МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ (национальный исследовательский университет)»

Институт №3 «Системы управления, информатика и электроэнергетика»

Кафедра № 304 «Вычислительные машины, системы и сети»

Технологии разработки программно-информационных систем

Отчет по практическому заданию № 6

Архитектуры и паттерны

Вариант «Добавление наблюдателя или медиатора/посредника.»

Выполнил студент группы М3О-108СВ-25

Давыдов А.П.

Принял:

Титов Ю.П.

Москва, 2025

Задание

Практическая работа нацелена на изучения основ проектирования архитектуры, изменения, модернизации кода для удобства модификации и понимания кода программного обеспечения. В рамках практической работы необходимо составить архитектуру (структуру) разрабатываемого в творческой работе программного обеспечения, выделить отдельные компоненты, блоки, указать паттерны проектирования. В отчете по практической работе рассмотреть архитектурные решения с точки зрения принципов SOLID. Кроме составленной схемы, которая может быть составлена коллективом разработчиков или выполняться в рамках творческого задания (отдельных task в системе управления проектами), студентам необходимо предложить модернизацию программного кода путем добавления необходимых паттернов, в данном варианте: **Добавления наблюдателя или медиатора/посредника.**

Процесс выполнения

Большую часть архитектурной работы в творческом проекте взял на себя фреймворк Django. По своей сути, команде разработке всего-лишь надо сделать Front-end, связать поля оттуда с нужными значениями, и затем ещё связать с Back-end, то-есть логикой БД и некоторыми вычислениями.

Основной паттерн в Django – это Model-Template-View, а также в Django имеются декораторы, например, @login_required, не позволяющий просматривать страницы без входа в систему.

Рассмотрю MVC с точки зрения SOLID:

- Single Responsibility: Model, View и Controller отвечает каждый за свою часть. Model – это доступ к данным, View – HTTP-Запросы, Controller (Template) – Внешний вид, отображение.
- Open-Closed: Архитектурно MVC/MTV в Django позволяет добавлять новые модели, views и URL-маршруты, не модифицируя существующие компоненты
- Liskov Substitution: наследники Model должны поддерживать ORM-операции и возможности для save/delete.
  Наследники generic class-based views (например, `ListView`, `CreateView`) обязаны сохранять ожидаемое поведение: возвращать корректный HTTP-ответ, работать с контекстом, шаблоном и т.п.
- Iterface Segregation: В контексте Django, языка Python и MVT/MVC совпадает с Single Responsibility
- Dependency Inversion: Модели не зависят от внешнего слоя, а слой зависит от их описаний. Это порождает трудности с миграциями, но позволяет создавать модели с меньшими ограничениями.

Предлагаемая модернизация:

Шаблон «Наблюдатель» является одним из очень хороших решений для согласования UI с логикой. Кроме того, этому способствует сама модель MVT.

Зачастую паттерны проектирования уже заложены в сам язык. Так, в C#, многие заготовленные делегаты уже являют собой готовое решение для реализации паттерна.

Делается это достаточно просто. Сначала надо зарегистрировать возможность использования наблюдателя:

```
from django.apps import AppConfig

class TRPISConfig(AppConfig):
    def ready(self):
        import myapp.signals
```

Затем поверх функций обновления можно добавлять расширяемую логику в следующем формате:

```
@receiver(post_save, sender=User)
def notify_profile_updated(sender, instance, created, **kwargs):
    if not created:
        {Логика}
```

Декоратор @reciever уже определяет, что эта функция слушает сигнал.

Таким образом можно создавать множество разных функций, слушающих одну и ту же систему, а поскольку Django с Python закрывают большую часть функционала, здесь будет достаточно сделать подписанные функции.

В них можно добавить в любом модуле (То-есть без зависимостей), даже планируемом и не до конца определённом:

- Уведомление пользователей об изменении рейтинга (Оценок)

- Новые отзывы
- Создание метрики и аналитики
- И даже обновление связанных моделей