

Министерство образования Республики Беларусь  
Учреждение образования  
«Брестский государственный университет»  
Кафедра ИИТ

**Отчет по лабораторной работе 3  
Дисциплина “ПИС”**

**Выполнила:**

Студентка группы ПО-3

Ковалева А. И.

**Проверил:**

Лаврущик А. И.

Брест 2021

## **Вариант 12**

**Цель работы.** Познакомиться с гексагональной архитектурой и проектированием систем на её основе.

**Задание для выполнения.** Определите структуру вашего приложения (ЛР №1–2) в соответствии с архитектурным стилем «гексагональная архитектура». Разработайте файловую структуру для всех уровней, примерно представьте, какие сущности, репозитории, сервисы вам понадобятся, где необходима инверсия зависимостей и т.д. Конкретную реализацию на каждом уровне вы будете делать на последующих работах.

**Предметная область.** Блог о фермерском хозяйстве.

### **Ход работы**

```
|---- Config  
    |---- Packages  
    |---- Routes  
|---- Migrations  
|---- Entity  
    |    |---- User  
    |    |---- Note  
|---- Repository  
    |    |---- UserRepository  
    |    |---- NoteRepository  
|---- Config  
    |    |---- Style  
|---- Controller  
    |    |---- UserController  
    |    |---- NoteController  
|---- Form  
    |---- NoteType
```

1. Config — конфигурация проекта которую можно подставлять в зависимости от изменения среды.
2. Migrations — упорядоченный набор миграций БД.
3. Controller — слой представления, организует связь от запроса на ответ к нему.
4. Entity — модели данных, соответствующие хранимым в БД.
5. Repository — слой для работы с внешними данными, такими как содержимое локальной БД либо внешний сервис.
6. Form — веб-формы для entity, позволяющие быстро создавать сущности и заполнять их предопределенными данными.
7. Файл .env — переменные окружения определяющие ключевые аспекты

Контроллеры:

1. UserController.php – контроллер для обработки всех запросов пользователя.
2. NoteController.php – контроллер для обработки всех запросов заметок.

Модели:

1. User – модель пользователя.
2. Note – модель заметки.

Репозитории:

1. UserRepository – репозиторий доступа к данным пользователя.
2. NoteRepository – репозиторий доступа к данным заметки.

Формы:

1. NoteType – форма быстрого создания заметки.

Инверсия зависимостей необходима для репозиториев UserRepository и NoteRepository, т.к. конкретные их реализации (для конкретных СУБД, например) будут реализованы отдельно.

**Вывод:** В данной лабораторной работе я познакомилась с гексагональной архитектурой и проектированием систем на её основе.