

# **АВТОМАТИЗАЦІЯ ЗБОРУ ДАНИХ З ВИКОРИСТАННЯМ DJANGO REST FRAMEWORK ДЛЯ ФІНАНСОВИХ ГАЛУЗЕЙ**

Вінницький національний технічний університет

## **Анотація**

*В розглянуто основні підходи до розробки серверної частини платформи конфігурації, зосереджені на використанні REST API та Django Rest Framework. В результаті були вибрані оптимальні засоби для створення системи та розроблена відповідна архітектура додатку. При реалізації було зосереджено увагу на розробці функціоналу авторизації та аутентифікації користувачів, а також розроблено REST API для забезпечення взаємодії з клієнтською частиною додатку.*

**Ключові слова:** REST API, Django Rest Framework, серверна частина додатку, авторизація та аутентифікація.

## **Abstract**

*The thesis considered the main approaches to the development of the server part of the configuration platform, focusing on the use of the REST API and the Django Rest Framework. As a result, the optimal tools for creating the system were selected and the appropriate architecture of the application was developed. During the implementation, attention was focused on the development of the user authorization and authentication functionality, as well as a REST API was developed to ensure interaction with the client part of the application.*

**Keywords:** REST API, Django Rest Framework, server-side of the application, authentication and authorization.

## **Вступ**

Збільшення техніки в повсякденному житті призвело до гострої потреби для зберігання великої кількості інформації, від автоматичного збору даних про наш сон фітнес трекерами до збору даних для налаштування точної імовірності випадіння розіграшу. Ручний збір інформації є часо- та працезатратним процесом, який до того ж є схильним до помилок, машина в свою чергу не має людського фактору і відразу прораховує помилки та не дозволяє завершити збереження завідома не правильних даних завдяки процесу валідації який закладається розробниками на етапі розробки та покращується під час всього циклу життя продукту. [1-5]

Існують рішення що мають аналогічні можливості для клієнта, але вони є обмеженими в функціоналі і дозволяють використання лише з іншим програмним забезпеченням платформи [2, 4, 6]. Тому робота з розробки сервісу який буде відповідати вимогам клієнта і не обмежувати його в використанні програмного забезпечення є актуальною.

## **Результати досліджень**

Було розглянуто провідні рішення на ринку надання послуг розіграшів для платформ. Виявлено основну проблему готових рішень, такі сервіси обмежені в налаштуваннях, в одному випадку через наявність готових ігрових рішень оператор вимушений для використання системи прийняття рішень цієї компанії інтегрувати також і інші ігрові рішення, в іншому випадку ускладнений процес додавання необхідного функціоналу через процедуру розгляду і ухвалення необхідних оновлень самою компанією і відносно не велика кількість налаштувань що безпосередньо впливають на накопичення самого розіграшу та вірогідність його виграшу, частоту і т.д.

Мовою програмування було обрано Python. Для мови програмування Python існує дві найбільш популярні середовища розробки PyCharm та VS Code. Обидва реалізують максимально схожий функціонал але мають різний формат взаємодії починаючи з графічного інтерфейсу закінчуючи розкладкою гарячих клавіш. Загалом вибір зводиться до зручності для розробника та можливостей додавання додаткових плагінів. Було обрано VS Code через більшу кількість плагінів що дозволяють розширити функціонал середовища розробки.

Також було обрано RestAPI - стиль інтерфейсу, що використовує HTTP запити, щоб отримати доступ до використання даних. Такі дані дозволяють використовувати типи даних GET, PUT, POST і DELETE, функції яких полягають в читанні, оновленні, створенні та видаленні операцій відносно ресурсів.

Архітектура проекту була розроблена з урахуванням принципів модульності, масштабованості та безпеки. Використання Django Rest Framework дозволяє ефективно обробляти дані, та зберігати їх забезпечуючи автентифікацію та авторизацію користувачів.

Алгоритм розробленої програми наведено на рисунку 1. Схема роботи фреймворку з базою даних даведена на рисунку 2.

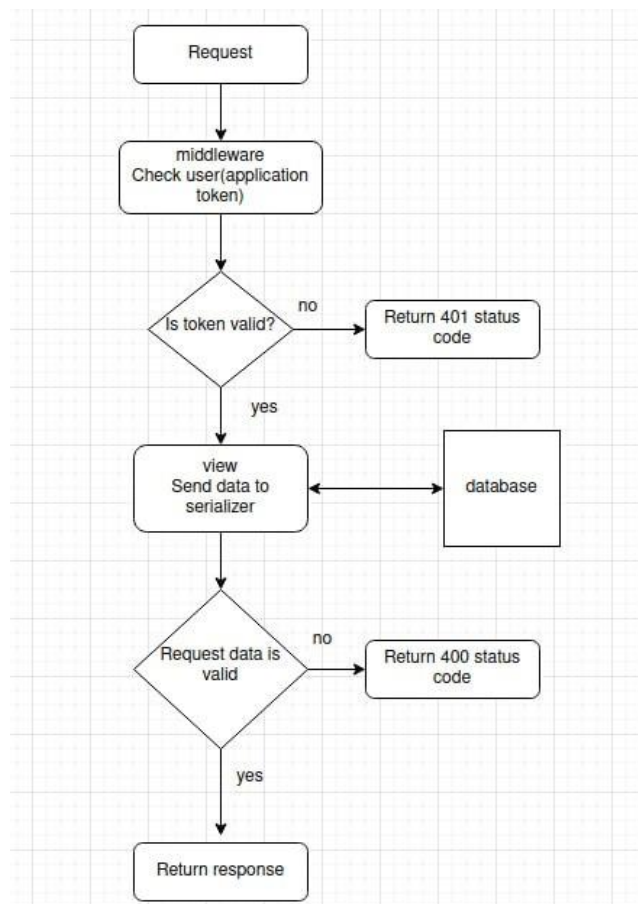


Рисунок 1 - Алгоритм роботи програми

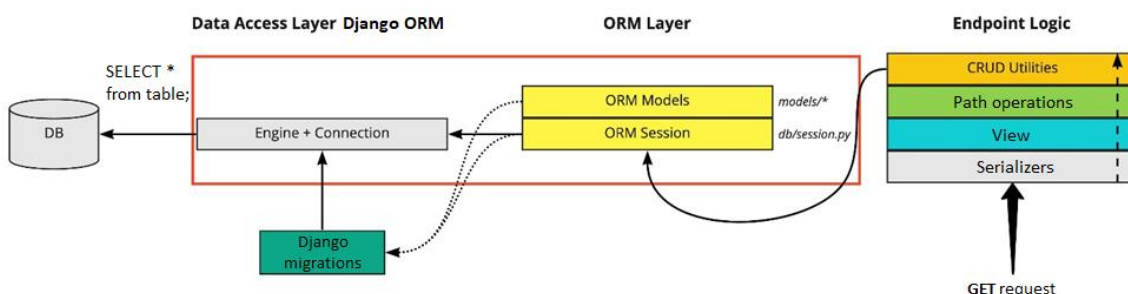


Рисунок 2 - Схема роботи фреймворку з базою даних

В результаті роботи було розроблено систему авторизації користувачів, системи авторизації

додатків, базову архітектуру, логіку збереження конфігурації платформи, обробник запиту на отримання резюме, обробник для створення конфігурації на стороні DSS.

### Висновки

Основними функціональними вимогами до платформи збору конфігурації були забезпечення можливості збору даних зі швидкою валідацією та серіалізацією тощо. За допомогою REST API було забезпечено зручний та стандартизований доступ для цілей інтеграції.

Використання PostgreSQL як реляційної бази даних було обрано з огляду на його гнучкість, масштабованість та швидкодію. PostgreSQL дозволяє ефективно зберігати та опрацьовувати великі обсяги даних, що є важливим для платформ конфігурації через додавання різними операторами великої кількості конфігурації та їх частого оновлення.

Розроблена серверна частина банківського додатку з використанням REST API та Django Rest Framework є потужним та безпечним інструментом для забезпечення функціональності та надійності. Результати дослідження свідчать про переваги використання цих технологій та можливість успішного впровадження системи в різних проектах фінансової галузі.

### СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Python. Python documentation. 2023 URL: <https://docs.python.org/3/> (дата звернення 21.05.2023)
2. FastAPI. FastAPI. URL: <https://fastapi.tiangolo.com/> (дата звернення 21.05.2023)
3. Django. Django REST framework. URL: <https://www.django-rest-framework.org/> (дата звернення 21.05.2023)
4. Django. Django. 2023. URL: <https://www.djangoproject.com/> (дата звернення 21.05.2023)
5. JetBrains. JetBrains. 2023. URL: <https://www.jetbrains.com/pycharm/> (дата звернення 21.05.2023)
6. Visual Studio Code. Visual Studio Code. 2023 URL: <https://code.visualstudio.com/> (дата звернення 21.05.2023)
7. Hightload. Як удосконалити Web3 UX. Безплатний онлайн-інвент. 2021-2023. URL: <https://highload.today/uk/subd-yaki-buvayut-yak-vibrati/> <https://isocpp.org/>

**Горбаченко Михайло Сергійович** — студент групи ІАКІТ-196, кафедра автоматизації та інтелектуальних інформаційних технологій, факультет інтелектуальних інформаційних технологій та автоматизації, Вінницький національний технічний університет, м. Вінниця

**Мальований Дмитро Вадимович** – студент групи ІСТ-22м, факультет інтелектуальних інформаційних технологій та автоматизації, Вінницький національний технічний університет, Вінниця, e-mail: dmytro.maliovani@gmail.com

**Богач Ілона Віталіївна** – к.т.н., доцент кафедри автоматизації та інтелектуальних інформаційних технологій, факультет комп'ютерних систем і автоматики, Вінницький національний технічний університет, м. Вінниця, e-mail: ilona.bogach@gmail.com

**Horbachenko Mykhailo S.** – student of ІАКІТ-19В group, Faculty of Intelligent Information Technology and Automation, Vinnytsia National Technical University, Vinnytsia, e-mail: oleksii.tkachuk878@gmail.com

**Maliovani Dmytro V.** – student of the ІСТ-22m group, Faculty of Intelligent Information Technology and Automation, Vinnytsia National Technical University, Vinnytsia, e-mail: dmytro.maliovani@gmail.com

**Bogach Ilona V.** – PhD, Associate Professor of Automation and Intelligent Information Technologies, Faculty of Computer Systems and Automatics, Vinnytsia National Technical University, Vinnytsia, e-mail: ilona.bogach@gmail.com