**Ургенчский филиал Ташкентского университета информационных технологий**

**имени Мухаммада Аль-Хоразмий**



Курсовая работа

Название: Разработка программы инвентаризации существующего оборудования организация

Спецификации требований к программному обеспечению.

Кафедра: Програмный инжинеринг

Сделал студент группы 943-20

Зарипов Шухрат Гуматдинович

Ургенч – 2022

СОДЕРЖАНИЕ

1. Введение \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_3

1.1 Цель\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_3

1.2 Определения и сокращения\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_4

2. Общее описание \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_4

2.1 Функциональность\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_4

2.2 Целевая аудитория\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_4

2.3 Масштаб проекта\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_4

3. Use case\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_5

4. Концептуальное представление структуры хранения данных\_\_\_\_6

5. UX/UI дизайн мобильного приложения\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_9

4. Приложение \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_11

7. Заключение\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_18

1. Введение

Инвентаризацией называется проведение проверки имущества предприятий и организаций в количественном и качественном эквиваленте за определенный период с помощью сверки данных бухгалтерского учета и имеющихся данных [1].

При проведении инвентаризации необходимо руководствоваться т принципами: количественная оценка фактического присутствия имущества, сличение данных бухгалтерского учета с данными фактического присутствия в наличии, проведение процесса полной отчетности и прозрачности в отображении данных по процессу проведения инвентаризационного учета

Целью проекта является повышение работоспособности предприятия за счет :

* Сокращения времени поиска информации по объектам инвентаризации
* Сокращение времени на предоставление отчетности по объектам инвентаризации
* Автоматического введения журналов списания и перемещения оборудования.

Для достижения поставленных целей, должны быть решены следующиезадачи:

1. Изучение процесса инвентаризации;
2. Нахождение этапов занимающих наибольшее время;

1.1 Цель

*Целью работы* является проектирование и разработка программного комплекса для проведения инвентаризационного учета оборудования на предприятиях с последующим составлением отчетности.

*Оригинальность и новизна результатов научного исследования* заключается в разработке собственных алгоритмов редактирования и форматирования данных, объединенных в адаптеры преобразования необходимого формата данных, внедренные в ПК.

*Практическая значимость работы* состоит в упрощении проведения инвентаризационного учета оборудования с автоматизированным проведением анализа и отчетности процессов для пользователей, относящихся к предметной области «инвентаризация».

*Область применения* программного обеспечения (ПО) на предприятиях любой категории и типа, занимающихся любым видом деятельности, начиная от малого частного бизнеса и заканчивая большими корпорациями, которым требуется составлять отчетность по проведению инвентаризационного учета.

# 1.2 УСЛОВНЫЕ ОБОЗНАЧЕНИЯ

|  |  |
| --- | --- |
| ПО | Программное обеспечение |
| ПП | Программный продукт |
| Штрихкод | Штриховой код |
| QR-код (англ.Quick Response) | Код быстрого отклика |
| ОЛ | Ответственное лицо |
| СМП | Сервер мобильного приложения |
| API (англ. Application programming  interface) | Программный интерфейс приложения |
| SQL (англ. Structured Query  Language) | Язык программирования  структурированных запросов |
| ЯП | Язык программирования |
| ГОСТ | Государственный стандарт |
| ТМЦ | Товарно-материальные ценности |

2. **Общее описание продукта**

Наименование продукта: **Essential inventory**

Инвентаризация оборудовании — обязательная процедура, которую все без исключения организации должны проводить не реже одного раза в три года. При этом для каждой учтенной позиции указывают не только наименование, но и материально ответственных лиц, а также место хранения или размещения оборудовании.

Данный продукт будет помогать организацию проводит ревизию оборудования с той периодичностью, которую предусматривают внутренние правила предприятия. Это делают для того, чтобы узнать точное количество остатков ТМЦ (товарно-материальных ценностей) на складе или в торговом зале, выявить пересортицу продукции, подтвердить факт кражи или не работоспособность оборудования.

**2.1 Функциональность продукта**

Функционал мобильного приложения позволяет непрерывно контролировать процесс проведения инвентаризации ответственным лицом и обеспечивать оперативную обратную связь с исполнителями.

**2.2 Целевая аудитория**

Целевым аудиторием могут оказаться предприятия любой категории и типа, занимающихся любым видом деятельности, начиная от частного бизнеса и заканчивая большими корпорациями, которым необходимо вести подотчёт учётных единиц и основных средств на предприятии.

По форме собственности предприятия делятся на основные три вида:

− Государственные – предприятия, которыми полностью и в значительной мере владеет государство;

− Частные – предприятия, которые принадлежат собственникам (физическим или юридическим лицам) с полными правами на собственность;

− Кооперативные – предприятия, владельцами которого могут стать собственники, имеющие право на собственность как частично (долевое), так и полностью.

**2.3 Масштаб проекта**

* + - Проект подразумевается в первую очередь региональным (г. Ургенч).

**3. Use case**

* Пользователи имеют 3 роли:
* **Директор отдела ––** ответственный за проведение процесса инвентаризации.

Имеет все привилегии, также:

- Отправка данных к бухгалтеру

* **Складчик**– сотрудник, осуществляющий пересчет общего количества товаров в складской зоне. (Рис 1,1)

Имеет привилегии как:

- Добавить тип оборудования

- Добавить дата выхода оборудования из склада

* **Сотрудник–** сотрудник, осуществляющий пересчет количества товаров в складской зоне в разрезе номенклатурных единиц.(Рис 1,2)

Имеет привилегии как:

- Местонахождения оборудования

- Писать тип оборудования

- Дата инвентаризации

- Сканировать штрих-коды

- Состояние оборудования

- Писать ответственное лицо

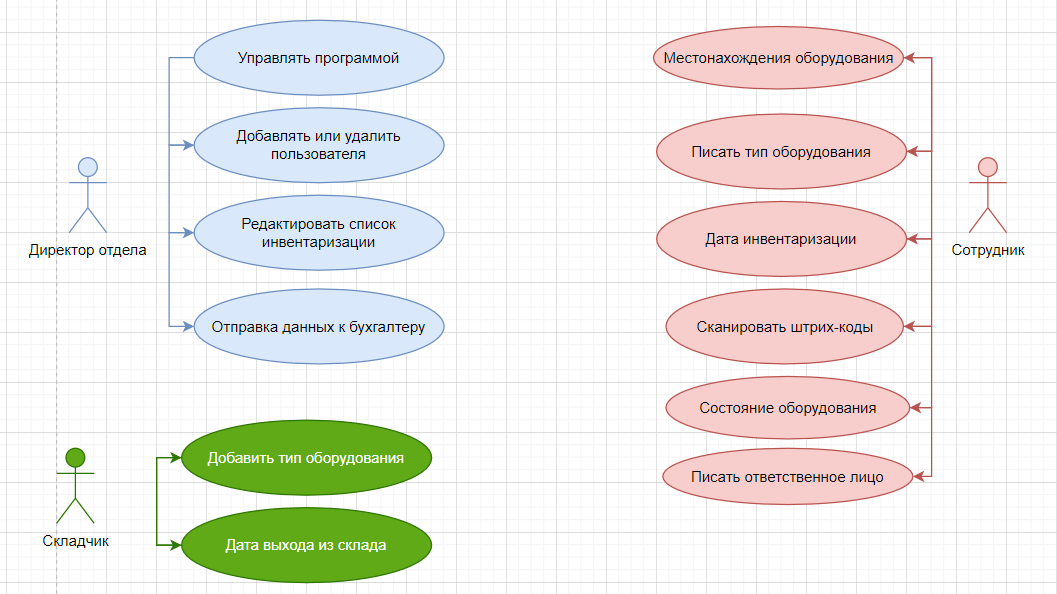


Рис 1: Use case диаграмма

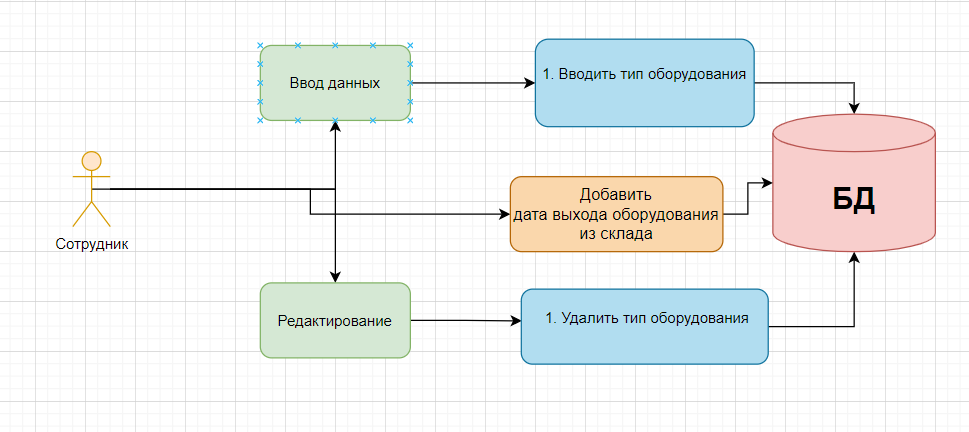


Рис 1,1 : **Складчик**

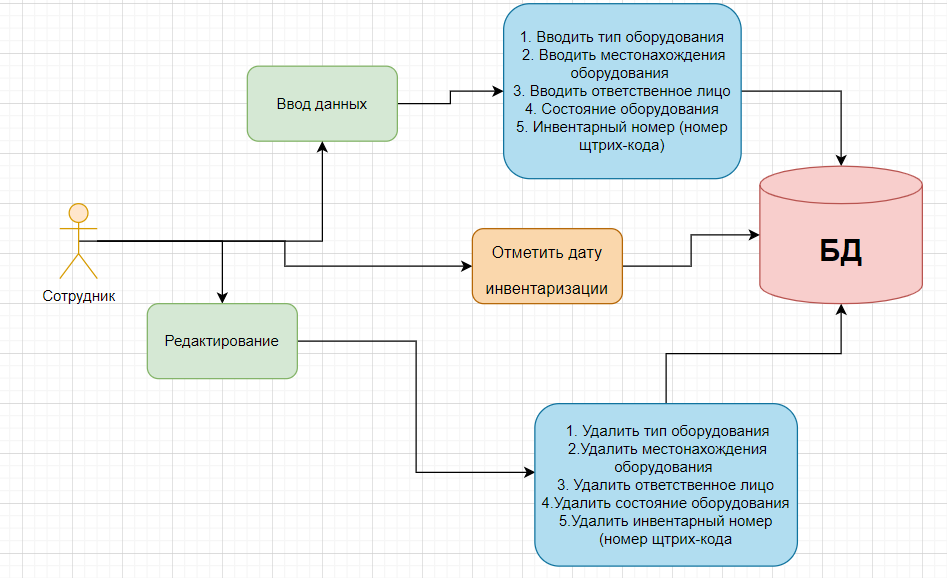


Рис 1,2: **Сотрудник** который проводит инвентаризацию

**4.** **Концептуальное представление структуры хранения данных**

Для абстрагирования от уже конкретных существующих систем управления базами данных и моделей разрабатываемого программного комплекса создаётся процесс концептуального проектирования, в ходе которого будет создана ER-модель работы с данными системы. Данная модель описывает предметную область «инвентаризация оборудования» с помощью заложенной модели «сущность-связь» [8]. Созданная модель характеризует и показывает структуру базы данных показана в виде логической модели базы данных на рисунке 3. Модель на рисунке 2. была спроектирована с помощью программной среды MySQL Workbench.

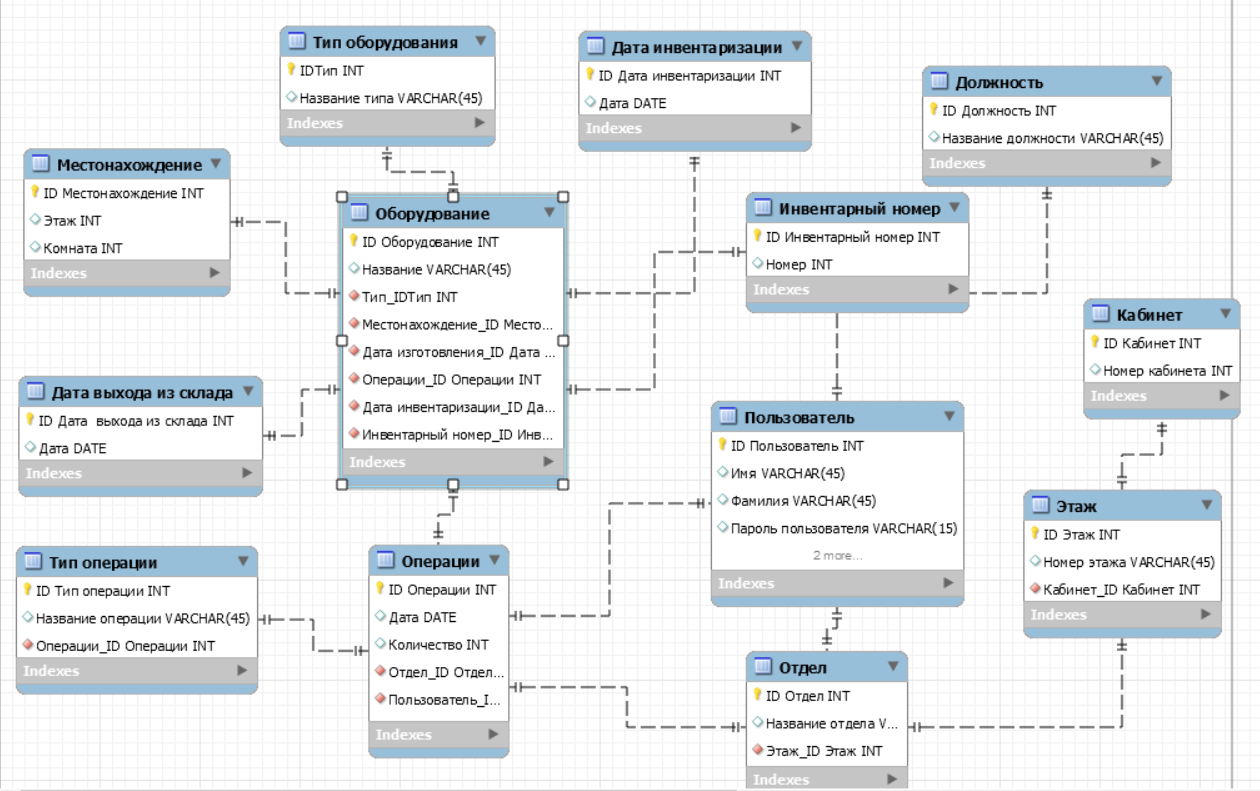
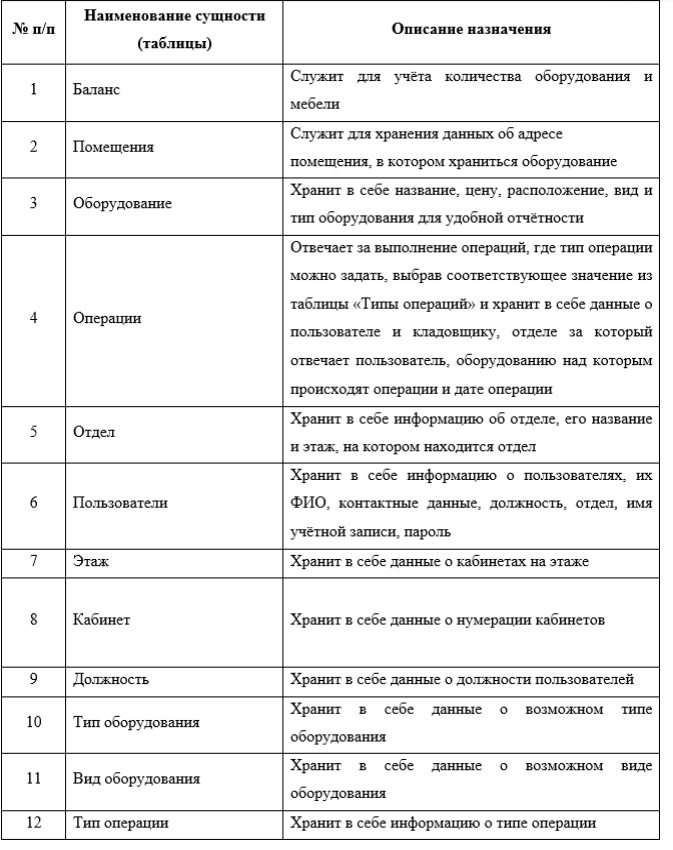


Рисунок 2. – Логическая модель структуры базы данных программного комплекса для инвентаризации оборудования

В спроектированной логической модели базы данных на рисунке 3.1 показаны сущности базы данных, взаимосвязанные согласно логике приложения для инвентаризации оборудования. Описание сущностей приведено в таблице:

**Разработка алгоритмов проведения инвентаризации**.

Документ «Инвентаризация товаров» считается полностью сформированным, если сумма количества товаров совпадает с количеством товаров контролера. В этом случае документу присваивается статус «К выгрузке». В случае разницы в указанных показателях документу присваивается статус «Редактируется». При повторном прохождении по складской зоне счетчика предыдущим документам в программе по данной зоне присваивается статус «Не выгружать».

**5. UX/UI дизайн мобильного приложения**

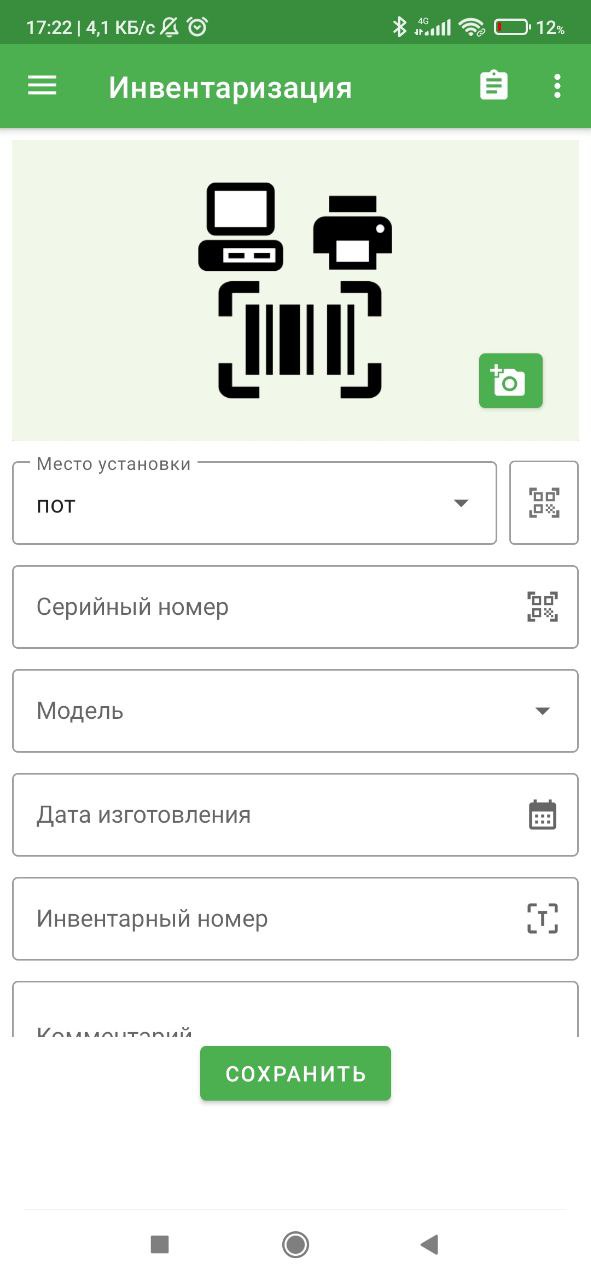


Рис 3: Главный экран

Главная страница продукта выглядит очень простым для удобства пользования.

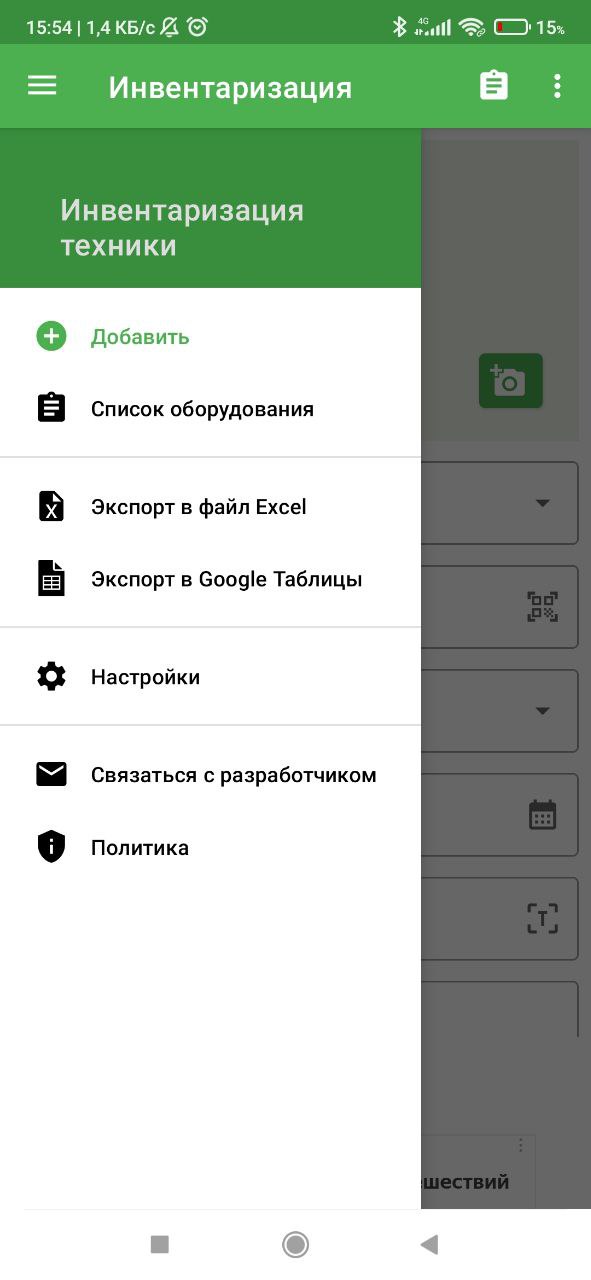
****

Рис. 4: Меню программы

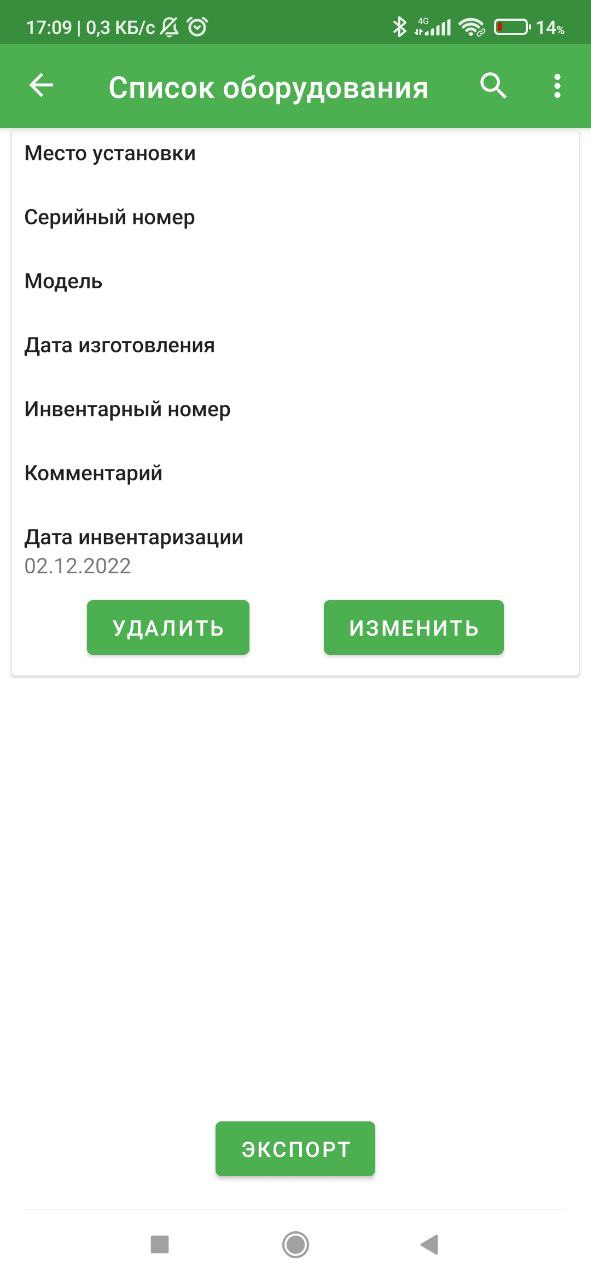
****

Рис. 5: Список оборудования

В данном окне мы можем:

* Удалить добавленное обородувание
* Изменить добавленное обородувание

**Приложение:**

class App(*QMainWindow*):

    def \_\_init\_\_(*self*):

*super*().\_\_init\_\_()

*self*.db = Database()

*self*.initUI()

    def initUI(*self*):

        # Настройка шрифта

*self*.font = QFont()

*self*.font.setFamily("Arial")

*self*.font.setPointSize(12)

*self*.setFont(*self*.font)

        # Создание меню

*self*.create\_menu()

        # Создание слоёв

*self*.grid = QGridLayout()

*self*.grid.setSpacing(5)

*self*.layoutInput = QGridLayout()

*self*.groupBox = QGroupBox("Добавление данных")

*self*.gridBox = QGridLayout()

        # Создание элементов

*self*.inputsCharacteristic = []

*self*.labelCharacteristic = []

*self*.comboBoxLocTabel = []

*self*.comboBoxResPerTabel = []

*self*.inpSearch = QLineEdit()

*self*.labelSearch = QLabel()

*self*.labelSearch.setText("Поиск")

*self*.labelSerType = QLabel("По типу")

*self*.comboBoxTypes = QComboBox()

*self*.cbTypesFilter = QComboBox()

        types = *self*.get\_types()

        for i, item in enumerate(types):

*self*.comboBoxTypes.insertItem(i, item)

*self*.cbTypesFilter.insertItem(i, item)

*self*.cbTypesFilter.insertItem(0, "Все")

*self*.cbTypesFilter.setCurrentIndex(0)

*self*.btnAdd = QPushButton()

*self*.btnAdd.setText("Добавить")

*self*.btnAdd.setMinimumSize(90, 40)

*self*.labelType = QLabel()

*self*.labelType.setText("Тип оборудования")

*self*.inpSerialNumber = QLineEdit()

*self*.labelSerialNumber = QLabel()

*self*.labelSerialNumber.setText("Серийный номер")

*self*.inputBoxLocation = QLineEdit()

*self*.labelLocation = QLabel()

*self*.labelLocation.setText("Месторасположение")

*self*.labelSerLoc = QLabel()

*self*.labelSerLoc.setText("Месторасположение")

*self*.cbLocFilter = QComboBox()

        locations = *self*.db.get\_location()

*self*.loc\_model = QStandardItemModel()

        for i, loc in enumerate(locations):

*self*.loc\_model.appendRow(QStandardItem(loc[1]))

*self*.cbLocFilter.insertItem(i, loc[1])

        completer\_loc = QCompleter(*self*.loc\_model, *self*)

        completer\_loc.setCaseSensitivity(Qt.CaseInsensitive)

        completer\_loc.setCompletionRole(Qt.DisplayRole)

        completer\_loc.setCompletionColumn(0)

*self*.inputBoxLocation.setCompleter(completer\_loc)

*self*.cbLocFilter.insertItem(0, "Все")

*self*.cbLocFilter.setCurrentIndex(0)

*self*.inputBoxResponsiblePerson = QLineEdit()

*self*.cbResPersFilter = QComboBox()

*self*.labelResponsiblePerson = QLabel()

*self*.labelResponsiblePerson.setText("Ответственное лицо")

*self*.labelSerResPers = QLabel()

*self*.labelSerResPers.setText("Ответственное лицо")

        responsible\_persons = *self*.db.get\_responsible\_person()

*self*.res\_pers\_model = QStandardItemModel()

        for i, res\_pers in enumerate(responsible\_persons):

*self*.res\_pers\_model.appendRow(QStandardItem(res\_pers[1]))

*self*.cbResPersFilter.insertItem(i, res\_pers[1])

        completer\_res\_pers = QCompleter(*self*.res\_pers\_model, *self*)

        completer\_res\_pers.setCaseSensitivity(Qt.CaseInsensitive)

        completer\_res\_pers.setCompletionRole(Qt.DisplayRole)

        completer\_res\_pers.setCompletionColumn(0)

*self*.inputBoxResponsiblePerson.setCompleter(completer\_res\_pers)

*self*.cbResPersFilter.insertItem(0, "Все")

*self*.cbResPersFilter.setCurrentIndex(0)

*self*.btnSearch = QPushButton()

*self*.btnSearch.setText("Найти")

*self*.table = *self*.\_\_create\_table()

*self*.fill\_table(*self*.db.get\_items())

        # Размещение слоёв

*self*.layoutInput.addWidget(*self*.labelSearch, 0, 0)

*self*.layoutInput.addWidget(*self*.inpSearch, 0, 1)

*self*.layoutInput.addWidget(*self*.labelSerType, 0, 2)

*self*.layoutInput.addWidget(*self*.cbTypesFilter, 0, 3)

*self*.layoutInput.addWidget(*self*.labelSerLoc, 0, 4)

*self*.layoutInput.addWidget(*self*.cbLocFilter, 0, 5)

*self*.layoutInput.addWidget(*self*.labelSerResPers, 0, 6)

*self*.layoutInput.addWidget(*self*.cbResPersFilter, 0, 7)

*self*.layoutInput.addWidget(*self*.btnSearch, 0, 8)

*self*.layoutInput.addWidget(*self*.groupBox, 1, 0, 3, 9)

        # Размещение объектов в GroupBox

*self*.groupBox.setLayout(*self*.gridBox)

*self*.font.setPointSize(10)

*self*.groupBox.setFont(*self*.font)

*self*.gridBox.addWidget(*self*.labelType, 0, 0)

*self*.gridBox.addWidget(*self*.comboBoxTypes, 0, 1)

*self*.gridBox.addWidget(*self*.labelLocation, 0, 2)

*self*.gridBox.addWidget(*self*.inputBoxLocation, 0, 3)

*self*.gridBox.addWidget(*self*.labelSerialNumber, 1, 0)

*self*.gridBox.addWidget(*self*.inpSerialNumber, 1, 1)

*self*.gridBox.addWidget(*self*.labelResponsiblePerson, 1, 2)

*self*.gridBox.addWidget(*self*.inputBoxResponsiblePerson, 1, 3)

*self*.gridBox.addWidget(*self*.btnAdd, 1, 10)

        # self.btnAdd.setIcon(QApplication.style().standardIcon(QStyle.SP\_FileDialogNewFolder))

*self*.grid.addWidget(*self*.table, 0, 0, 1, 1)

*self*.grid.addLayout(*self*.layoutInput, 1, 0, 1, 1)

        # Подключение событий

*self*.inpSearch.textChanged.connect(*self*.search\_input)

*self*.comboBoxTypes.currentIndexChanged.connect(*self*.change\_input)

*self*.btnAdd.clicked.connect(*self*.add\_item)

*self*.btnSearch.clicked.connect(*self*.search)

*self*.mainLayout = QWidget()

*self*.mainLayout.setLayout(*self*.grid)

*self*.setCentralWidget(*self*.mainLayout)

*self*.change\_input()

*self*.setWindowTitle('Инвентаризация')

*self*.setMinimumSize(600, 600)

*self*.setGeometry(300, 25, 700, 700)

*self*.showMaximized()

*self*.show()

*self*.statusBar().showMessage("Database loaded", 3000)

    def update\_mode(*self*):

        if *self*.editMode:

            for elem in *self*.comboBoxLocTabel:

                elem.setDisabled(*self*.editMode)

            for elem in *self*.comboBoxResPerTabel:

                elem.setDisabled(*self*.editMode)

*self*.editMode = False

        else:

            for elem in *self*.comboBoxLocTabel:

                elem.setDisabled(*self*.editMode)

            for elem in *self*.comboBoxResPerTabel:

                elem.setDisabled(*self*.editMode)

*self*.editMode = True

    def inventory\_mode(*self*):

        if not *self*.inventoryMode:

*self*.inventoryModeItemsID = []

*self*.comboBoxLocTabel = []

*self*.comboBoxResPerTabel = []

*self*.table.clear()

*self*.table\_header()

*self*.table.setRowCount(0)

*self*.inventoryMode = not *self*.inventoryMode

        else:

*self*.search()

*self*.inventoryMode = not *self*.inventoryMode

    def create\_menu(*self*):

        # Действия

        addResPersAction = QAction("Добавить ответственное лицо", *self*)

        addResPersAction.triggered.connect(*self*.add\_res\_pers)

        deleteResPersAction = QAction("Удалить ответственное лицо", *self*)

        deleteResPersAction.triggered.connect(*self*.delete\_res\_pers)

        addLocationAction = QAction("Добавить месторасположение", *self*)

        addLocationAction.triggered.connect(*self*.add\_location)

        deleteLocationAction = QAction("Удалить местоположение", *self*)

        deleteLocationAction.triggered.connect(*self*.delete\_location)

        addTypeAction = QAction("Добавить тип оборудования", *self*)

        addTypeAction.triggered.connect(*self*.add\_type)

        deleteTypeAction = QAction("Удалить тип оборудования", *self*)

        deleteTypeAction.triggered.connect(*self*.delete\_type)

        editMode = QAction("Режим редактирования", *self*)

        editMode.triggered.connect(*self*.update\_mode)

        inventoryMode = QAction("Режим инвентаризации", *self*)

        inventoryMode.triggered.connect(*self*.inventory\_mode)

        printLabel = QAction("Создать документ штрихкодов", *self*)

        printLabel.triggered.connect(*self*.print\_label)

        # Меню бар

        menubar = *self*.menuBar()

        # Добавление

        addMenu = menubar.addMenu("&Добавить")

        addMenu.addAction(addResPersAction)

        addMenu.addAction(addLocationAction)

        addMenu.addAction(addTypeAction)

        addMenu.addAction(printLabel)

        # Удаление

        deleteMenu = menubar.addMenu("&Удалить")

        deleteMenu.addAction(deleteResPersAction)

        deleteMenu.addAction(deleteLocationAction)

        deleteMenu.addAction(deleteTypeAction)

        # Режимы

        modeMenu = menubar.addMenu("&Режимы")

        modeMenu.addAction(editMode)

        modeMenu.addAction(inventoryMode)

*self*.editMode = False

*self*.inventoryMode = False

*self*.print = False

    def print\_label(*self*):

        dlg = PrintDialog(*self*)

        dlg.exec()

    def add\_res\_pers(*self*):

        text, ok = QInputDialog.getText(*self*, 'Добавить ответственное лицо',

                                        'Введите ФИО:')

        if ok:

            text = text.title()

*self*.db.insert\_res\_pers(text)

*self*.cbResPersFilter.insertItem(1, text)

*self*.res\_pers\_model.appendRow(QStandardItem(text))

            for i in *self*.comboBoxResPerTabel:

                i.insertItem(1, text)

*self*.statusBar().showMessage("Новое ответственное лицо добавлен", 5000)

    def add\_location(*self*):

        text, ok = QInputDialog.getText(*self*, 'Добавить новое местоположение',

                                        'Введите название:')

        if ok:

            text = text[0].upper() + text[1:]

*self*.db.insert\_location(text)

*self*.cbLocFilter.insertItem(1, text)

*self*.loc\_model.appendRow(text)

            for i in *self*.comboBoxLocTabel:

                i.insertItem(1, text)

*self*.statusBar().showMessage("Новое местоположение добавлено", 5000)

**Заключение**

В ходе выполнения данной курсовой работы был создан проект (**Esential inventory**) программного продукта для инвентиризации существующего организации и получен опыт проектирования программных обеспечений. Также был спроектирован UI/UX дизайн на платформе Figma. Были применены знания по языку программирования Python.