

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ**

Инженерно-конструкторское направление. Профиль  
информационные технологии

**Командный кейс №5  
«Электронный склад»**

**Выполнили:**  
Ювенский Лев  
Холебенкова Варвара  
Мишаков Андрей

Москва  
2020/2021 уч. г

## Анализ технических требований

Для запуска сервиса необходимо установить язык программирования Python версии 3.6 или выше. Выбрать дистрибутив для вашей ОС и скачать его можно скачать с официального сайта: <https://www.python.org/downloads/>

Программы, написанные на Python, могут быть выполнены в любой операционной системе.

В проекте использованы четыре модуля из стандартной библиотеки Python. Их дополнительная установка не требуется:

- **tkinter** для работы с графическим интерфейсом (gui).
- **pickle** для хранения объекта модели склада в файле в бинарном режиме.
- **uuid** для генерации случайного индивидуального идентификатора товара.
- **json** для работы с форматом json (<https://www.json.org/json-ru>).

И один нестандартный, который устанавливается при помощи установщика пакетов *pip* (*pip install requests*):

- **requests** для составления HTTP-запросов.

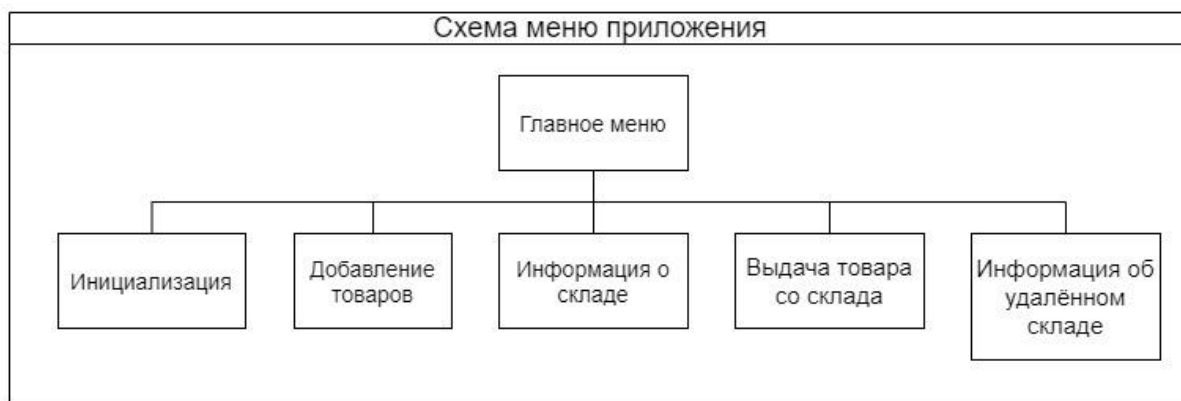
Подробная информация про установку модулей при помощи *pip*:  
<https://pythonru.com/baza-znaniy/ustanovka-pip-dlja-python-i-bazovye-komandy>

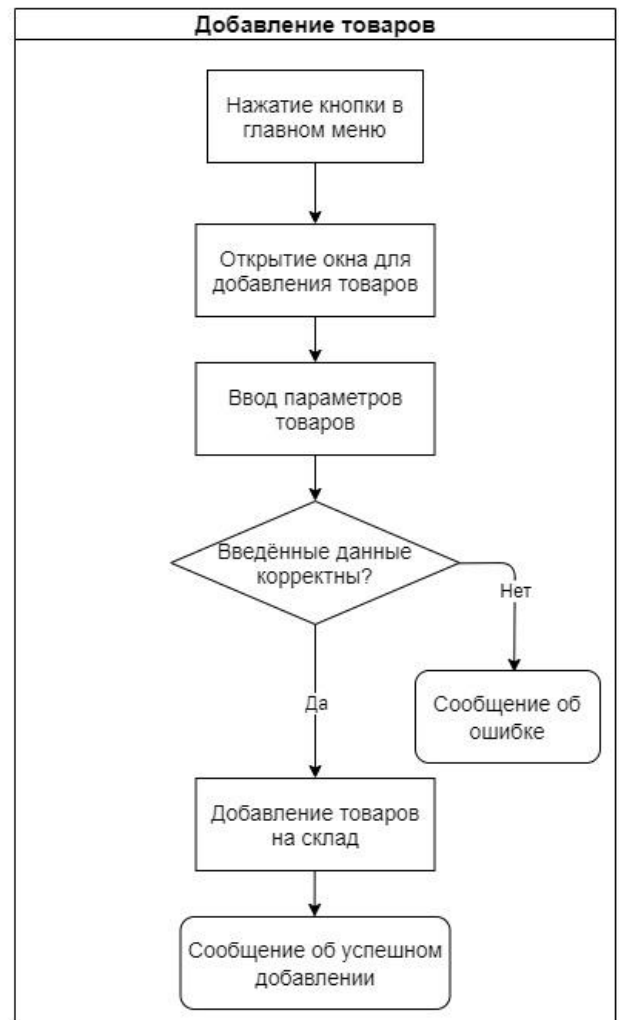
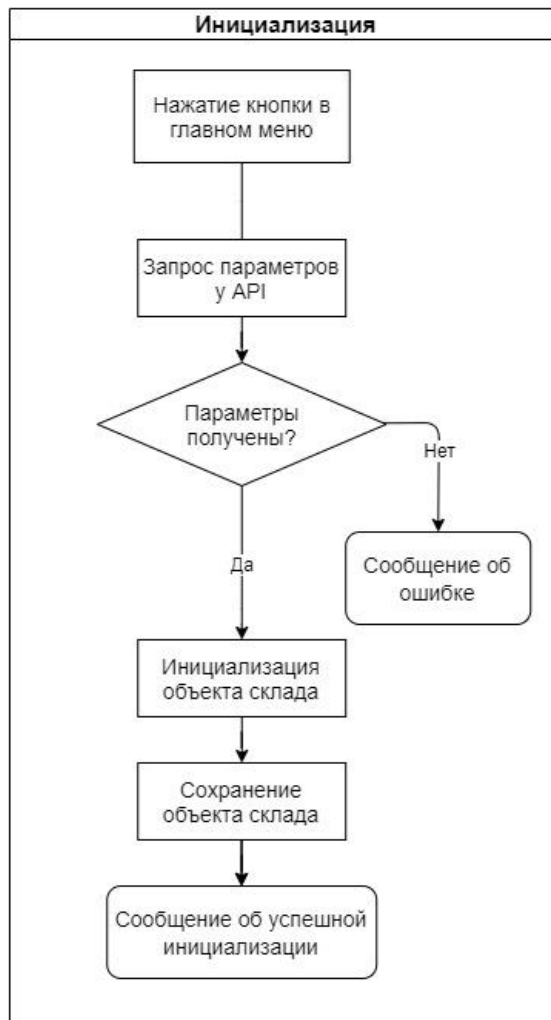
## Обоснование выбора языка программирования и используемых программных средств

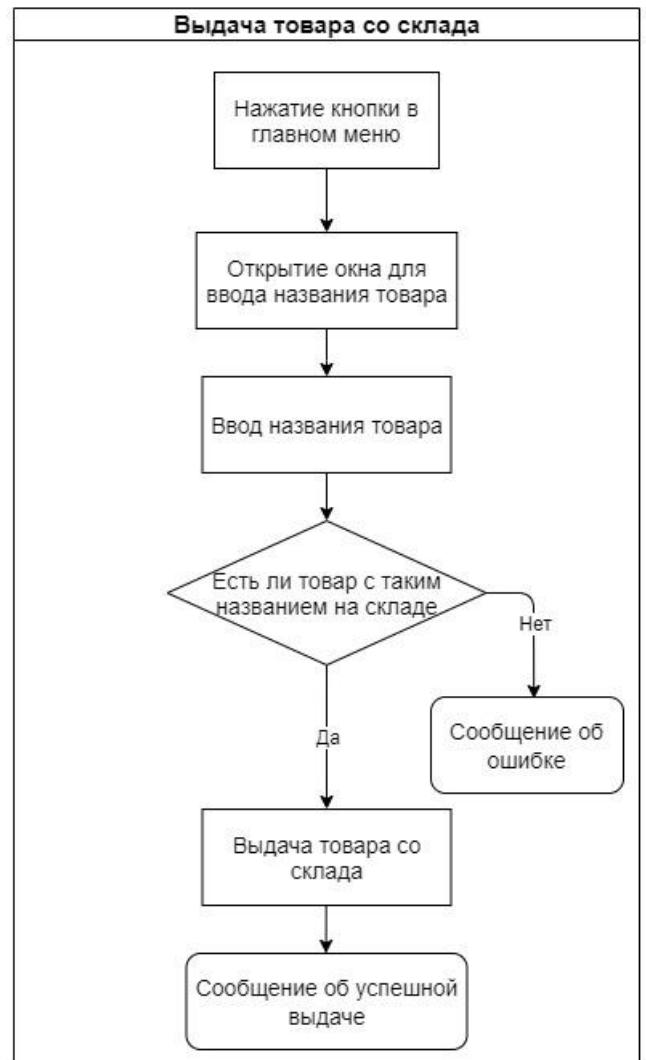
Нами был выбран язык программирования Python, потому что он обладает следующими преимуществами:

1. Простой синтаксис, позволяющий написать приложение быстро.
2. Удобная работа с созданием приложений с графическим интерфейсом.
3. Возможность разделить программный код на файлы (модули), каждый из которых отвечает за свой логический блок, и подключать их друг в друга в рамках одного проекта.
4. Python прост в установке и доступен в большинстве операционных систем.

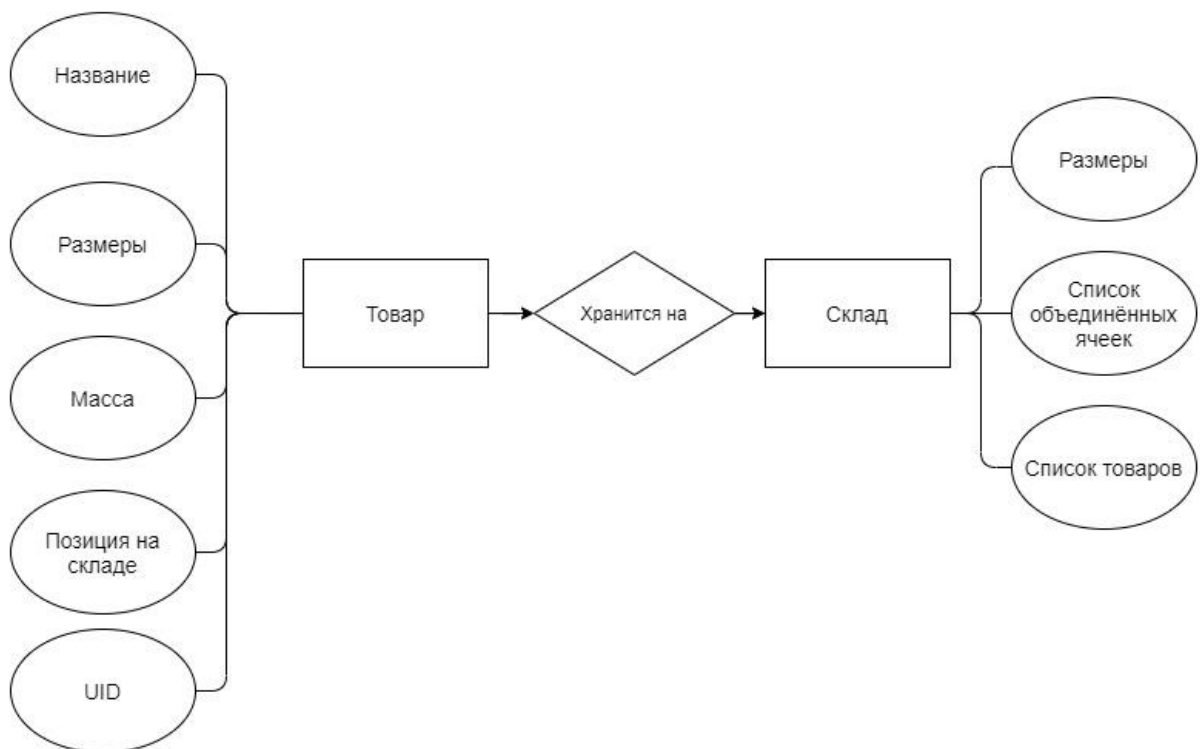
## Структурная и функциональная и схема программного продукта, workflow-диаграмма







## ER-модель



## Программный код

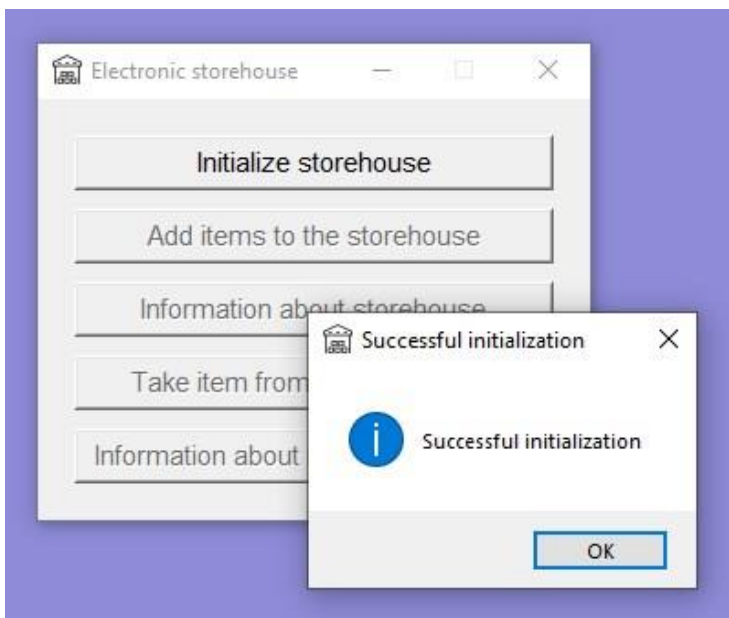
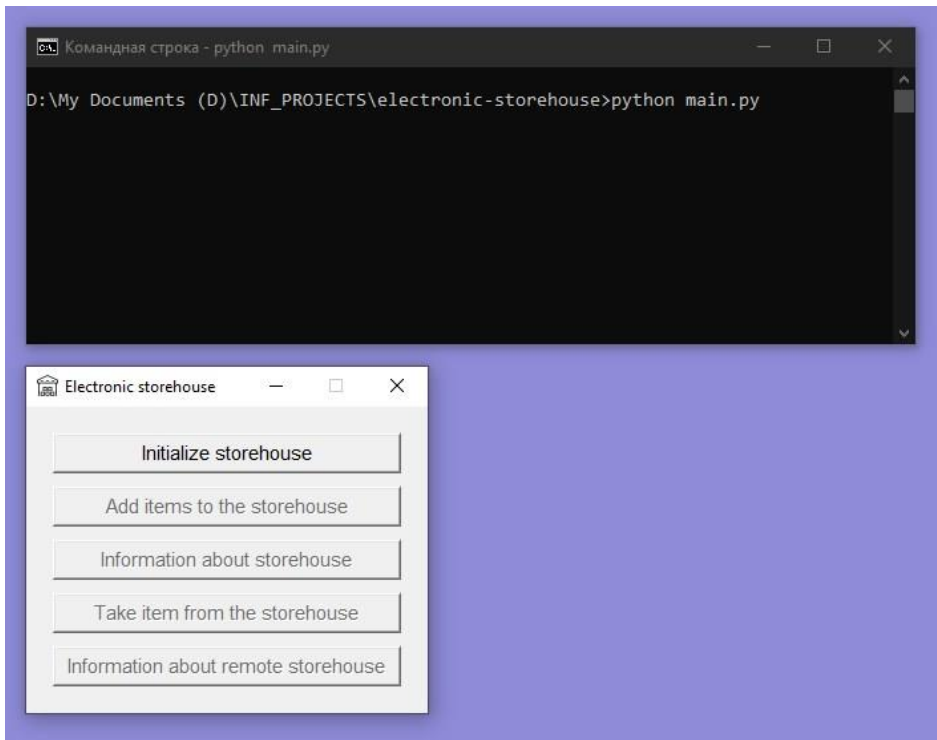
Ссылка на репозиторий: <https://github.com/W-A-L-L-3/electronic-storehouse>

Видео с демонстрацией работы: <https://vimeo.com/514721328>

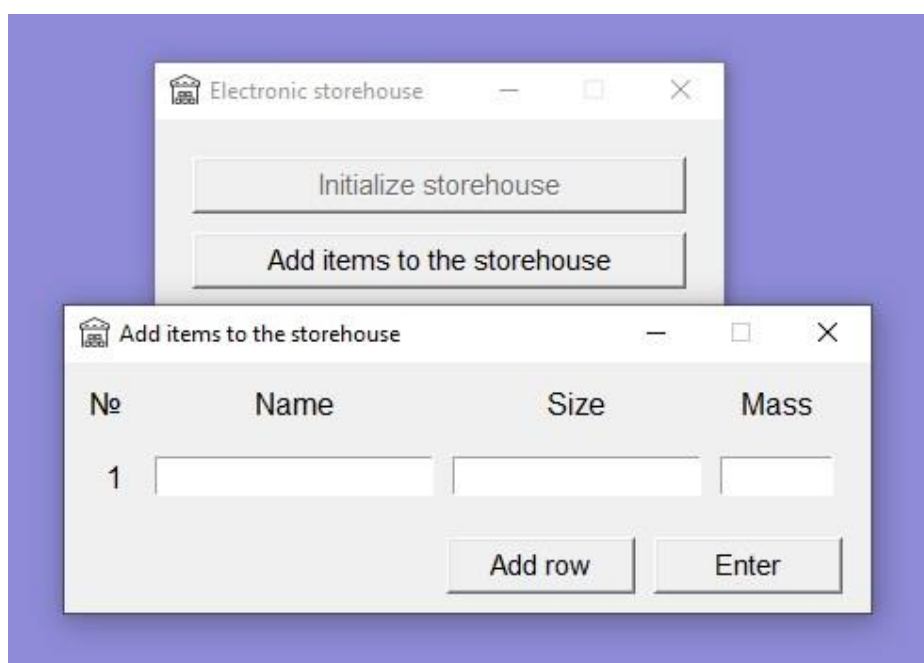
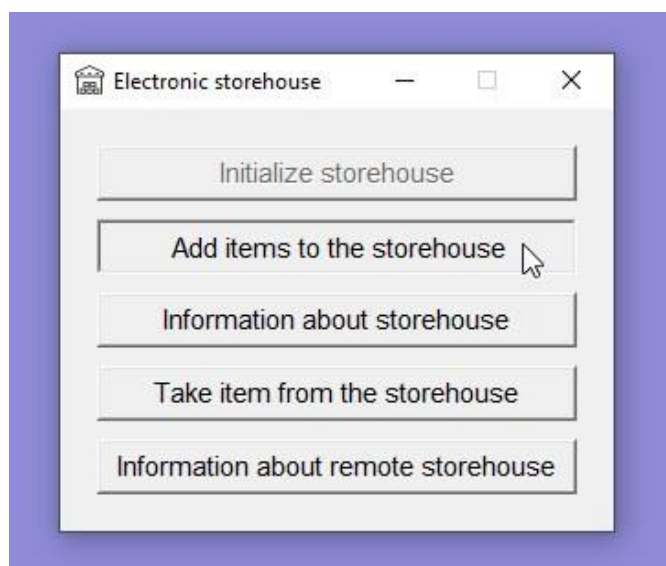
## Описание проведенных испытаний в соответствии с регламентом кейса.

1. Производится запуск и инициализация сервиса, включающие в себя получения от API аппаратной части схемы ячеек.

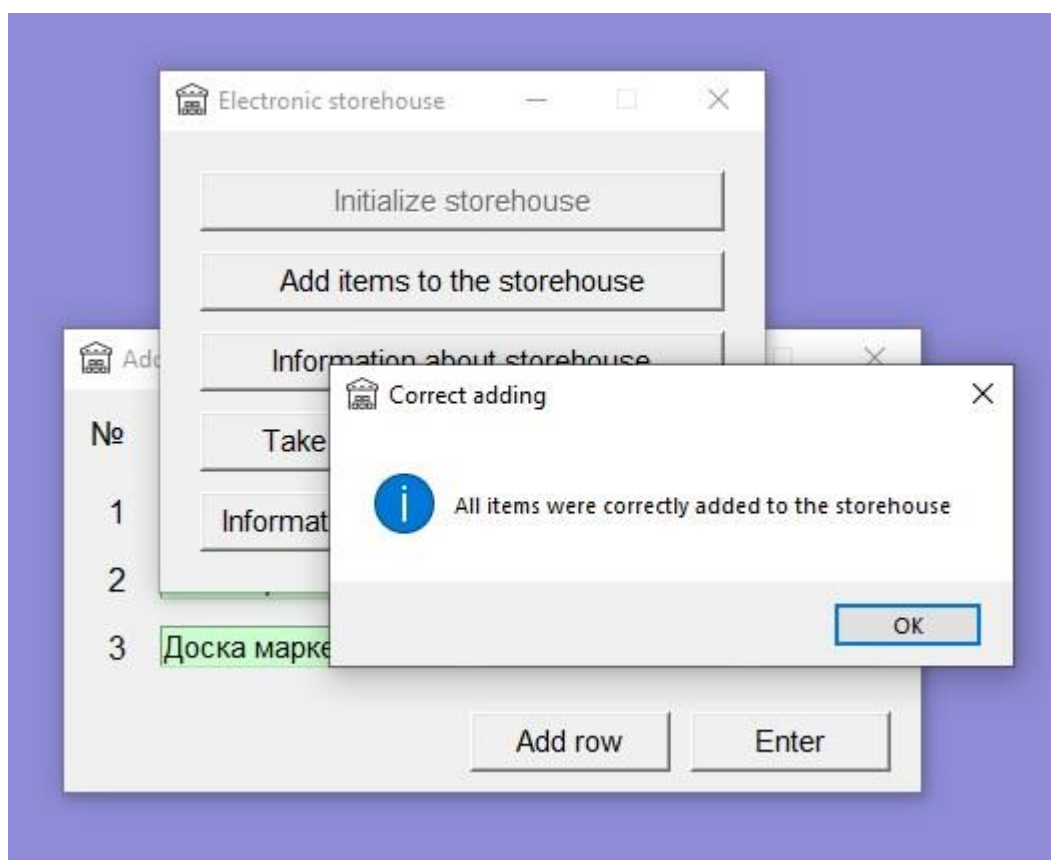
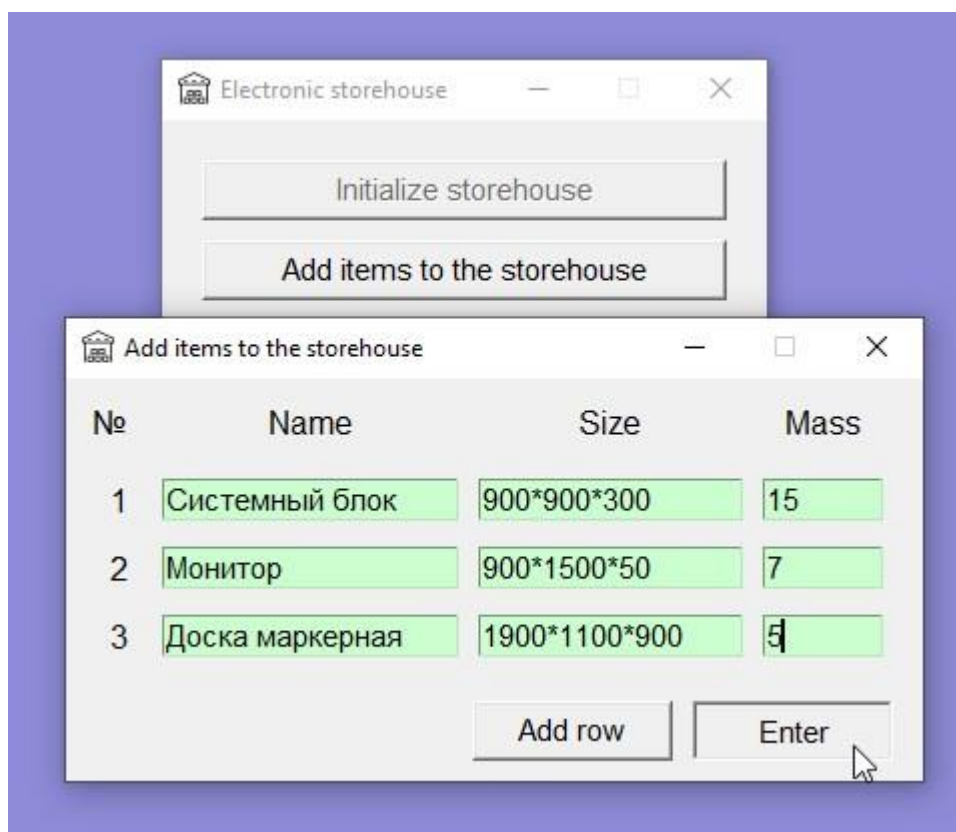
```
Командная строка - python app.py
D:\My Documents (D)\INF_PROJECTS\electronic-storehouse\API>python app.py
* Serving Flask app "app" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: on
* Restarting with stat
* Debugger is active!
* Debugger PIN: 197-954-872
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```



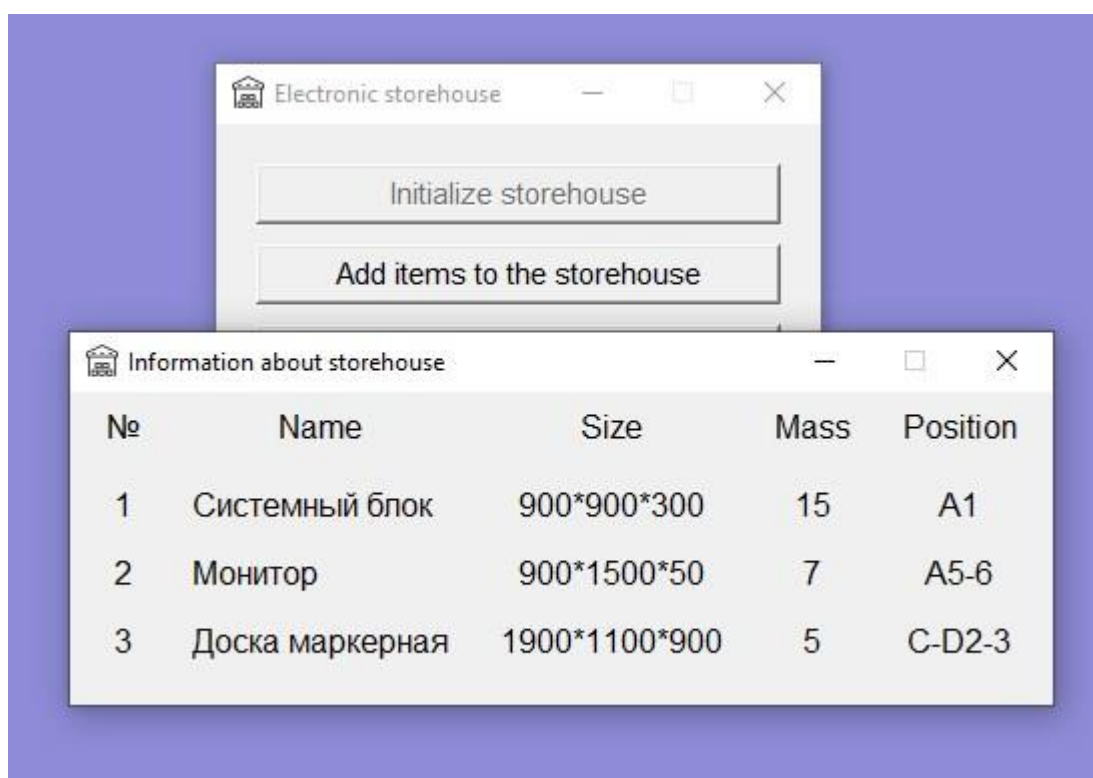
2. Участники демонстрируют добавление накладной посредством передачи сервису таблицы с позициями.





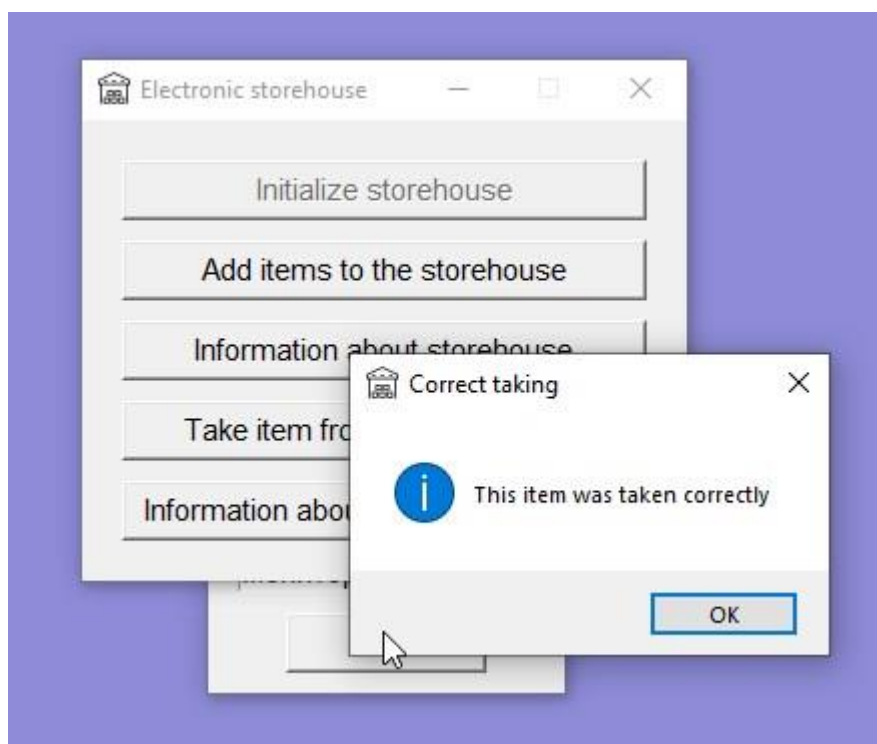


3. Участники демонстрируют список позиций, добавленных на склад.

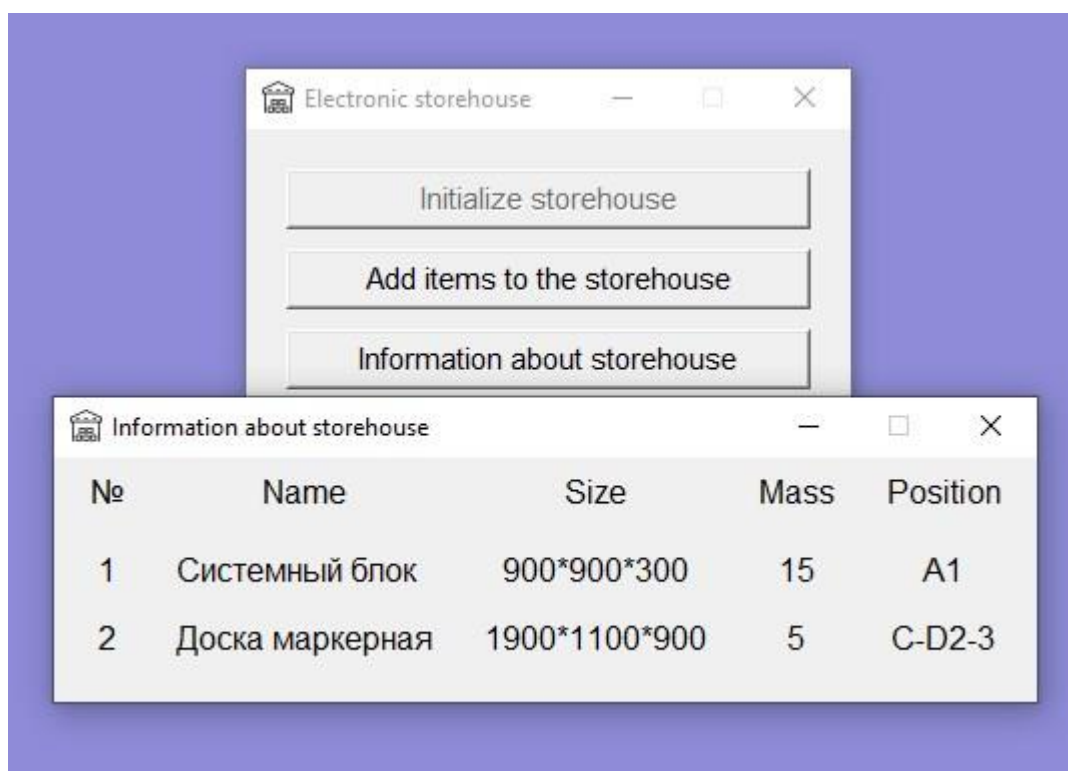


4. Участники демонстрируют выбор позиции со склада для выдачи.





5. Участники демонстрируют список позиций на складе после выдачи.



6. Участники демонстрируют список позиций, отправленных на удаленный склад.

