

Разработка библиотеки функций на языке Python, реализующей автоматизированное построение динамических графических пользовательских интерфейсов в рамках CMS Django

Василян Артур Размикович, студент группы РК6-73Б

var19r219@student.bmstu.ru

Научный руководитель: Соколов Александр Павлович

Место проведения: МГТУ им. Н. Э. Баумана

Кафедра: Системы автоматизированного проектирования (РК-6)



Россия, Москва, 2023 г.

Содержание

- Введение.
- Подходы к разработке пользовательского интерфейса.
- Постановка задачи.
- Разработка тестового web-приложения.
- Запуск web-приложения на тестовом сервере.
- Дальнейшие планы.
- Список литературы.

Введение

- **Интерфейс¹** — это совокупность средств методов и правил взаимодействия управления контроля и т.д. между элементами системы;
- **Пользовательский интерфейс¹** — это разновидность интерфейсов, в котором одна сторона представлена человеком-пользователем, другая — машиной-устройством.
- **Графический пользовательский интерфейс¹ (GUI)** — это разновидность пользовательского интерфейса, в котором элементы интерфейса меню кнопки значки списки и т.п., представленные пользователю на дисплее, исполнены в виде графических изображений.

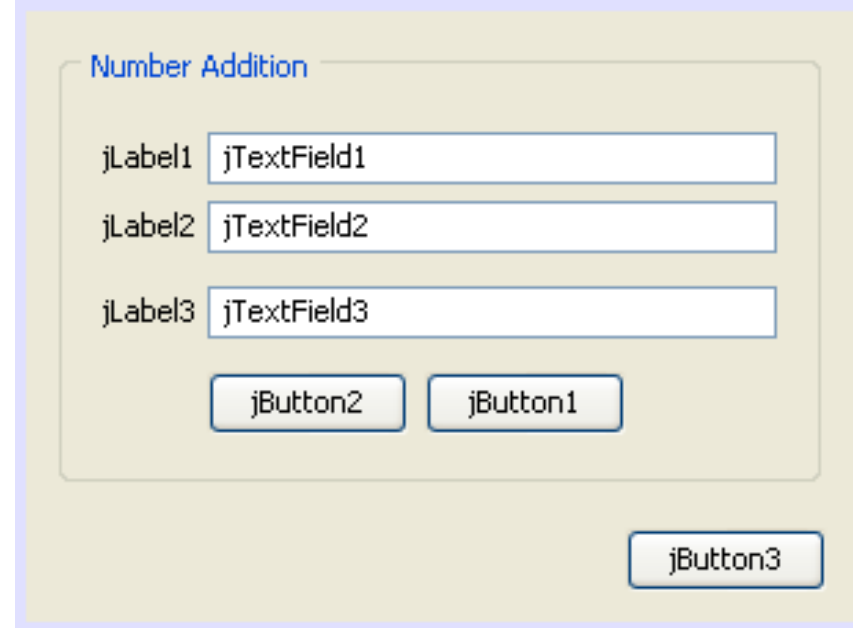


Рис. 1. Пример GUI

Подходы к разработке пользовательского интерфейса

При изучении статьи о **построении интерфейса на основе пользовательских целей**² были выделены два метода взаимодействия человека и ЭВМ:

1. **Ограничительный**. Пользователь должен иметь необходимые знания для того, чтобы самому планировать ход выполнения своего задания из предоставляемых ему операций.
2. **Направляющий**. Каждая из целей соответствует определенному пользовательскому заданию, которое может выполнить ЭВМ, взаимодействуя с пользователем.

Методический подход к созданию средства построения пользовательского интерфейса³.

1. Мониторинг действий оператора.
2. Применения типовой системы показателей качества. Обеспечивается оценка качества пользовательского интерфейса.
3. Автоматизация программирования и документирования пользовательского интерфейса (возможность автоматизированного документирования интерфейса программы).
4. Отображение некоторого абстрактного сценария осуществляет механизм его интерпретации в стандартные программные процедуры.

Так же существует метод **построение пользовательского интерфейса с использованием интерактивного машинного обучения**⁴.

Постановка задачи

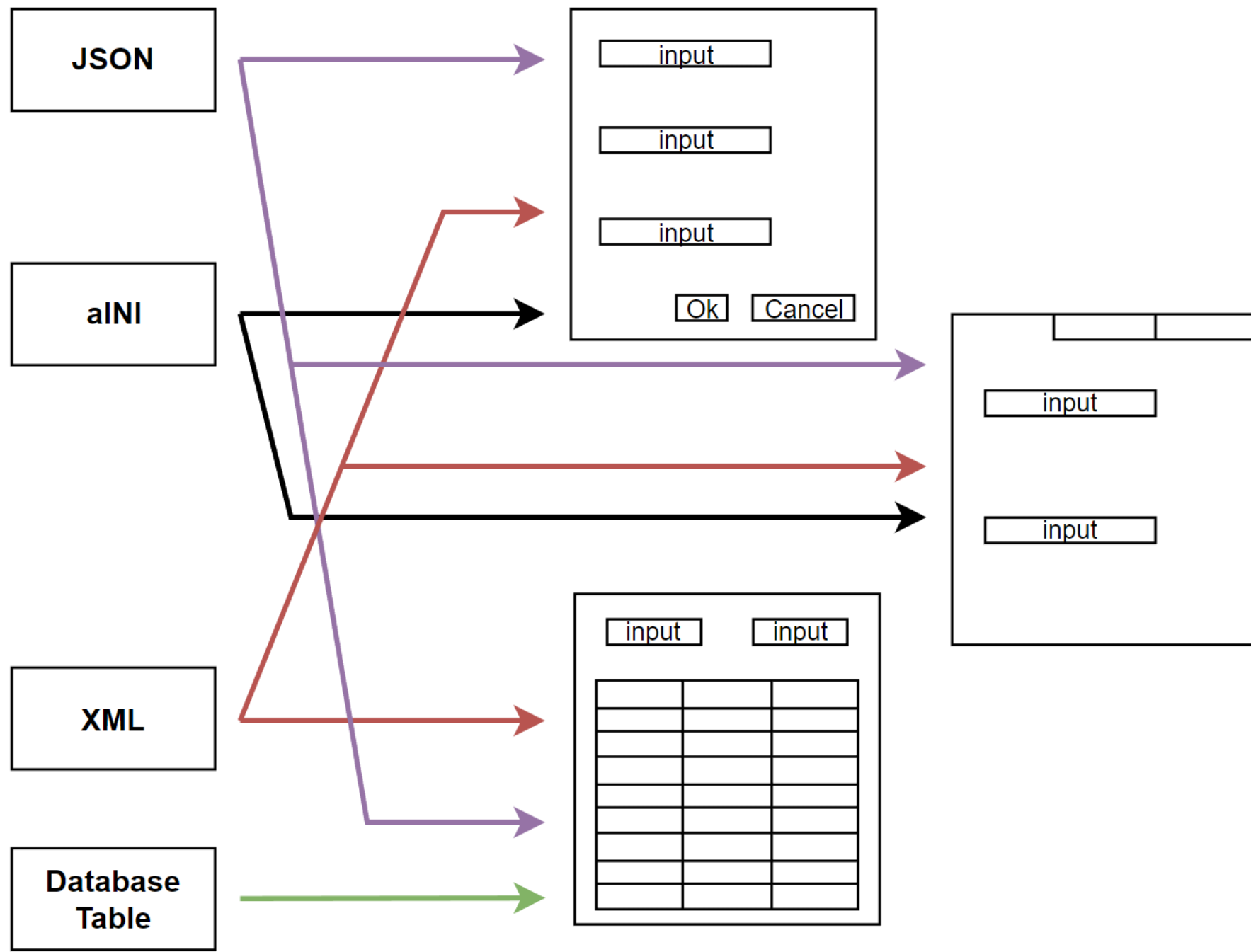
Цель: разработать библиотеку, обеспечивающую автоматизацию построения динамических пользовательских интерфейсов и создать основу для разработки web-ориентированного препроцессора инженерной программной системы.

Задачи:

- Рассмотреть существующие подходы разработки GUI.
- Разработать тестовое web-приложение.
- Запустить разработанное web-приложение на тестовом сервере.
- Изучение существующего web-приложения comwrc.
- Доработка библиотеки русomsdk в части возможности генерации GUI.
- Интеграция разработки в состав web-приложения comwrc и тестирование.

Постановка задачи

Различные варианты GUI



Разработка тестового web-приложения

Были использованы **Django** и **Docker**.

Django — это Python web-фреймворк, с помощью которого можно вести разработку web-приложения.

Docker — программное обеспечение, применяемое для разработки web-приложений в средах с поддержкой контейнеризации.

Docker-образ — шаблон для создания Docker-контейнеров.

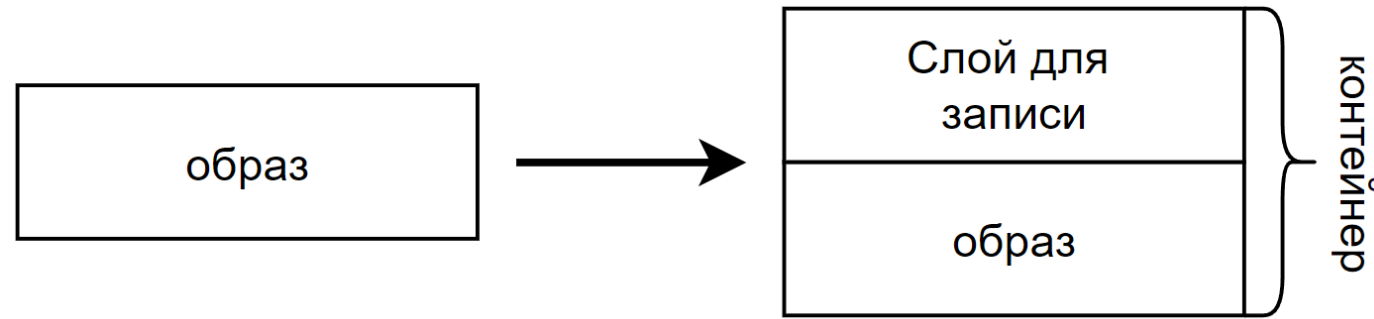


Рис. 2. Создание контейнера на основе образа

Разработка тестового web-приложения

`urlpatterns` – список всех URL, которые обрабатываются web-приложением.

`image_upload` – функция-представления, которая сопоставлена с адресом URL с помощью функции `path()` и которая обрабатывает запрос по этому адресу.

```
...
urlpatterns = [
    path("", image_upload, name="upload"),
    path("admin/", admin.site.urls),
]
...
```

Листинг 1. Содержимое `urls.py`

В данном случае в функции-представления используется html файл (`upload.html`), который так раз можно генерировать и заменять, если потребуется.

Но в случае изменения взаимодействия пользователя со страницей, помимо html файла требуется так же переписать функцию-представления.

```
def image_upload(request):
    if request.method == "POST" and request.FILES["image_file"]:
        image_file = request.FILES["image_file"]
        fs = FileSystemStorage()
        filename = fs.save(image_file.name, image_file)
        image_url = fs.url(filename)
        print(image_url)
        return render(request, "upload.html", {
            "image_url": image_url
        })
    return render(request, "upload.html")
```

Листинг 2. функция-представления `image_upload` в `views.py`

Разработка тестового web-приложения

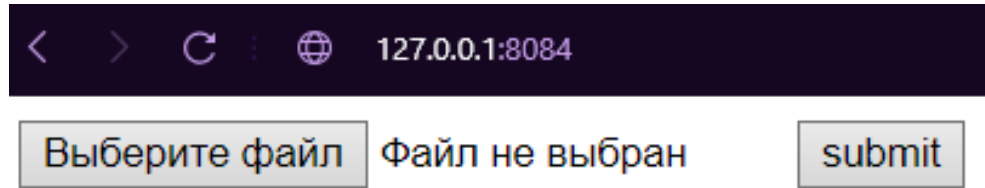


Рис. 3. Работа web-приложения

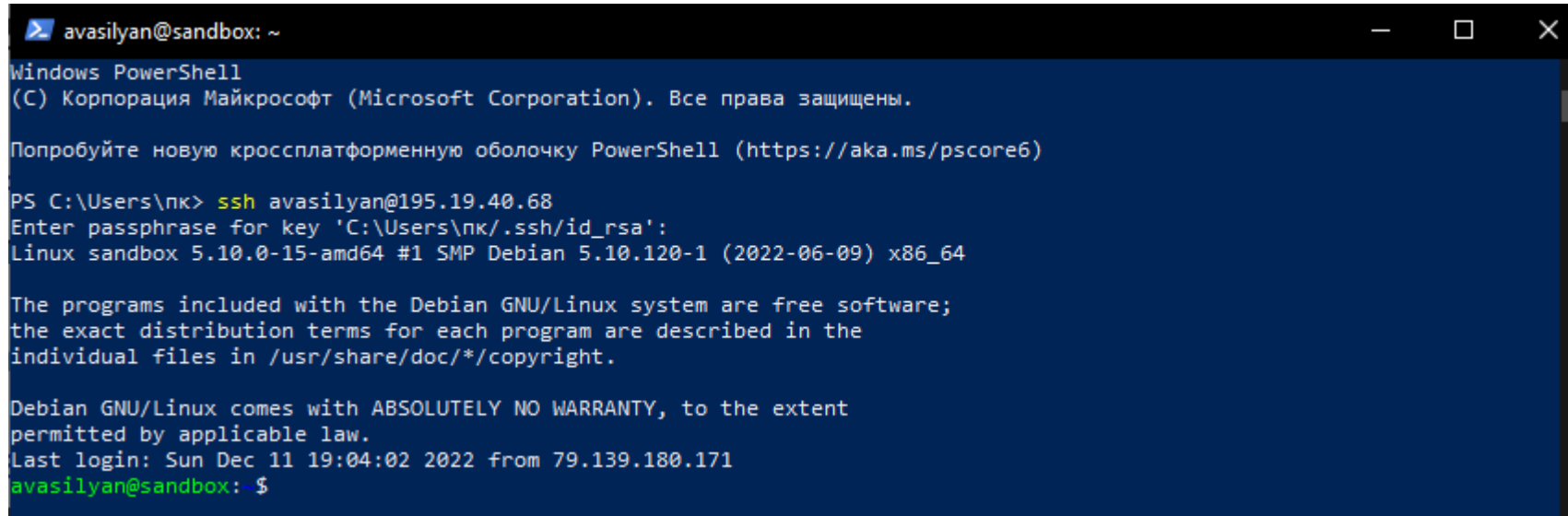
```
PS D:\PyCharm\django-on-docker\app> docker-compose exec db psql --username=hello_django --dbname=hello_django_dev
psql (13.0)
Type "help" for help.

hello_django_dev=# \l

               List of databases
  Name          | Owner      | Encoding | Collate  | Ctype    | Access privileges
-----+-----+-----+-----+-----+-----
hello_django_dev | hello_django | UTF8     | en_US.utf8 | en_US.utf8 | 
postgres        | hello_django | UTF8     | en_US.utf8 | en_US.utf8 | 
template0       | hello_django | UTF8     | en_US.utf8 | en_US.utf8 | =c/hello_django +
                |              |          |            |            | hello_django=Ctc/hello_django
template1       | hello_django | UTF8     | en_US.utf8 | en_US.utf8 | =c/hello_django +
                |              |          |            |            | hello_django=Ctc/hello_django
(4 rows)
```

Рис. 4. Проверка таблиц

Запуск web-приложения на тестовом сервере



```
avasilyan@sandbox: ~  
Windows PowerShell  
(C) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.  
  
Попробуйте новую кроссплатформенную оболочку PowerShell (https://aka.ms/pscore6)  
  
PS C:\Users\нк> ssh avasilyan@195.19.40.68  
Enter passphrase for key 'C:\Users\нк\.ssh/id_rsa':  
Linux sandbox 5.10.0-15-amd64 #1 SMP Debian 5.10.120-1 (2022-06-09) x86_64  
  
The programs included with the Debian GNU/Linux system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.  
  
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent  
permitted by applicable law.  
Last login: Sun Dec 11 19:04:02 2022 from 79.139.180.171  
avasilyan@sandbox:~$
```

Рис. 5. Вход на сервер

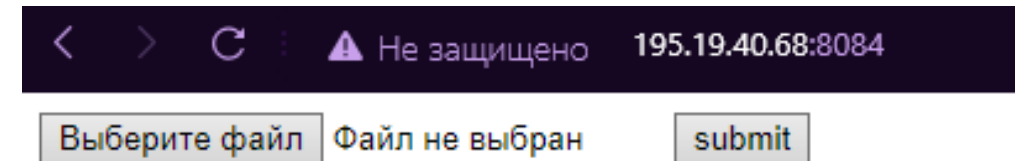


Рис. 6. Работа web-приложения на тестовом сервере

Дальнейшие планы

- Изучение существующего web-приложения comwps (проект по разработке web-клиента для доступа к подсистемам PBC GCD и другим программным системам).
- Доработка библиотеки ruscomsdk (SDK для программных реализаций сложных вычислительных методов в рамках графоориентированной технологии GBSE) в части возможности генерации GUI.
- Интеграция разработки в состав web-приложения comwps и тестирование работоспособности созданных программных средств.

Список литературы

- [1] Лукьянов Д. В. Разработка графического пользовательского интерфейса // Новые информационные технологии в автоматизированных системах. 2012
- [2] Санковский Ю.Е. Метод построения оконного интерфейса пользователя на основе моделирования пользовательских целей. 1998
- [3] Казаков Г.В., Корянов В.В., Чемирисов В.В., Уваров А.В. Методический подход к созданию универсального пользовательского интерфейса // Наука и инновации. 2020. №11.
- [4] Юркин В.А., Сараджишвили С.Э. Построение пользовательского интерфейса с использованием интерактивного машинного обучения // СОВРЕМЕННАЯ НАУКА: АКТУАЛЬНЫЕ ПРОБЛЕМЫ ТЕОРИИ И ПРАКТИКИ. 2020. №3.