ на закупку
* Партия
* Количество по ИНВ-3
* Единица измерения количества по ИНВ-3
* Учтенное количество
* Единица измерения учтенного количества по ИНВ-3
* Место

1. После выбора партии из списка партий должен открыться диалог для заполнения фактического количества.

* + 1. **Вспомогательный бизнес процесс для выбора места хранения из списка мест указанной партии.**

Данный процесс предназначен для заполнения поля партия путем выбора партии из списка партий текущего ТМЦ по номеру заказа на закупку с конкретного места хранения, выбранного из списка мест указанной партии.



Рисунок 1.6.Общая схема вспомогательного бизнес процесса для выбора места хранения указанной партии.

* + - 1. **Функциональные требования вспомогательного процесса для выбора места хранения указанной партии.**

1. Команда для выбора места хранения указанной партии.
2. Команда должна быть доступна, когда партия хранится на разных местах.
3. Команда формирует список мест по выбранной партии.
4. Список мест должен отображаться на отдельном экране в виде карточек с полями:

* Заказ на закупку
* Партия
* Количество по ИНВ-3
* Единица измерения количества по ИНВ-3
* Учтенное количество
* Единица измерения учтенного количества по ИНВ-3

1. После выбора места из списка мест поле место на главном экране должно заполниться выбранным значением, а соответствующее место в данных инвентаризации обновлено.
2. После выбора места из списка мест должен открыться диалог для заполнения фактического количества.
   * 1. **Вспомогательный бизнес процесс для отсутствующей партии в заказах на циклическую инвентаризацию**.

Данный процесс определяет поведение мобильного приложения при отсутствии партии в заказах на циклическую инвентаризацию.



Рисунок 1.7.Общая схема вспомогательного бизнес процесса для отсутствующей партии.

* + - 1. **Функциональные требования вспомогательного процесса для отсутствующей партии в заказах на циклическую инвентаризацию.**

1. Предоставить выбор из создания новой позиции в ИНВ-3 или определения партии из списка партий по номеру заказа на закупку.
   * 1. **Вспомогательный бизнес процесс для фактического хранения запаса на месте отличном от системного**.

Данный процесс определяет поведение мобильного приложения при несовпадении места хранения по данным Infor LN и фактического.

Рисунок 1.8.Общая схема вспомогательного бизнес процесса в случае несовпадения мест хранения.

* + - 1. **Функциональные требования вспомогательного процесса для фактического хранения запаса на месте отличном от системного.**

1. Когда по данным Info LN партия хранится на одном месте, а сканирование происходит с другого места необходимо предупредить, что партия хранится на другом месте с выводом значения другого места и предложить изменить системное место. При изменении системного места это позиция ИНВ-3 обновляется этим значением, и после должен открыться диалог для заполнения фактического количества. Иначе, переход на главный экран.
2. Когда по системе Infor LN партия храниться на одном конкретном месте, но физически лежит на нескольких местах. И есть необходимость добавить новое место хранения. Необходима проверка, лежит ли запас на данном месте и предупредить о том, что партия хранится на другом месте и задать уточняющий вопрос «Изменить системное место?» В случае добавления нового места хранения необходимо нажать команду «Нет и после должен открыться диалог для заполнения фактического количества.
3. Когда по системе Infor LN партия хранится на многих местах, но отсканированное место не совпадает ни с одним из системы, то в данном случае необходимо предупредить о том, что партия хранится на других местах и задать уточняющий вопрос «Выбрать системное место?». При нажатии на кнопку «Выбрать» открывается список «Места», где отображаются все места, на которых хранится партия в системе Infor LN.

Необходимо выбрать место из имеющихся мест. При нажатии на кнопку «Нет» система оставляет введенное место и переходит в окно ввода количества.

1. После изменения системного места необходимо продолжить работу обычном режиме, ввести количество к учету и сохранить данные инвентаризации.
   1. **Нефункциональные требования.**
      1. Мобильное приложение должно работать в офлайн режиме без доступа к локальной сети и производить интеграцию с производственными системами через файлы импорта / экспорта в соответствии с требованиями служб безопасности предприятия.
      2. Мобильное приложение должно импортировать / экспортировать данные в/из ERP Infor LN
      3. Мобильное приложение должно работать в операционной системе Android 7.1
      4. Мобильное приложение должно работать на терминале сбора данных Honeywell EDA50k-1.
      5. Мобильное приложение должно выполнять функции сканирования и распознавание DataMatrix кодов
      6. Мобильное приложение должно использовать штатный сканер терминала сбора данных Honeywell EDA50k-1.
      7. Мобильному приложению запрещено использовать штатную фотокамеру в соответствии с требованиями служб безопасности предприятия.
      8. Промежуточные результаты инвентаризации должны сохраняться в память устройства в базу данных SQLite.
      9. Заполненные, но не учтенные поля данных инвентаризации необходимо сохранять при выходе из приложения и заполнять последними сохраненными значениями при запуске приложения.
      10. Для импорта/экспорта данных в/из мобильное приложение использовать файлы в формате csv.
      11. Расширяемость и адаптация приложения для реализации новых функциональных требований.
2. **ПРОЕКТИРОВАНИЕ АРХИТЕКТУРЫ**
   1. **Чистая архитектура.**

Основываясь на нефункциональные требования о расширяемости и адаптации для реализации приложения необходимо придерживаться принципов чистой архитектуры, схематично представленных на рисунке 2.1.

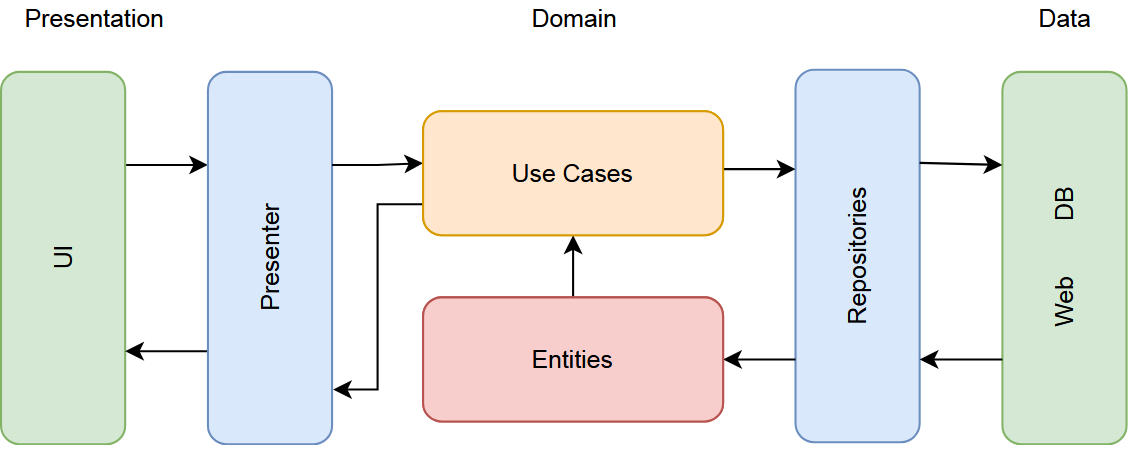


Рисунок 2.1.Clean Architecture

Важно создавать приложения, которые будет легко поддерживать, расширять и развивать в долгосрочной перспективе. Приложения, созданные с помощью принципов чистой архитектуры, обладают высокой устойчивостью к изменениям. Они быстро адаптируются к новым требованиям и возможностям, сохраняя при этом высокое качество кода и производительность. Кроме того, система становится гораздо проще для понимания и сопровождения, что делает ее более доступной для других разработчиков.

Принципы чистой архитектуры также помогают избежать проблем с зависимостями и разделить приложение на логические блоки. Это повышает удобство добавления новых функций и изменения уже существующих. В результате, время разработки сокращается, а приложение становится более гибким и масштабируемым.

Какие преимущества могут дать принципы Clean Architecture:

* Улучшение качества кода;
* Быстрая адаптация к изменениям и новым функциям;
* Удобство сопровождения и расширения приложений;
* Минимизация проблем с зависимостями;
* Повышение гибкости и масштабируемости приложений;
* Сокращение времени разработки и улучшение производительности.1

Принципы чистой архитектуры позволяет разделить приложение на слои. У каждого слоя своя зона ответственности. Обычно выделяют следующие:

* Presentation - отвечает за отображение пользовательского интерфейса и реагирование на его события
* Domain - бизнес-логика, изолированная от деталей реализации, определяет правила и операции, как приложение должно взаимодействовать с данными
* Data - хранилище данных
  1. **Hilt**

Один из важнейших принципов который гласит о том, что стоит использовать общий контракт, такой как интерфейс или абстрактный класс, вместо прямой зависимости слоя верхнего уровня от компонентов слоя нижнего. Таким образом, каждый слой использует этот контракт, что обеспечивает изоляцию изменений в верхнем слое.

В Clean Architecture центральным слоем является domain, тогда можно схематично представить подобную связь зависимостей, рисунок 2.2.:

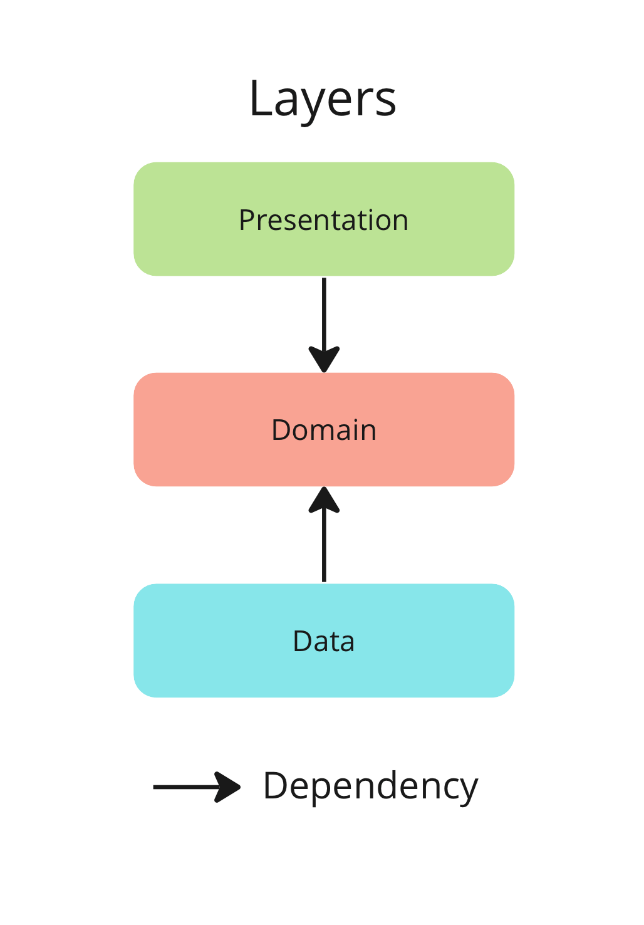


Рисунок 2.2. Связь зависимостей

Инверсия зависимости является архитектурным принципом, для его реализации будет использоваться практическая реализация - [Hilt](https://developer.android.com/training/dependency-injection/hilt-android).4 [Hilt](https://developer.android.com/training/dependency-injection/hilt-android) - это библиотека для внедрения зависимостей, разработанная Google, и предназначена специально для Android-проектов. Она основана на Dagger и предоставляет автоматизированный подход к внедрению зависимостей.

* 1. **MVVM**

MVVM — это паттерн архитектуры, который разделяет приложение на три ключевых компонента: Model, View и ViewModel. MVVM позволяет связывать элементы View со свойствами и событиями ViewModel. При этом ViewModel — абстракция представления.

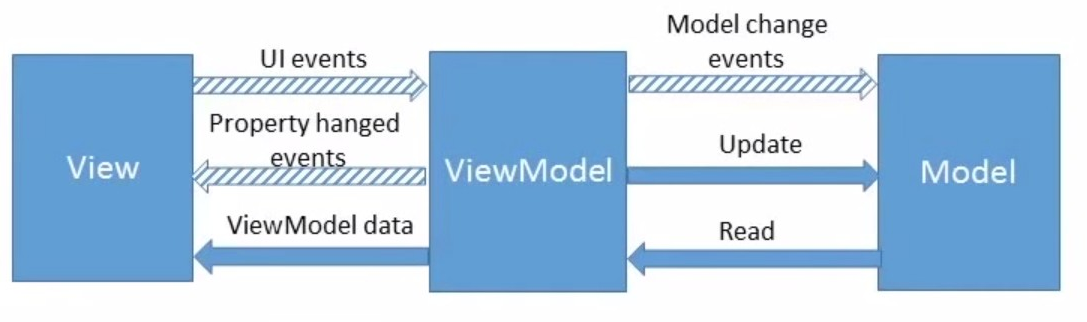


Рисунок 2.3.MVVM

* Model - это бизнес-логика и данные. Здесь обрабатываются данные, выполняются операции и взаимодействуют с источниками данных.
* View - это пользовательский интерфейс (UI) приложения. View ответственен за отображение данных и реагирование на взаимодействие пользователя.
* ViewModel - является посредником между Model и View. Он содержит логику, связанную с UI, и управляет состоянием приложения. ViewModel предоставляет данные, которые View отображает, и обрабатывает действия пользователя.3
  1. **Navigation**

Архитектурный компонент Navigation позволяет упростить реализацию навигации между экранами назначения (destinations) в приложении. По умолчанию, Navigation поддерживает фрагменты (Fragments) и активности (Activities) в качестве экранов назначения, но также можно [добавить поддержку новых типов экранов назначения](https://developer.android.com/topic/libraries/architecture/navigation/navigation-add-new). Набор экранов назначения называется навигационным графом (navigation graph) приложения.6

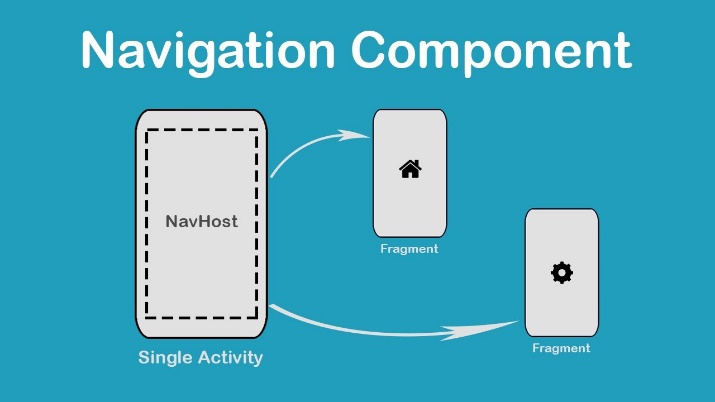


Рисунок 2.4. Navigation Component

* 1. Room

В соответствии с нефункциональными требованиями промежуточные результаты инвентаризации должны храниться в базе данных SQLite в слое Data шаблона чистая архитектура. Так же приложение должно обладать высокой адаптивностью для новой функциональности. Для этих целей оптимально подходит технология Room.



Рисунок 2.5.Room

Room — это способ сохранить данные приложений в Android-приложении, высокоуровневый интерфейс для низкоуровневых привязок SQLite, встроенных в Android. Он выполняет большую часть своей работы во время компиляции, создавая API-интерфейс поверх встроенного SQLite API, поэтому не нужно работать с Cursor или ContentResolver.2

* 1. **DataStore**

В соответствии с нефункциональными требованиями о сохранении состояния полей при входе/выходе из приложения необходимо использовать технологию DataStore, рисунок 2.5.

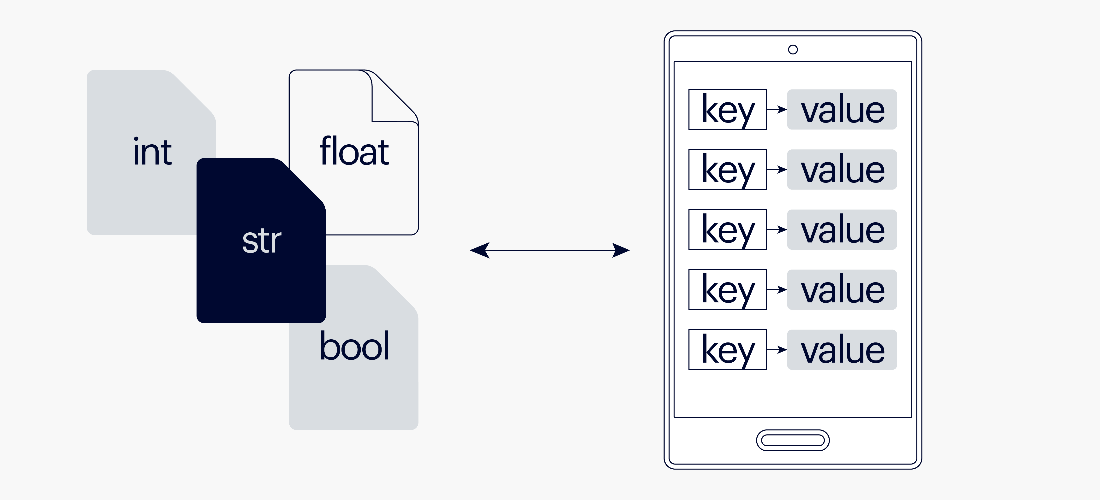


Рисунок 2.6. DataStore

DataStore работает асинхронно. Вся работа с библиотекой выполняется с помощью Kotlin Coroutines и Flow. DataStore позволяет хранить данные двумя способами:

* По принципу «ключ — значение», аналогично SharedPreference.
* Хранить типизированные объекты, основанные на protocol buffers.5
  1. **Архитектура приложения**

Архитектура приложения представлена на рисунке 2.7.

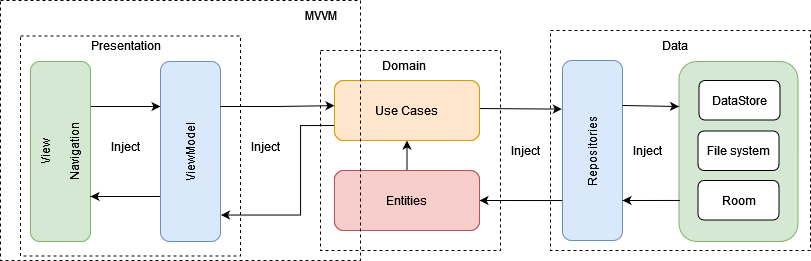


Рисунок 2.7.

* 1. Используются принципы Clean Architecture.
  2. В слое Data используются следующие технологии и инструменты:
* Room – для хранения промежуточных данных циклической инвентаризации
* DataStore – для сохранения значений полей склад, место, изделие, партия при входе/выходе в/из приложения.
* File system – для импорта/экспорта данных циклической инвентаризации

1. В слое Presentation используется реализация паттерна MVVM:

* View – пользовательский интерфейс
* ViewModel – логика работы пользовательского интерфейса
* Navigation –навигация между экранами

1. Связи между слоями:

* Hilt – внедрение зависимостей

1. **ПРОГРАММНАЯ РЕАЛИЗАЦИЯ**
   1. **Среда разработки и система сборки проекта.**

Популярные среды разработки для Android приложений это Eclipse (Eclipse Foundation), IntelliJ IDEA (JetBrains), Android Studio (JetBrains). Все эти среды предоставляют необходимые возможности для разработки. У каждой есть свои отличительные особенности. Для реализации данного приложения предпочтение отдается на основании удобства и привычки. Выбор был сделан в сторону Android Studio (JetBrains).

Что касается системы сборки проектов, то необходимо следовать рекомендациям Google из которых следует, что начиная с Android Studio Giraffe Kotlin DSL становится новым стандартом для Gradle-скриптов в разработке Android.

* 1. **Зависимости**.

После инициализации проекта устанавливаются необходимые зависимости:

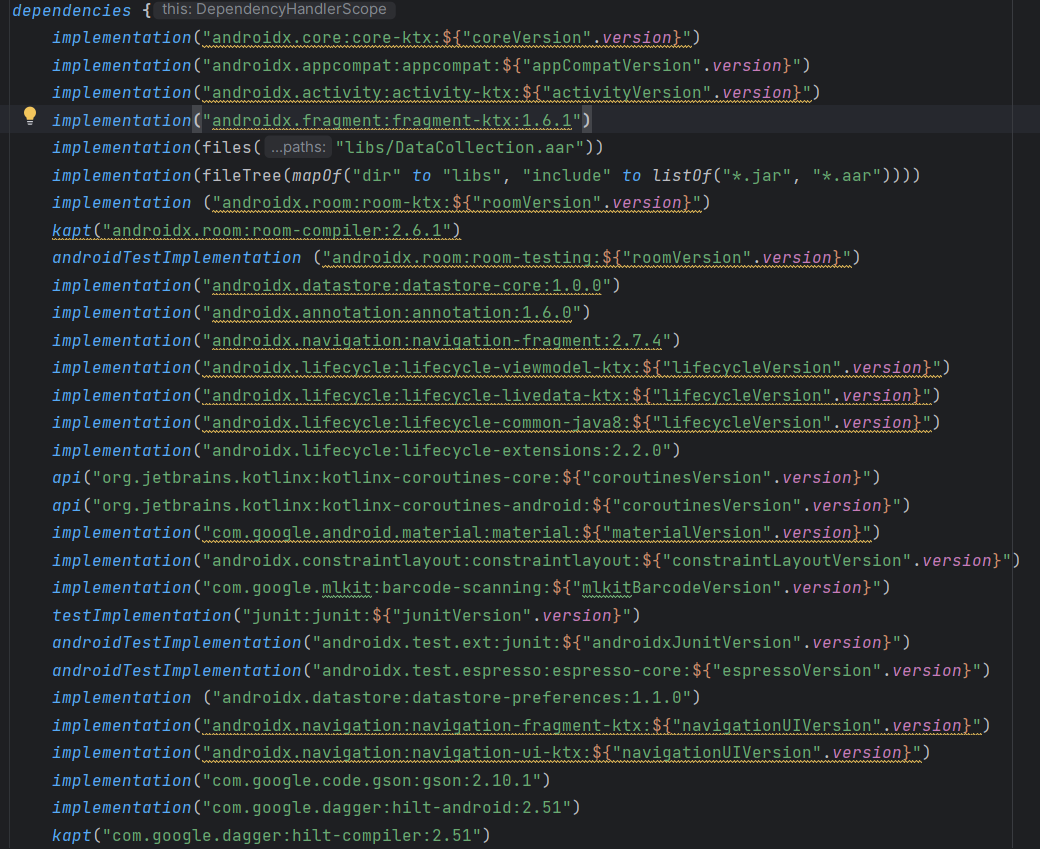


Рисунок 3.1. Зависимости

* 1. **Создание структуры проекта.**

Структура папок проекта, представленная на рисунке 3.2, вносит определенный порядок в проект, позволяющий быстро ориентироваться и организовывать работу с проектом.

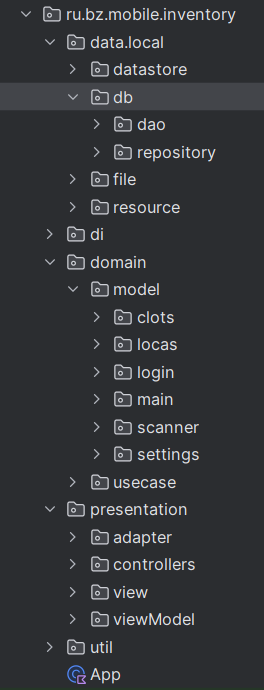


Рисунок 3.2. Структура проекта

* 1. **База данных.**

База данных предназначена для хранения промежуточных данных циклической инвентаризации. Состоит из одной таблицы «whinh501». Название и структура данных были выбраны по аналогии с соответствующей таблицей «whinh501» ERP Infor LN.

Поля таблицы «whinh501»:

* cwar – код склада (String)
* loca – место (String)
* item – код изделия (запас, ТМЦ) (String)
* clot – код партии (String)Ы
* orno – номер заказа на циклическую инвентаризацию (String)
* pono – строка заказа на циклическую инвентаризацию (Long)
* qstr – количество запаса по данным ERP Infor LN (Double)
* qnty – количество учтенного фактического запаса (Double)
* utcDate – дата и время учета запаса (Long)
* stkr – признак печати этикетки (Long)
* unit – единица измерения запаса (String)
* lock – признак брака (Long)
* illiquid – признак неликвида (Long)
* porn – номер заказ на закупку позиции (String)
* sloca – системное место (String)

Таблица и поля отображаются в виде сущности данных в data class IOP:



Рисунок 3.3. Entity

Объекты доступа к данным (DAO), предоставляющие методы, которые приложение может использовать для запросов, обновления и удаления в базе данных для удобства разделены на три интерфейса IOPDao, ClotDao, LocaDao:



Рисунок 3.4. IOPDao

Класс AppDatabase определяет конфигурацию базы данных и служит основной точкой доступа приложения к сохраненным данным:

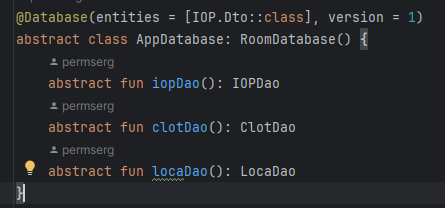


Рисунок 3.5. AppDatabase

Базовую настройку RoomDB необходимо предоставить как зависимость к остальным компонентам приложения. Для этого в пакете di создается RoomModule:

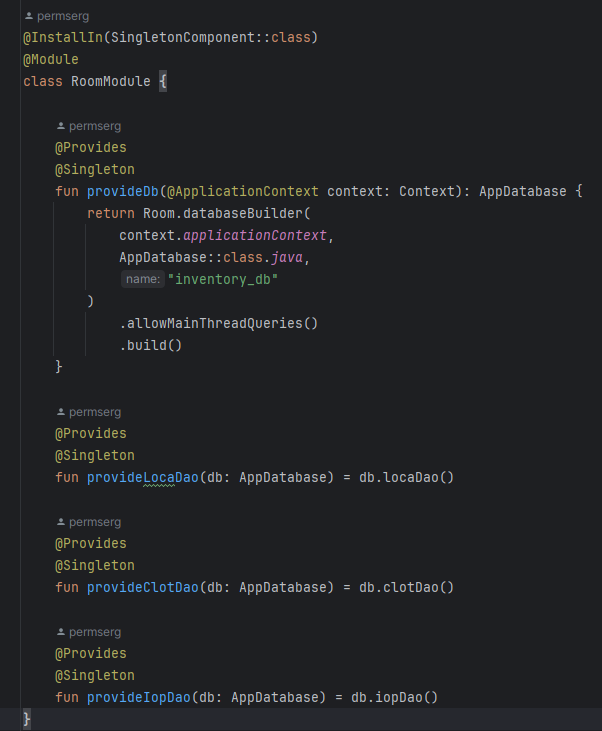


Рисунок 3.6. RoomModule

В соответствии с принципами чистой архитектуры необходимо создать репозиторий для доступа к данным базы данных и внедрить зависимость базовой настройки RoomDB. Для удобства этот репозиторий разделен на три MainRepository, ClotRepository, LocaRepository:

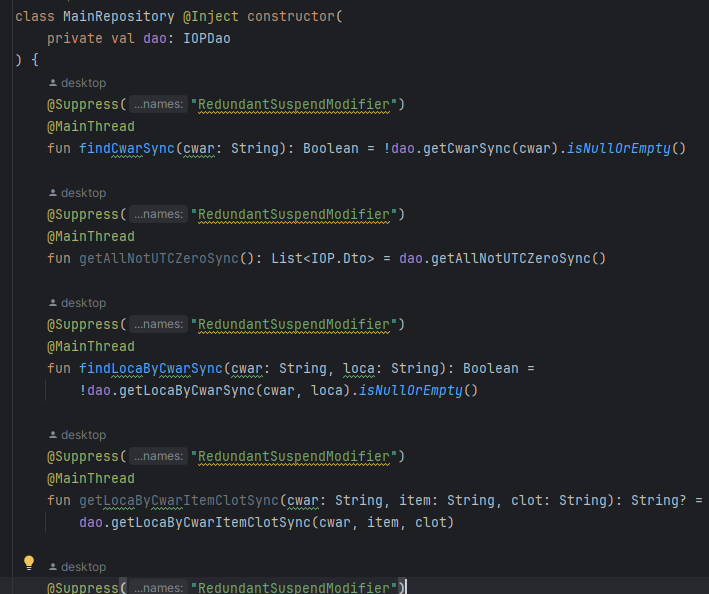


Рисунок 3.7. MainRepository

* 1. **Верстка интерфейса**.

В соответствии с функциональными требованиями, определяющими набор полей, экранов и их взаимодействие необходимо создать соответствующие фрагменты и ресурсы:

* fragment\_main – главный фрагмент для реализации интерфейса пользователя основного бизнес процесса

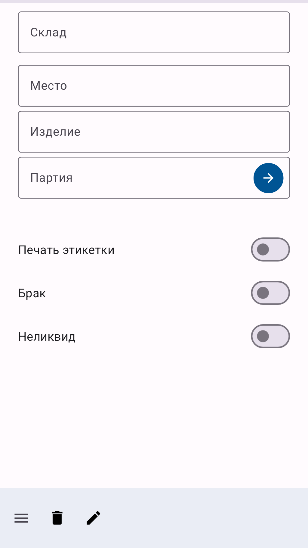


Рисунок 3.8. fragment\_main.

* fragment\_clots, fragment\_clots\_recycler, item\_clot – набор фрагментов для реализации RecyclerView из карточек партий вспомогательного бизнес процесса выбора партий из списка партий

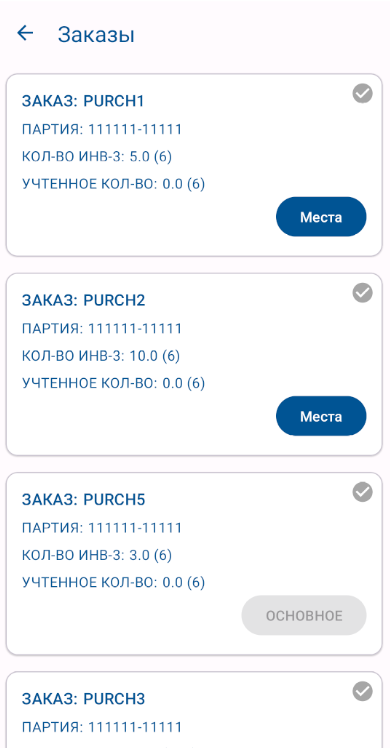


Рисунок 3.9. Список партий.

* fragment\_locas, fragment\_locas\_recycler, item\_loca – набор фрагментов для реализации RecyclerView из карточек мест вспомогательного бизнес процесса выбора мест хранения из списка мест указанной партии.

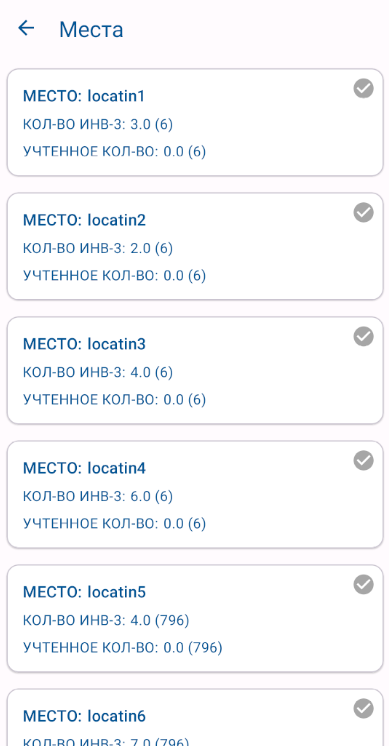


Рисунок 3.10. Список мест.

* save\_input – фрагмент для диалога ввода фактического количества

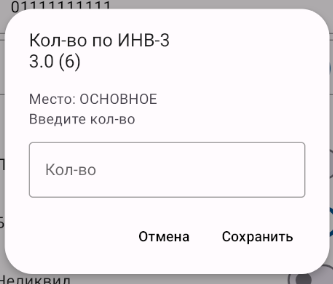


Рисунок 3.11. Диалог для ввода фактического количества.

* fragment-login – фрагмент для ввода пароля для доступа к импорту/экспорту данных инвентаризации

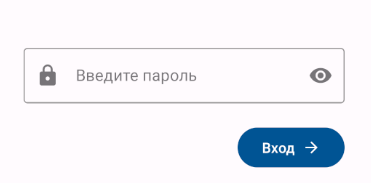


Рисунок 3.12. Фрагмент ввода пароля.

* fragment\_settings – фрагмент для импорта/экспорта данных инвентаризации

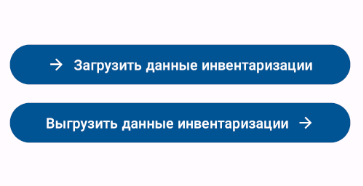


Рисунок 3.13. Фрагмент для импорта/экспорта данных инвентаризации.

* fragment\_bottom\_sheet\_menu\_dialog, header\_navigation\_drawer – группа фрагментов для нижнего меню

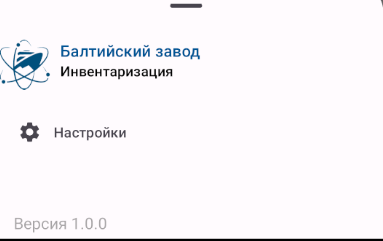


Рисунок 3.14. Нижнее меню.

* nav-graph – граф, ресурс, который содержит всю связанную с навигацией информацию для списка фрагментов

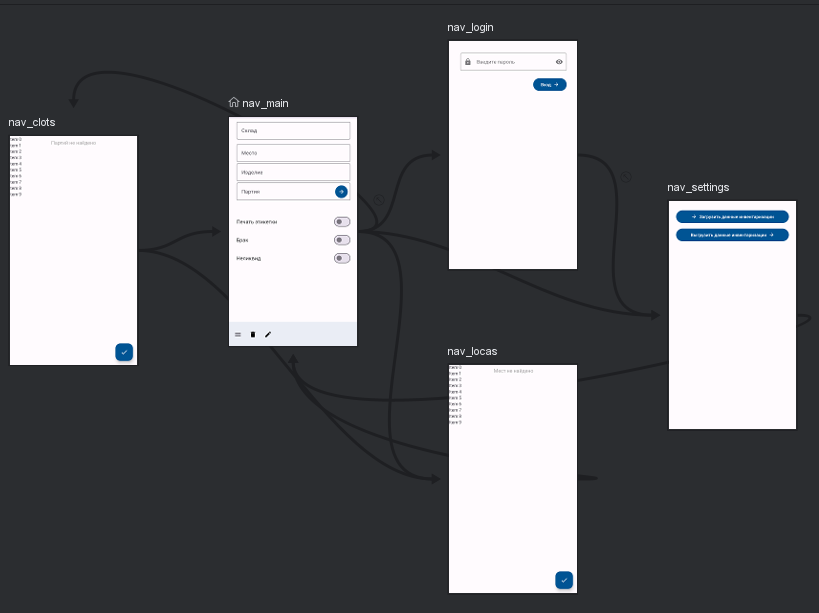


Рисунок 3.15. nav-graph.

* 1. **Файловые ресурсы**

В соответствии с функциональными требованиями необходим файловый репозиторий для импорта/экспорта данных инвентаризации. Для этого в слое Data необходимо создать FileRepository с методами записи/чтения в/из файла.

Файловый репозиторий необходимо предоставить как зависимость к остальным компонентам приложения. Для этого в пакете di создается FileModule:

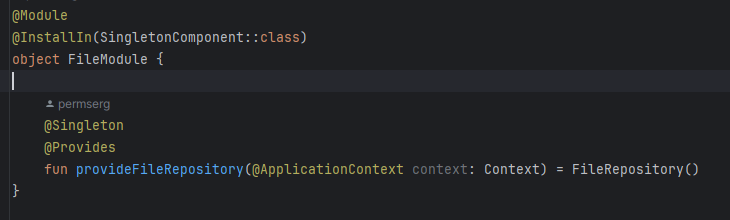


Рисунок 3.16. FileModule.

* 1. **DataStore ресурсы.**

В соответствии с требованиями необходима функциональность для сохранения значений в полях пользовательского интерфейса при входе/выходе из приложения. Для этого необходимо воспользоваться библиотекой DataStore и в слое Data создать DataStoreRepository с методами чтения/записи необходимых объемов данных. Данные определяются классом domain/model/main/MainModel объекты которого отражают состояния полей фрагмента MainFragment. В зависимости от состояния жизненного цикла фрагмента данные сериализуются/десериализуются и сохраняются/считываются в/из DataStore.

DataStore репозиторий необходимо предоставить как зависимость к остальным компонентам приложения. Для этого в пакете di создается DataStoreModule:

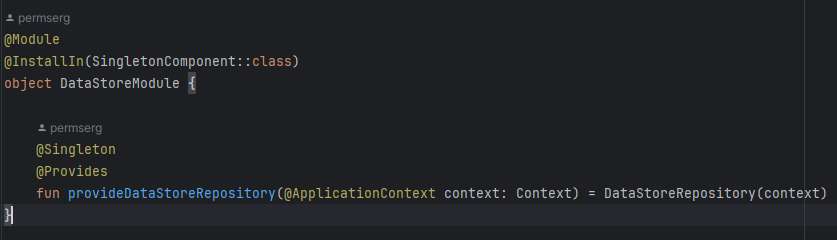


Рисунок 3.17. DataStoreModule.

* 1. **Barcode**

В соответствии с требованиями заполнение полей фрагмента MainFragment необходимо производить через штатный сканер терминала сбора данных Honeywell EDA50k-1. Для этого необходимо воспользоваться поставляемыми вместе с Honeywell EDA50k-1 документацией и библиотеками. В соответствии с документацией необходимо создать BarcodeController, где реализуются настройки сканера, жизненного цикла и слушатели.

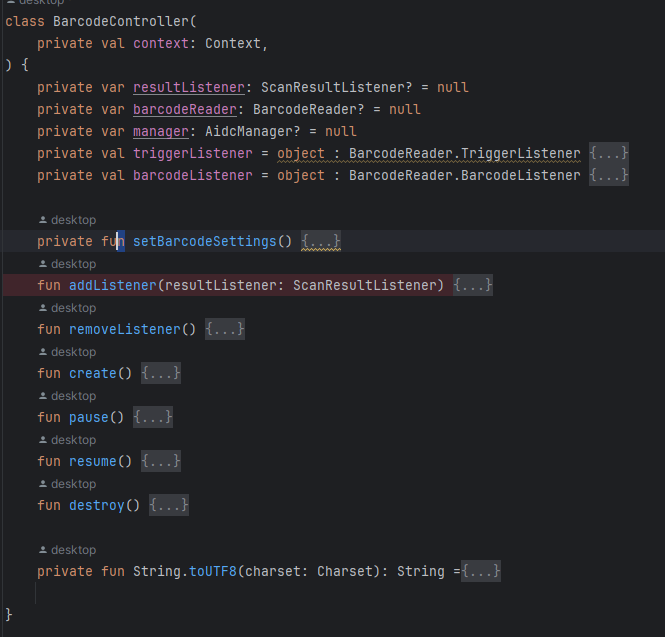


Рисунок 3.18. BarcodeController

BarcodeController необходимо предоставить как зависимость к остальным компонентам приложения. Для этого в пакете di создается BarcodeModule:

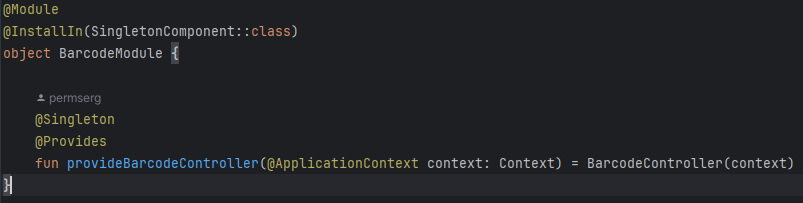


Рисунок 3.19. BarcodeController

* 1. **Use cases**

В соответствии с принципами чистой архитектуры в domain слое необходимо создать use cases, которые будут разделять функции репозитория (SOLID, interface segregation). Следуя рекомендациям от Google, что каждый класс должен отвечать только за одну функциональность, разделим их на:

* MainUseCase – бизнес логика для MainViewModel
* ClotUseCase – бизнес логика для формирования списка партий для ClotsViewModel
* LocaUseCase - бизнес логика для формирования списка мест для LocasViewModel
* DataStoreUseCase – бизнес логика сохранения значений полей для MainViewModel
* FileUseCase - бизнес логика чтения/сохранения данных инвентаризации для SettingsViewModel
* ResourcesUseCase – для предоставления ресурсов из strings.xml
* SettingsUseCase – бизнес логика создания/удаления данных из базы данных при импорте/экспорте данных инвентаризации
  1. **ViewModel**

Логику связанную с пользовательским интерфейсом необходимо реализовать в соответствующих классах ViewModel:

* MainViewModel – реализуется логика заполнения или изменения полей фрагмента MainFragment данными в соответствии с функциональными требованиями через штатный сканер терминала сбора данных или ручным вводом. Состояния полей сохраняются в базу данных или в DataStore для чего в MainViewModel внедряются зависимости MainUseCase, DataStoreUseCase.
* ClotsViewModel – реализуется логика для построения списка партий с соответствующими полями на основе функциональных требований. Для построения и отображения списка используется технология RecyclerView с созданием соответствующего адаптера ClotAdapter. Получение данных для списка реализуется через внедренную зависимость ClotUseCase.
* LocasViewModel – реализуется логика для построения списка мест с соответствующими полями на основе функциональных требований. Для построения и отображения списка используется технология RecyclerView с созданием соответствующего адаптера LocaAdapter. Получение данных для списка реализуется через внедренную зависимость LocaUseCase.
* SettingsViewModel - реализуется логика обработки команд импорта/экспорта данных из файла и последующим обновлением состояния базы данных для чего в SettingsViewModel внедряются соответствующие зависимости SettingsUseCase и FileUseCase.
* LoginViewModel – реализуется логика обработки поля пароль в соответствии с функциональными требованиями.

1. **ТЕСТИРОВАНИЕ**

Для функционального тестирования работы составлены тест-кейсы, представленные в таблице 4.1.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№ тест-кейса** | **Заголовок** | **Предусловия** | **Шаги проверки** | **Ожидаемый результат** |
| 1 | Замена системного места на фактическое путем сканирования данных | матрикс-код склада | 1 Сканирование склада | 1 Проверка на валидность кода склада |
| матрикс-код места | 2 Сканирование места | 2 Проверка на валидность кода места |
| матрикс-код изделия | 3 Сканирование изделия | 3 Проверка на валидность кода изделия |
| матрикс-код партии | 4 Сканирование партии | 4 Проверка на валидность кода партии |
| количество изделий | 5 Предложение ввода количества | 5 Ввод количества |
|  | 6 Ввод количества | 6 Сохранение в БД |
| 2 | Замена системного места на фактическое путем ввода данных вручную | матрикс-код склада | 1 Сканирование склада (или по умолчанию стоит выбранный склад) | 1 Проверка на валидность кода склада |
| код места | 2 Нажать на иконку "карандаш" (кнопка ручного ввода и изменения данных) | 2 Поля для записи изменений становятся доступны |
| код изделия | 3 Заполнение данных по полям "место"," изделие", "партия" | 3 Проверка на валидность кодов места, изделия, партии |
| код партии | 4 Предложение ввод количества | 4 Ввод количества |
| количество изделий | 5 Ввод количества | 5 Сохранение в БД |
| 3 | Замена системного места на фактическое путем гибридного заполнения полей (скан и ручной ввод) | матрикс-код склада | 1 Сканирование склада (или по умолчанию стоит выбранный склад) | 1 Проверка на валидность кода склада |
| код (матрикс-код) места | 2 Сканирование данных | 2 Проверка на валидность сканированных данных |
| код (матрикс-код) изделия | 3 Нажать на иконку "карандаш" (кнопка ручного ввода и изменения данных) | 3 Поля для записи изменений становятся доступны |
| код (матрикс-код) партии | 4 Редактирование данных по полям "место"," изделие", "партия" | 4 Проверка на валидность кодов места, изделия, партии |
| количество изделий | 5 Предложение ввод количества | 5 Ввод количества |
|  | 6 Ввод количества | 6 Сохранение в БД |
| 4 | Замена системного места на фактическое путем ввода данных вручную без привязки к заказу | матрикс-код склада | 1 Сканирование склада (или по умолчанию стоит выбранный склад) | 1 Проверка на валидность кода склада |
| код места | 2 Сканирование данных | 2 Проверка на валидность введенных данных |
| код изделия | 3 Нажать на иконку "карандаш" (кнопка ручного ввода и изменения данных) | 3 Поля для записи изменений становятся доступны |
| код партии, которая не привязана к заказу | 4 Заполнение данных по полям "место"," изделие", "партия" | 4 Проверка на валидность кодов места, изделия, партии |
| количество изделий | 5 Предложение выбора известной партии (или оставить введенную вручную) | 5 Нажать "Выбрать", чтоб выбрать партию (или нажать на кнопку "Нет", чтобы оставить заполненные данные) |
|  | 6 Предложение ввод количества | 6 Ввод количества |
|  | 7 Ввод количества | 7 Сохранение в БД |
| 5 | Сканирование данных в режиме редактирования | данные об изделии | 1 Сканирование склада (или по умолчанию стоит выбранный склад) | 1 Проверка на валидность кода склада |
|  | 2 Нажать на иконку "карандаш" (кнопка ручного ввода и изменения данных) | 2 Поля для записи изменений становятся доступны |
|  | 3 Сканирование данных | 3 Сканируемые данные не заполняются в поля |
| 6 | Загрузка данных из файла | файл с данными об изделиях | 1 Нажать на кнопку меню (три полоски) | 1 Открывается bottom sheet |
|  | 2 На всплывающем меню нажать кнопку "Настройки" | 2 Открывается окно для авторизации |
|  | 3 Предложение авторизации путем ввода пароля | 3 Ввод пароля |
|  | 4 Нажать кнопку "Загрузить данные инвентаризации" | 4 Открывается файловый каталог устройства |
|  | 5 Выбрать файл из памяти устройства с данными для инвентаризации | 5 Данные успешно загружены |
| 7 | Выгрузка данных в файл | БД с данными после инвентаризации | 1 Нажать на кнопку меню (три полоски) | 1 Открывается bottom sheet |
|  | 2 На всплывающем меню нажать кнопку "Настройки" | 2 Открывается окно для авторизации |
|  | 3 Предложение авторизации путем ввода пароля | 3 Ввод пароля |
|  | 4 Нажать кнопку "Загрузить данные инвентаризации" | 4 Выводится сообщение о выгрузке данных |
|  | 5 Выбрать файл из памяти устройства с данными для инвентаризации | 5 Данные успешно выгружены |
| 8 | Удаление внесенных изменений | данные об изделии | 1 Ввести данные (или отсканировать) | 1 Данные заполнены |
|  | 2 Нажать на кнопку "удалить" | 2 Вывод информационного сообщения с подтверждением удаления |
|  | 3 Подтвердить удаление | 3 Данные очищены |
| 9 | Изменение системных параметров на фактические путем сканирования данных | матрикс-код склада | 1 Сканирование склада | 1 Проверка на валидность кода склада |
| матрикс-код места | 2 Сканирование места | 2 Проверка на валидность кода места |
| матрикс-код изделия | 3 Сканирование изделия | 3 Проверка на валидность кода иделия |
| матрикс-код партии | 4 Сканирование партии | 4 Проверка на валидность кода партии |
| количество изделий | 5 Предложение ввода количества | 5 Выбор дальнейшего действия (отмена) |
|  | 6 Отмена ввода количества путем нажатия на кнопку "Отмена" | 6 Окно для ввода количества скрывается |
|  | 7 Нажать на иконку "карандаш" для изменения параметров | 7 Поля для редактирования становятся доступными |
|  | 8 Заполнение и сохранение путем нажатия на кнопку "Сохранить" | 8 Сохранение введенной информации |
|  | 9 Предложение ввода количества | 9 Ввод количества |
|  | 10 Ввод количества | 10 Сохранение в БД |
| 10 | Выбор другой партии | данные об изделии с разными партиями | 1 Сканирование данных (или занесение вручную через кнопку "карандаш") | 1 Проверка на валидность данных об изделии |
|  | 2 Отмена ввода количества путем нажатия кнопки "Отмена" | 2 Поля с данными заполнены |
|  | 3 Нажать на кнопку "стрелки" в поле партии | 3 Открывается окно с выбором партии, которая относится к изделию |
|  | 4 Выбрать партию | 4 Данные о партии записываются в поле |
|  | 5 Предложение ввода количества | 5 Ввод количества |
|  | 6 Ввод количества | 6 Сохранение в БД |
| 11 | Выбор другой партии без загруженных данных о партии | данные об изделии с неизвестной системе партией | 1 Сканирование данных (или занесение вручную через кнопку "карандаш") | 1 Проверка на валидность данных об изделии |
|  | 2 Отмена ввода количества путем нажатия кнопки "Отмена" | 2 Поля с данными заполнены |
|  | 3 Нажать на иконку "карандаш" | 3 Поля для редактирования становятся доступными |
|  | 4 Заполнить данные партии | 4 Проверка на валидность, если успешно, то в БД партий добавляется новая ветка партий |
|  | 5 Сохранить изменения | 5 Сохранение введенной информации |
|  | 6 Предложение ввода количества | 6 Ввод количества |
|  | 7 Ввод количества | 7 Сохранение в БД |
| 12 | Выбор определенного действия с изделием(check boxes) | данные об изделии | 1 Выбрать нужные check boxes | 1 Проверка на валидность данных об изделии |
|  | 2 Сканирование данных (или занесение вручную через кнопку "карандаш") | 2 Поля с данными заполнены |
|  | 4 Сохранить изменения путем нажатия кнопки "Сохранить" | 3 Сохранение записанных данных |
|  | 5 Предложение ввода количества | 4 Ввод количества |
|  | 6 Ввод количества | 5 Сохранение в БД |
| 13 | Выбор определенного действия с изделием(check boxes) | данные об изделии | 1 Сканирование данных (или занесение вручную через кнопку "карандаш") | 1 Проверка на валидность данных об изделии |
|  | 2 Отмена ввода количества путем нажатия кнопки "Отмена" | 2 Поля с данными заполнены |
|  | 3 Выбрать нужные check boxes | 3 Отображение выбранных check boxes |
|  | 4 Сохранить изменения путем нажатия кнопки "Сохранить" | 4 Сохранение записанных данных |
|  | 5 Предложение ввода количества | 5 Ввод количества |
|  | 6 Ввод количества | 6 Сохранение в БД |
| 14 | Добавление нового места | данные об изделии с неизвестным системе местом | 1 Сканирование данных (или занесение вручную через кнопку "карандаш") | 1 Проверка на валидность данных об изделии |
|  | 2 Отмена ввода количества путем нажатия кнопки "Отмена" | 2 Поля с данными заполнены |
|  | 3 Нажать на иконку "карандаш" | 3 Поля для редактирования становятся доступными |
|  | 4 Заполнить данные места | 4 Проверка на валидность, если успешно, то в БД партий добавляется новая ветка места (в выборе определенной партии добавляется кнопка выбора места, чтобы выбрать другое место нужно нажать на кнопку "места" -> выбрать место) |
|  | 5 Сохранить изменения | 5 Сохранение введенной информации |
|  | 6 Предложение ввода количества | 6 Ввод количества |
|  | 7 Ввод количества | 7 Сохранение в БД |
| 15 | Добавление нового изделия | данные об изделии | 1 Сканирование данных (или занесение вручную через кнопку "карандаш") | 1 Проверка на валидность данных об изделии |
|  | 2 Отмена ввода количества путем нажатия кнопки "Отмена" | 2 Поля с данными заполнены |
|  | 3 Нажать на иконку "карандаш" | 3 Поля для редактирования становятся доступными |
|  | 4 Заполнить данные изделия | 4 Проверка на валидность, если успешно, то в новой ветке партии должныдобавиться данные об изделии |
|  | 5 Сохранить изменения | 5 Сохранение введенной информации |
|  | 6 Предложение ввода количества | 6 Ввод количества |
|  | 7 Ввод количества | 7 Сохранение в БД |

Тестирование представляет из себя проверку работы всего списка тест-кейсов, представленных таблице 4.1. Любое несоответствие полученного результата ожидаемому запускает процесс отладки и исправления ошибок работы приложения.

Весь представленный список тест-кейсов прошел функциональное тестирование, и полученные результаты в точности соответствуют ожидаемым результатам.

**ЗАКЛЮЧЕНИЕ**

Цель данной работы — разработать прототип мобильного приложение на базе операционной системы Android для терминала сбора данных Honeywell EDA50k-1, реализующее задачи инвентаризации складских запасов АО «Балтийский завод».

Следующие подзадачи реализованы:

- получен и оформлен список функциональных требований;

- получен и оформлен список нефункциональных требований;

- спроектирована архитектура;

- подобраны необходимые инструменты;

- выполнена программная реализация;

- проведено функциональное тестирование;

Разработка прототипа мобильного приложения, реализующее задачи инвентаризации складских запасов АО «Балтийский завод» успешно завершена.

Мобильное приложение готово для тестовой эксплуатации в АО «Балтийский завод»

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Clean Architecture. [Электронный документ]. (https://habr.com/ru/companies/otus/articles/732178/). Проверено 15.04.2024.
2. Room: Хранение данных на Android для всех и каждого [Электронный документ]. (https://habr.com/ru/articles/336196/). Проверено 15.04.2024.
3. Простая архитектура с использованием MVVM и делегатов в Android. Оптимальное решение для малых проектов [Электронный документ]. (https://habr.com/ru/articles/776344/). Проверено 16.04.2024.
4. Просто об архитектуре в Android. [Электронный документ]. (https://teletype.in/@javalib/3tb9u5Kip4d). Проверено 16.04.2024.
5. Обзор DataStore Library. Прощаемся с SharedPreference? [Электронный документ]. (https://habr.com/ru/companies/tinkoff/articles/525010/). Проверено 16.04.2024.
6. Реализация навигации в Android приложениях с помощью Navigation Architecture Component. [Электронный документ] (https://habr.com/ru/articles/416025/) Проверено 16.04.2024.

**Приложение 1**

Этикетки и DataMatrix коды

1. DataMatrix код склада



Рисунок 1 DataMatrix код склада «W3333»

1. DataMatrix код места



Рисунок 2 DataMatrix код места «ОСНОВНОЕ»



Рисунок 3 DataMatrix код места «locatin3»



Рисунок 4 DataMatrix код места « locatin12»

1. Этикетка ТМЦ



Рисунок 5 Этикетка ТМЦ – изделие 01111111111, партия 111111-11111

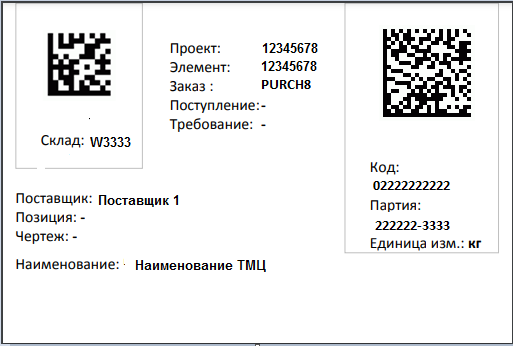


Рисунок 6 Этикетка ТМЦ – изделие 02222222222, партия 222222-3333

**Приложение 2**

**Пример файла импорта**

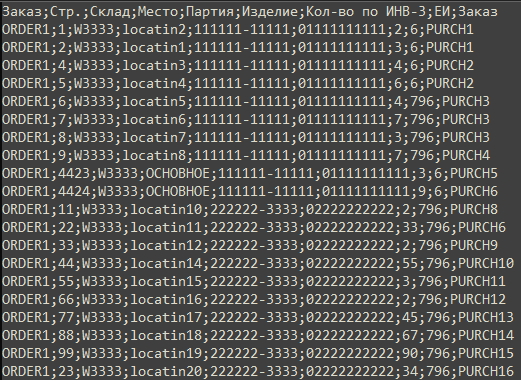


Рисунок 1 Пример файла импорта

**Приложение 3**

**Пример файла экспорта**



Рисунок 1 Пример файла экспорта

**Приложение 4**

Рабочая инструкция по использованию мобильного приложения «Инвентаризация»

Термины и сокращения

Таблица 1

Термины и сокращения

|  |  |
| --- | --- |
| **Общие термины** | |
| ТСД | Терминал сбора данных. |
| ИНВ-3 | Инвентаризационная опись товарно-материальных ценностей |
| Система Infor LN | Информационная система управления на основе Infor LN. |

### Сканирование данных с актуальной этикетки

### Спецификация

Таблица 2

Спецификация

|  |  |
| --- | --- |
| Условия выполнения | Данный шаг выполняется при сканировании напечатанной этикетки из системы Infor LN |
| Исполнитель | Оператор склада |
| Вход | Распечатана актуальная этикетка на запас ТМЦ |
| Выход | Введены данные с этикетки в ТСД |

#### Назначение

Данный шаг предназначен для ввода данных с этикетки в мобильное приложение ТСД для проведения инвентаризации.

#### Последовательность действий:

1. Запустить мобильное приложение на ТСД.

Для этого необходимо найти иконку приложения «Инвентаризация» на рабочем экране ТСД (Рисунок 1)



Рисунок 1

Открывается главный экран приложения. (Рисунок 2)

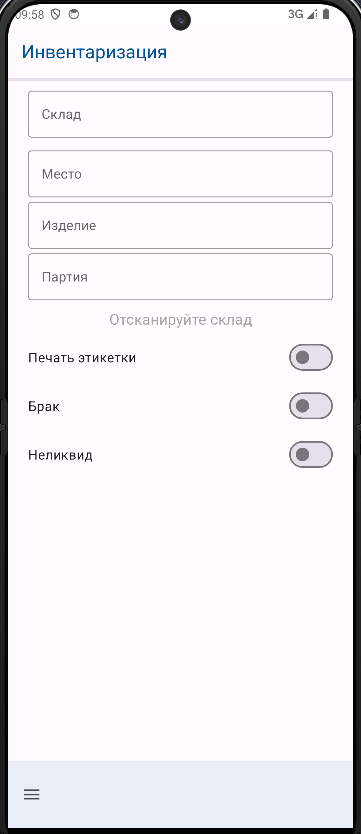


Рисунок 2 Главный экран приложения «Инвентаризация»

1. В открывшемся окне необходимо заполнить поля данными с этикетки путем сканирования:

* Склад
* Место
* Изделие
* Партия

Сканер находится в верхней части ТСД, для сканирования необходимо зажать зеленую кнопку «SCAN» на клавиатуре и навести на штрих код на этикетке.

ТСД сканирует этикетки, распечатанные из системы Infor LN:

1. Большая этикетка два кода (Data Matrix) (Рисунок 3)

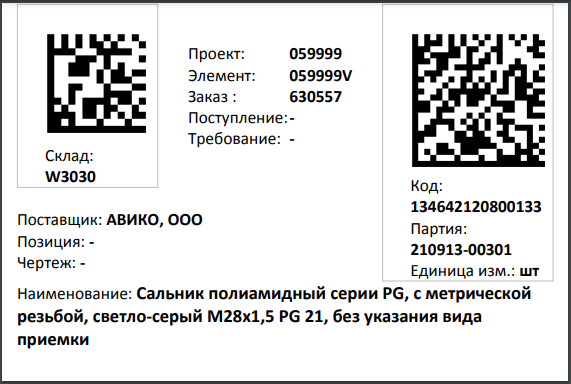


Рисунок 3 Большая этикетка

* 1. Сканирование этикетки начинают с заполнения поля «Склад»

Примечание: Поле сканируется 1 раз в начале инвентаризации и остается заполненным до тех пор, пока не появится необходимость изменить склад

* 1. Далее сканируется штрих код «Места» (складские места, которые имеют номера, имеют свои отдельные штрих-коды) (Рисунок 4)

Примечание: поле место остается неизменным пока не отсканировать другое место хранения (полку, ячейку)

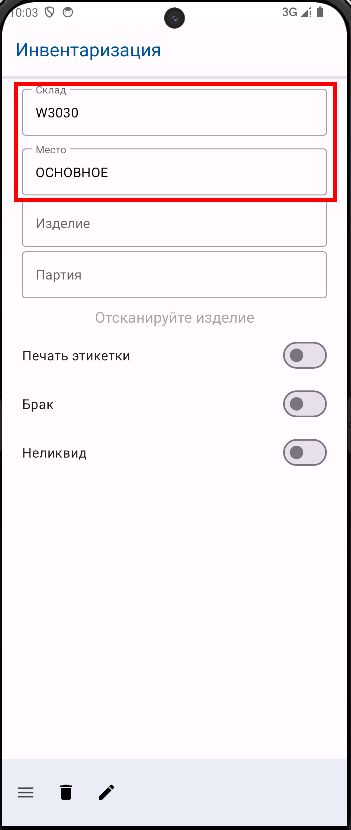


Рисунок 4 Поля, заполненные отсканированными данными

* 1. Далее необходимо отсканировать и заполнить данными поля «Изделие» и «Партия» (Рисунок 5)

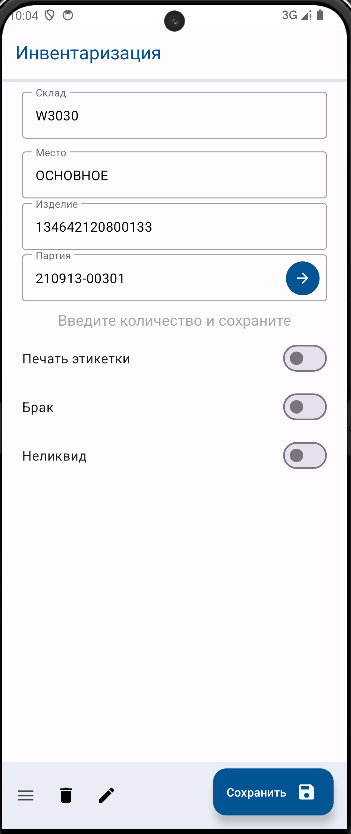


Рисунок5 Поля, заполненные отсканированными данными

После того, как данные с этикетки будут считаны, на экране мобильного приложения автоматически откроется окно с вводом кол-ва.

1. Ввод посчитанного кол-ва ТМЦ к учету осуществляется в окне «Кол-во» (Рисунок 6)

На экране отображается:

* Кол-во по инвентаризационной ведомости из системы Infor LN (Кол-во по ИНВ-3)
* Единицы измерения (796, 166, 006 и тд)
* Место, по которому идет учет
* Поле кол-во, которое необходимо заполнить полученными данными инвентаризации

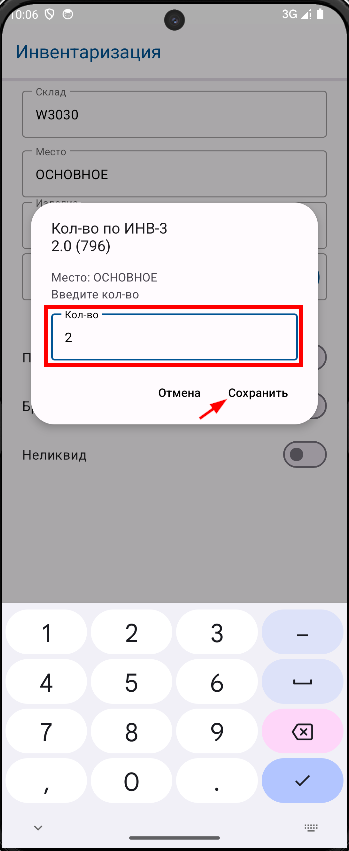


Рисунок 6 Ввод кол-ва к учету

После ввода кол-ва необходимо нажать на кнопку «Сохранить».

1. После сохранения данных на экране появится сообщение «Запись успешно сохранена». (Рисунок 7)

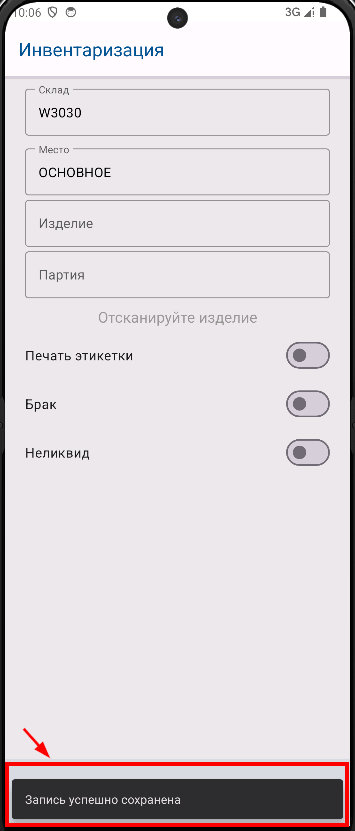


Рисунок 7 Сообщение «Запись успешно сохранена»

Данные инвентаризации внесены в базу данных.

### Ввод данных инвентаризации вручную в ТСД

#### Спецификация

Таблица 3

Спецификация

|  |  |
| --- | --- |
| Условия выполнения | Данный шаг выполняется при отсутствии актуальной этикетки |
| Исполнитель | Оператор склада |
| Вход | Распечатана этикетка из BAAN или рукописная этикетка |
| Выход | Введены данные инвентаризации в ТСЦ |

#### Назначение

Данный шаг предназначен для ввода данных с этикетки старого образца, этикеток плохого качества или же ручных этикеток в мобильное приложение ТСД для проведения инвентаризации.

#### Последовательность действий

1. Запустить мобильное приложение на ТСД.
2. На главном экране нажать на команду  «Карандаш» (ручной ввод данных) (Рисунок 8)

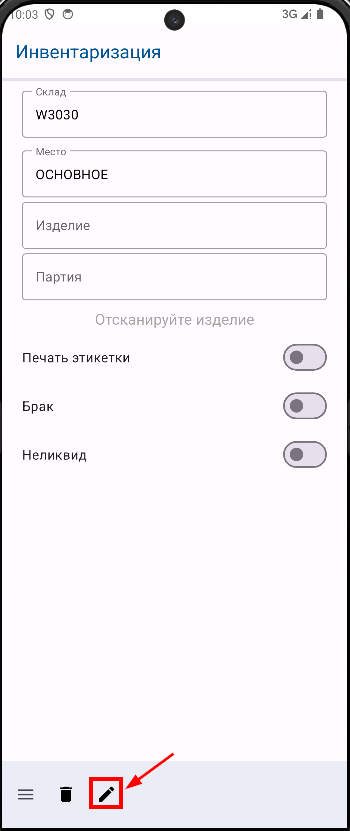


Рисунок 8 Главный экран приложения «Инвентаризация», вызов команды «Ручной ввод»

1. В открывшемся окне необходимо заполнить поля вручную через клавиатуру на ТСД или через виртуальную клавиатуру на экране приложения.

Для редактирования доступны поля: (Рисунок 9)

* Место
* Изделие
* Партия

ВАЖНО! Есть возможность заполнять или менять данные только в определенных полях по отдельности или заполнять все данные ручную



Рисунок 9 Изменение и сохранение данных введенных вручную

После заполнения полей данным необходимо нажать на кнопку «Изменить».

Примечание: При ошибке ввода изделия (например: введены лишние символы, более 15 символов) после сохранения выйдет предупреждение «Неправильное изделие» (Рисунок 10)

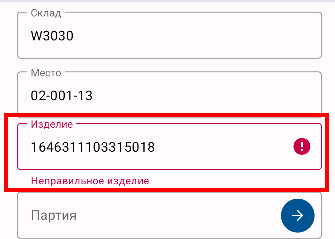


Рисунок10 Предупреждение

1. Выбор партии из заказа на инвентаризацию системы Infor LN.

Если не заполнять поле «Партия» вручную, то можно воспользоваться выбором партии из списка имеющихся в ИНВ-3 по номеру заказа на закупку.

Для этого на главном экране необходимо нажать на кнопку  в поле «Партия» и перейти в сеанс «Заказы» (Рисунок 11)

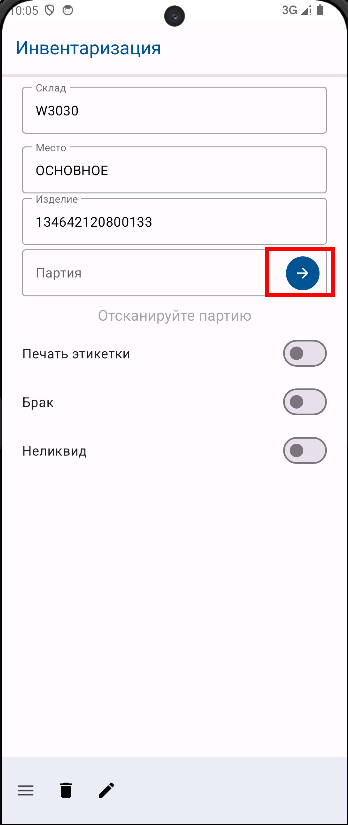


Рисунок11 Переход в сеанс «Заказы»

В сеансе «Заказы» представлены карточки партий с информацией:

* Заказ (по какому заказу на закупку поступила данная партия ТМЦ на склад )
* Партия (номер партии, который имеется в ИНВ-3)
* Кол-во по ИНВ-3 (кол-во по данной партии, которое числится в ИНВ-3 в единицах измерения системы Infor LN)
* Учтенное кол-во (кол-во, которое уже посчитали и ввели в мобильное приложение)
* Место (номер места по ИНВ-3 если партия находится на одном месте или же кнопка «МЕСТА» для просмотра всех мест, на которых находится данная партия) (Рисунок 12)

Выбрать партию можно нажав на карточку выделив ее и нажать на кнопку . (Рисунок 12)

После чего данная партия будет выбрана и откроется экран ввода кол-ва.

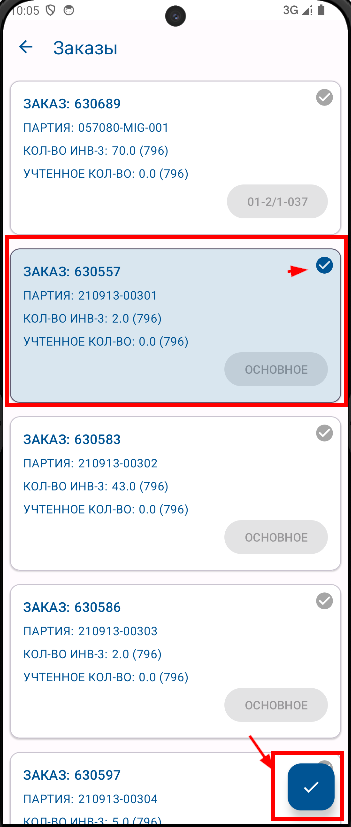


Рисунок12 Выбор партии в сеансе «Заказы»

В случае, когда партия хранится на разных местах хранения в системе Infor LN, можно выбрать партию с конкретного места хранения.

Для этого в сеансе «Заказы» нажать на кнопку «МЕСТА» (Рисунок 13) и перейти в сеанс «Места».

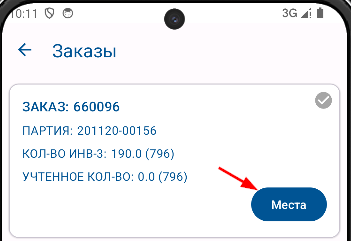


Рисунок 13 Переход в сеанс с местами хранения

В сеансе «Места» представлены карточки конкретной партии в разрезе складских мест в системе Infor LN

* Место (номер места по ИНВ-3 на котором лежит запас в системе Infor LN)
* Кол-во по ИНВ-3 (в единицах измерения системы Infor LN)
* Учтенное кол-во (кол-во, которое уже посчитали и ввели в мобильное приложение)

Можно выбрать конкретное место, которое подлежит инвентаризации. (Рисунок 14)

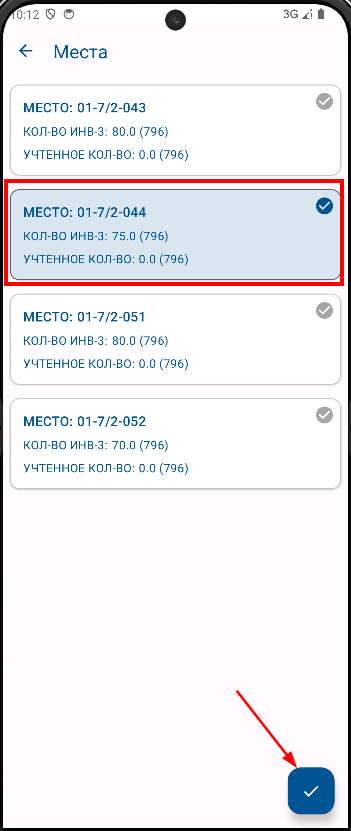


Рисунок14 Выбор партии с конкретного места хранения

ВАЖНО! Если выбрать карточку с конкретным местом, то данное место и запишется в данные инвентаризации (т.е. подставится на главный экран приложения)

1. Ввод посчитанного кол-ва ТМЦ к учету осуществляется в окне «Кол-во» (Рисунок 15)

После выбора конкретной партии или конкретной партии и места, автоматически откроется окно ввода кол-ва.

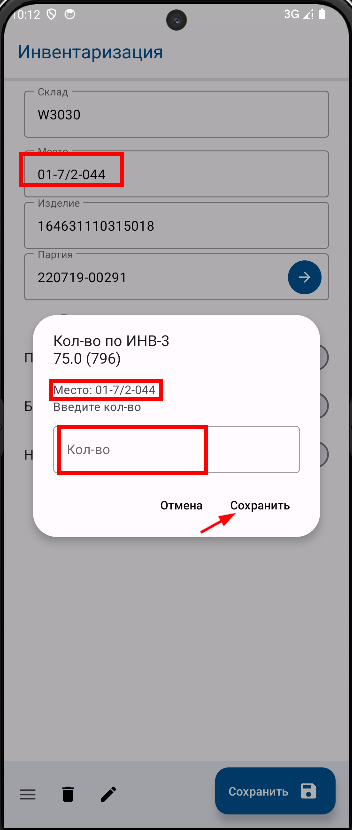


Рисунок 15 Ввод кол-ва к учету

После ввода кол-ва необходимо нажать на кнопку «Сохранить».

И дождаться сообщения «Запись успешно сохранена».

### Ввод партии вручную в ТСД

#### Спецификация

Таблица 4

Спецификация

|  |  |
| --- | --- |
| Условия выполнения | Данный шаг выполняется при отсутствии партии в ИНВ-3 |
| Исполнитель | Оператор склада |
| Вход | Рукописная этикетка, партия не привязана к заказу на закупку |
| Выход | Введены данные инвентаризации в ТСД |

#### Назначение

Данный шаг предназначен для ввода данных с ручной этикетки, когда отсутствует заказ на закупку и нет возможности идентифицировать данный запас, в мобильное приложение ТСД для проведения инвентаризации.

#### Последовательность действий

1. Запустить мобильное приложение на ТСД.
2. На главном экране нажать на команду  «Карандаш» (ручной ввод данных)
3. В открывшемся окне необходимо заполнить поля вручную через клавиатуру на ТСД или через виртуальную клавиатуру на экране приложения и сохранить данные
4. В случае, когда партия отсутствует в заказе на инвентаризацию выйдет предупреждающее сообщение «Нет позиций по ИНВ-3. Выберите партию по номеру закупок» (Рисунок 16)

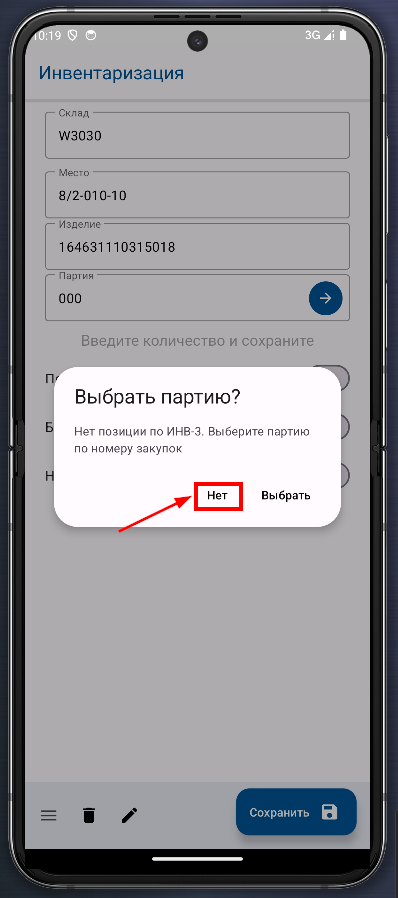


Рисунок 16 Ввод отсутствующей партии в данные инвентаризации

Для ввода отсутствующей партии в систему необходимо нажать на кнопку «Нет» и

1. Ввод посчитанного кол-ва ТМЦ к учету осуществляется в окне «Кол-во» (Рисунок 17)

После записи новой партии автоматически откроется окно для ввода кол-ва.

В данном окне будет доступна информация:

* Единицы измерения (единицы, в которых хранится запас по данному изделию в системе Infor LN)
* Место (место по которому ведется инвентаризация и где обнаружен данный запс ТМЦ)

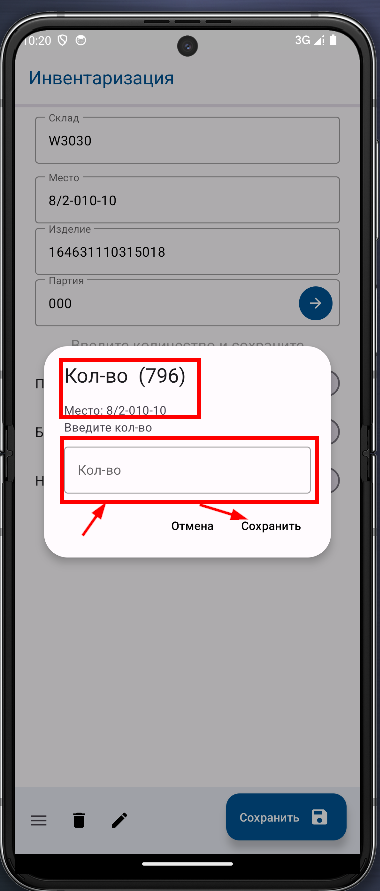


Рисунок 17 Ввод кол-ва к учету

После ввода кол-ва необходимо нажать на кнопку «Сохранить». И дождаться сообщения «Запись успешно сохранена».

### Сканирование места отличного от системного

#### Спецификация

Таблица 5

Спецификация

|  |  |
| --- | --- |
| Условия выполнения | Данный шаг выполняется при сканировании запаса, который физически лежит на месте отличном от места в системе Infor LN |
| Исполнитель | Оператор склада |
| Вход | Физически запас ТМЦ лежит на месте не как в системе Infor LN |
| Выход | Введены фактические данные в ТСД |

#### Назначение

Данный шаг предназначен для ввода физического места, на котором хранится запас ТМЦ, в мобильное приложение ТСД для проведения инвентаризации.

#### Последовательность действий

1. Запустить мобильное приложение на ТСД.
2. В открывшемся окне необходимо заполнить поля данными с этикетки:

* Склад
* Место
* Изделие
* Партия

Вносить данные можно путем сканирования или же использовать ручной ввод через команду  «Карандаш».

1. В случае, когда физическое отсканированное место отличается от места по ИНВ-3 из системы Infor LN, в мобильном приложении выйдет предупреждающее сообщение с вопросом о дальнейшем действии.

Пример №1: Когда по системе Infor LN партия хранится на одном конкретном месте (02-020-011) , а сканирование происходит с другого места(01-000-02).

В данном случае приложение предупредить о том, что партия хранится на другом месте и задаст уточняющий вопрос «Изменить системное место?» (Рисунок 18)

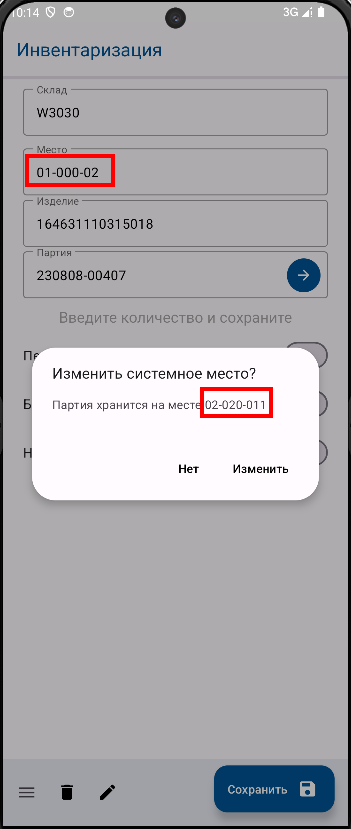


Рисунок 18 «Изменить системное место? Партия хранится на другом месте»

Если партия физически лежит именно на месте, с которого идет сканирование, а в системе некорректные/устаревшие данные, то необходимо нажать на «Изменить».

В таком случае запас будет перемещен на физическое место и в дальнейшем будут изменения в системе Infor LN. (перемещение по местам)

Если данные в системе Infor LN корректны, а отсканированное физическое место ошибочное, то необходимо выбрать команду «Нет». Вернуться на главный экран ввода данных и изменить отсканированное место.

ВАЖНО! Необходимо правильно заполнять поле место на главном экране, так как именно это место будет записываться в результат Инвентаризации.

Пример № 2: Когда по системе Infor LN партия храниться на одном конкретном месте, но физически лежит на нескольких местах. И есть необходимость добавить новое место хранения.

Приложение проверит, лежит ли запас на данном месте и предупредит о том, что партия хранится на другом месте и задаст уточняющий вопрос «Изменить системное место?»

В случае добавления нового места хранения необходимо нажать команду «Нет».

После чего автоматически откроется окно с вводом «Кол-ва» на новое место хранения. (Рисунок 19)

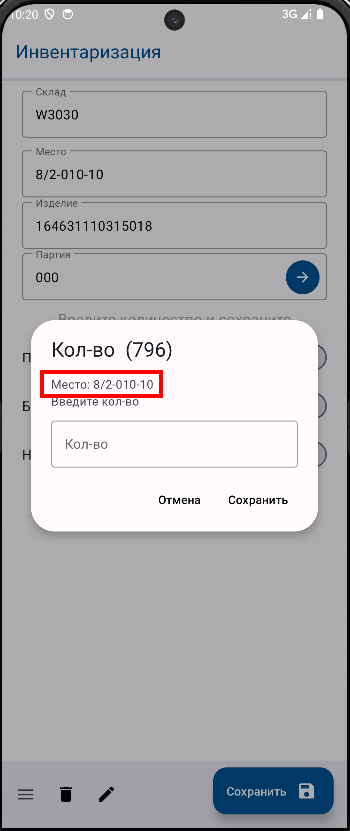


Рисунок 19 Ввод кол-ва на новое место хранения.

ВАЖНО! Так как в системе отсутствует данное место хранения, подсказки в виде кол-ва по ИНВ-3 не будет.

Пример №3: Когда по системе Infor LN партия храниться на многих местах, но отсканированное место не совпадает ни с одним из системы.

В данном случае приложение предупредит о том, что партия хранится на других местах и задаст уточняющий вопрос «Выбрать системное место?» (Рисунок 20)

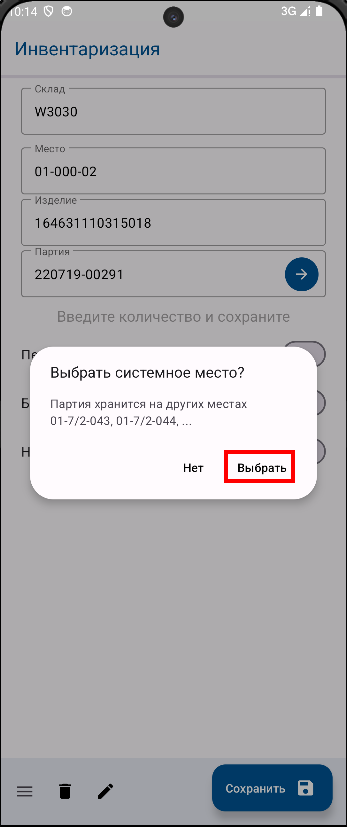


Рисунок 20 «Выбрать системное место? Партия хранится на других местах»

При нажатии на кнопку «Выбрать» открывается сеанс «Места», где отображаются все места, на которых хранится партия в системе Infor LN.

Необходимо выбрать место из имеющихся мест.( нажать на карточку с местом)

При нажатии на кнопку «Нет» система оставляет введенное место и переходит в окно ввода кол-ва

1. После изменения системного места необходимо продолжить работу обычном режиме, ввести кол-во к учету и сохранить данные инвентаризации.

### Изменение/просмотр учтенного запаса.

#### Спецификация

Таблица 5

Спецификация

|  |  |
| --- | --- |
| Условия выполнения | Данный шаг выполняется при наличии учтенного запаса по изделию |
| Исполнитель | Оператор склада |
| Вход | В приложение внесены данные по кол-ву ТМЦ |
| Выход | Изменение кол-ва учтенного ТМЦ |

#### Назначение

Данный шаг предназначен для изменения ранее учтенного запаса по ТМЦ и просмотра кол-ва, принятого к учету в разрезе заказа или места.

#### Последовательность действий

1. Запустить мобильное приложение на ТСД.
2. Открывается главный экран приложения.

В открывшемся окне необходимо заполнить поля данными с этикетки:

* Склад
* Место
* Изделие
* Партия

1. При повторном вводе данных в мобильное приложение открывается окно «Кол-во» с введенным ранее кол-вом к учету. (Рисунок 21)

Для изменения ранее введенного кол-ва, необходимо очистить поле и ввести новое кол-во.

Нажать на кнопку «Сохранить» и данные актуализируются.

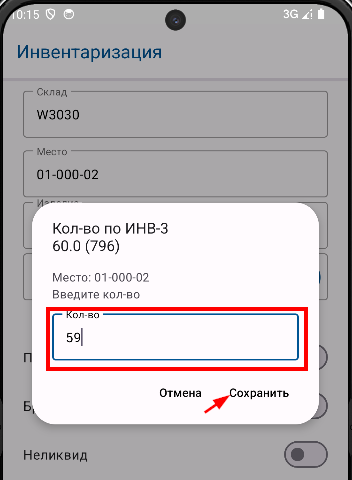


Рисунок 21 Изменение введенного кол-ва

1. Для просмотра введенных ранее данных в систему можно ввести данные с этикетки и перейти в сеанс «Заказы» через поле «Партия» нажав на кнопку  (Рисунок 22)

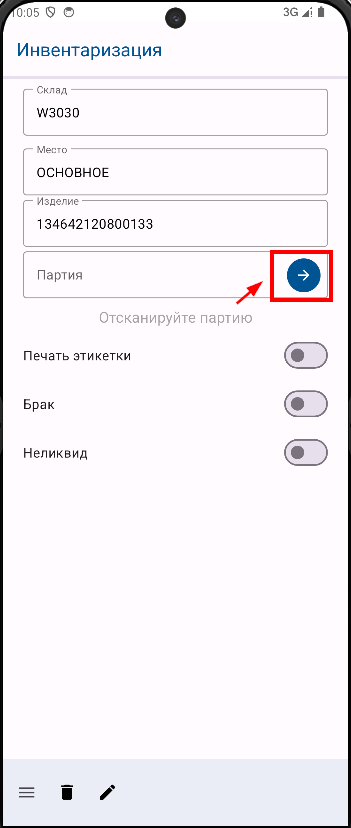


Рисунок 22 Переход в сеанс «Заказы»

1. В сеансе «Заказы» партии, по которым был учет подсвечиваются зеленым цветом.

Также в карточке отображается учтенное кол-во и наглядно представлено, сколько числится по ведомости инвентаризации и сколько запаса принято к учету. (Рисунок 23)

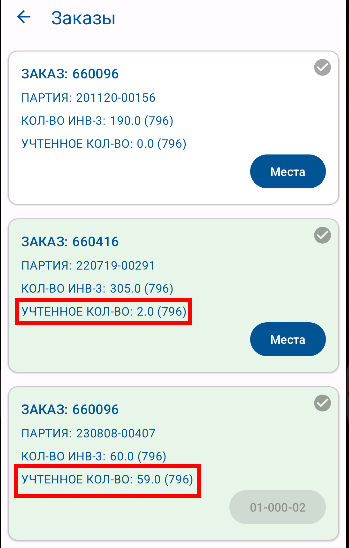


Рисунок 23 Учтенное кол-во по партии.

1. В сеансе «Места» подсвечиваются зеленым цветом места, по которым был учтен запас.

Также отображается учтенное кол-во и кол-во по ведомости инвентаризации из системы Infor LN. (Рисунок 24)

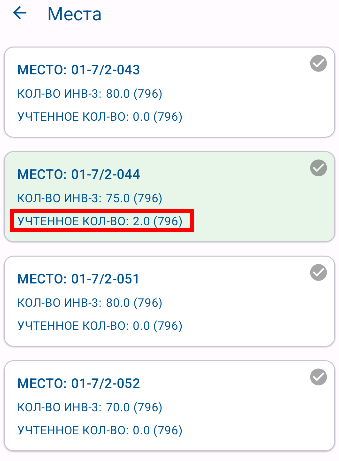


Рисунок 24 Учтенное кол-во на определённом месте