

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

Лабораторная работа №3

По дисциплине «ПИС»
за 5-й семестр

Выполнил:
студент 2 курса
группы ПО-3 (1)
Афанасьев В.В.

Проверил:
Лаврущик А.И.

Цель работы: познакомиться с гексагональной архитектурой и проектированием систем на ее основе.

«Гексагональная архитектура»

Вариант: 2

Задание:

Задание на выполнение: определите структуру вашего приложения (ЛР №№1-2) в соответствии с архитектурным стилем «гексагональная архитектура». Разработайте файловую структуру всех уровней, примерно представьте, какие сущности, репозитории, сервисы вам понадобятся, где необходима инверсия зависимостей и т.д.

Предметная область: продажа билетов на мероприятия

Ход работы:

Папка **CONTROLLERS** обрабатывает реакции на пользовательские действия, такие как отправка форм. Предварительно определено 3 контроллера:

ClientController – необходим для работы с клиентами

EventController – необходим для работы с мероприятиями

TicketController – необходим для проведения операций по покупке/продаже билетов на мероприятия

Папка **ENTITIES** содержит необходимые сущности, было определено 3:

Client – модель, содержащая информацию о клиенте системы

Event - модель, содержащая информацию о мероприятии (название и время проведения)

Ticket - модель, содержащая информацию о билете (информация о клиенте, купившем билет, информация о мероприятии, стоимость)

Папка **REPOSITORIES /INTERFACES** содержит интерфейс для работы с базой данных, а конкретно – с таблицами: дает возможность обмена информацией с таблицами – чтение и запись.

IClientRepository – интерфейс для взаимодействия с таблицей клиентов

IEventRepository - интерфейс для взаимодействия с таблицей мероприятий

ITicketRepository - интерфейс для взаимодействия с таблицей билетов

Папка **REPOSITORIES** содержит конкретную реализацию под конкретную СУБД.

ClientRepository – механизм для взаимодействия с таблицей клиентов в PostgreSQL

EventRepository – механизм для взаимодействия с таблицей мероприятий в PostgreSQL

TicketRepository – механизм для взаимодействия с таблицей билетов в PostgreSQL

Папка **APPLICATION** содержит файлы с логикой приложения:

ClientRegistrationService – запись клиента в систему

EventService – запись мероприятия в систему

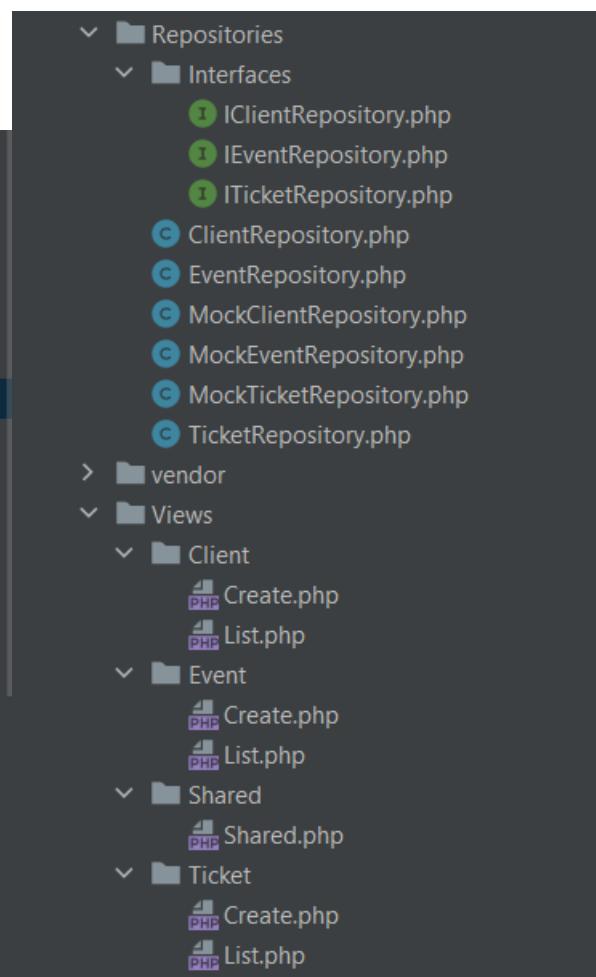
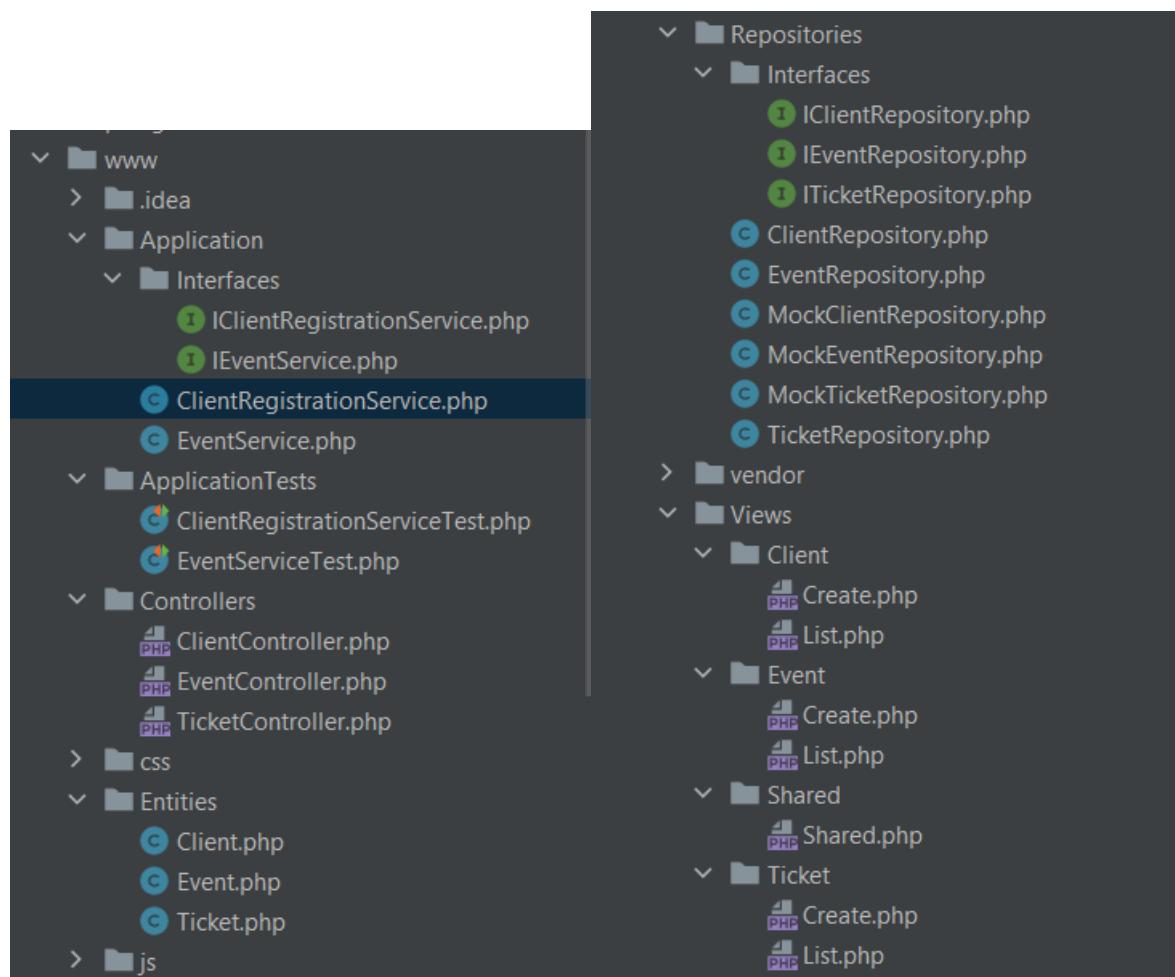
Папка **VIEWS** содержит разметку документов:

Client (Create/List) – форма для работы с клиентами

Event (Create/List) – форма для работы с мероприятиями

Ticket (Create/List) – форма для работы с билетами на мероприятия

Shared – стартовая форма



Выводы: в ходе выполнения лабораторной работы были ознакомлены с гексагональной архитектурой и проектированием систем на ее основе.