|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

**Факультет «Информатика и системы управления»**

**Кафедра ИУ5 «Системы обработки информации и управления»**

Домашнее задание

по дисциплине «Технологии разработки программного обеспечения»

Вариант № 86

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы ИУ5-15М |  | преподаватель каф. ИУ5 |
| Ролдугин Е.В. |  | Виноградова М. В. |
| Подпись и дата: |  | Подпись и дата: |

Москва 2022

**Цель работы:**

Изучить основные паттерны проектирования, их особенности и область применения. Получить практические навыки программирования паттернов. Освоить технологию включения паттернов в собственную программу.

|  |  |  |
| --- | --- | --- |
| Паттерн бизнес-логики | Паттерн базы данных | альтернатива |
| service layer | table data gateway | наблюдатель (gof) |

**Задание:**

1) Реализовать в программе паттерны (по варианту) бизнес-логики и работы с БД.

2) Составить набор диаграмм классов и последовательностей, которые демонстрируют структуру и поведение программы.

3) Отдельно составить диаграммы классов и последовательностей для иллюстрации примененных паттернов.

**Условия задания:**

Предметная область: «Веб-сайт по подбору фильмов»

Паттерны к реализации: «service level» - БЛ, «table data gateway» - БД, «наблюдатель(gof)».

**Описание реализации**

**Table Data Gateway (Шлюз таблицы данных)**

* Объект выступает в качестве шлюза между данными в приложении и в БД
* Объект (Класс) содержит методы для доступа к отдельной таблице или
* представлению (view): выборка, обновление, вставка, удаление (CRUD) и
* вызов хранимых процедур.
* Один объект работает сразу со всеми записями в таблице.
* Возможен один шлюз на несколько таблиц
* Может быть Класс со статическими методами

**Service Layer (Слой служб)**

* Определяет границу между приложением и слоем сервисов, который
* образует набор доступных операций и управляет ответом приложения в каждой операции.
* Определяет для приложения границу и набор допустимых операций с точки зрения взаимодействующих с ним клиентских модулей.
* Он инкапсулирует бизнес-логику приложения, управляя транзакциями и управляя ответами в реализации этих операций.

**Наблюдатель**

Наблюдатель – это поведенческий паттерн проектирования, который создаёт механизм подписки, позволяющий одним объектам следить и реагировать на события, происходящие в других объектах.

Наблюдатель позволяет объекту текстового редактора оповещать другие объекты об изменениях своего состояния.

Список подписчиков составляется динамически, объекты могут как подписываться на определённые события, так и отписываться от них прямо во время выполнения программы.

Редактор не ведёт список подписчиков самостоятельно, а делегирует это вложенному объекту. Это даёт возможность использовать механизм подписки не только в классе редактора, но и в других классах программы.

+Издатели не зависят от конкретных классов подписчиков и наоборот.

+Вы можете подписывать и отписывать получателей на лету.

+Реализует принцип открытости/закрытости.

-Подписчики оповещаются в случайном порядке.

**Система добавления комментария к фильму**

Для реализации в примере была взята часть системы – система добавления комментария. Данная часть предназначена для использования клиентом-пользователем.

Рассмотрим основные классы, применяющиеся в программе:

**Интерфейс пользователя**

**Client-** класс содержит методы клиента.

+create\_comment()

+get\_film()

**Create\_comment-**добавление комментария

+create\_comment()

**Get\_film-**страница товара

+get\_film()

+show()

**Уровень бизнес-логики**

Для работы с паттерном Transaction Script для каждого сценария необходимо создавать свой класс с 1 методом. В системе будут доступны следующие функции: авторизация, создание отзыва, просмотр отзыва. Под каждый сценарий создаем отдельный класс: UserScript, CreateFeedbackscript, SchowFeedbackscript. Все классы наследуются от AbstractScript с методом run и переопределяют его.

**Leave\_Comment-**класс сценария добавления отзыва. Состоит из одного метода. В классе происходит валидация входных данных, создание комментария.

**Observer(gof) -** паттерн проектирования, который создаёт механизм подписки

Атрибуты

+id: int

+name: string

+category: string

+description: string

+price: int

Методы

+Subscribe()-подписать

+Get() - получить

+cacheCache() -очистить кеш

**Уровень работы с БД**

**Film -** класс реального объекта.

Атрибуты

+name: string

+genre: string

+description: string

+price: int

Методы

+get()

**CommentRecord**

Атрибуты

+id: int

+type: string

+date: string

+description: string

+UserID: int

+FilmID: int

Методы

+Setld-установка в классе идентификатора

+SetDesc() -установка в классе содержания

+SetDate()-установка в классе даты

+SetFilmID() -установка в классе идентификатора фильма

+GetUserID() -установка в классе идентификатора пользователя

+insert() -добавление нового отзыва (открывается соединение с бд, добавляются данные, закрывается соединение, возвращается id)

**UserAutentificationData**

Атрибуты

+id:int

+email: string

+password: string

+name: string

+type: string

Методы

+Setstrlog - строки логина и пароля

+SetstrBD()-установка строки данных

+GetbyEmail()-получение данных по идентификатору почты

+GetbyPassword() -получение по id

+GetPasswod- получение пароля

+GetId-получение id

+GetType-получение типа пользователя

**Диаграммы**

**Диаграмма классов системы**

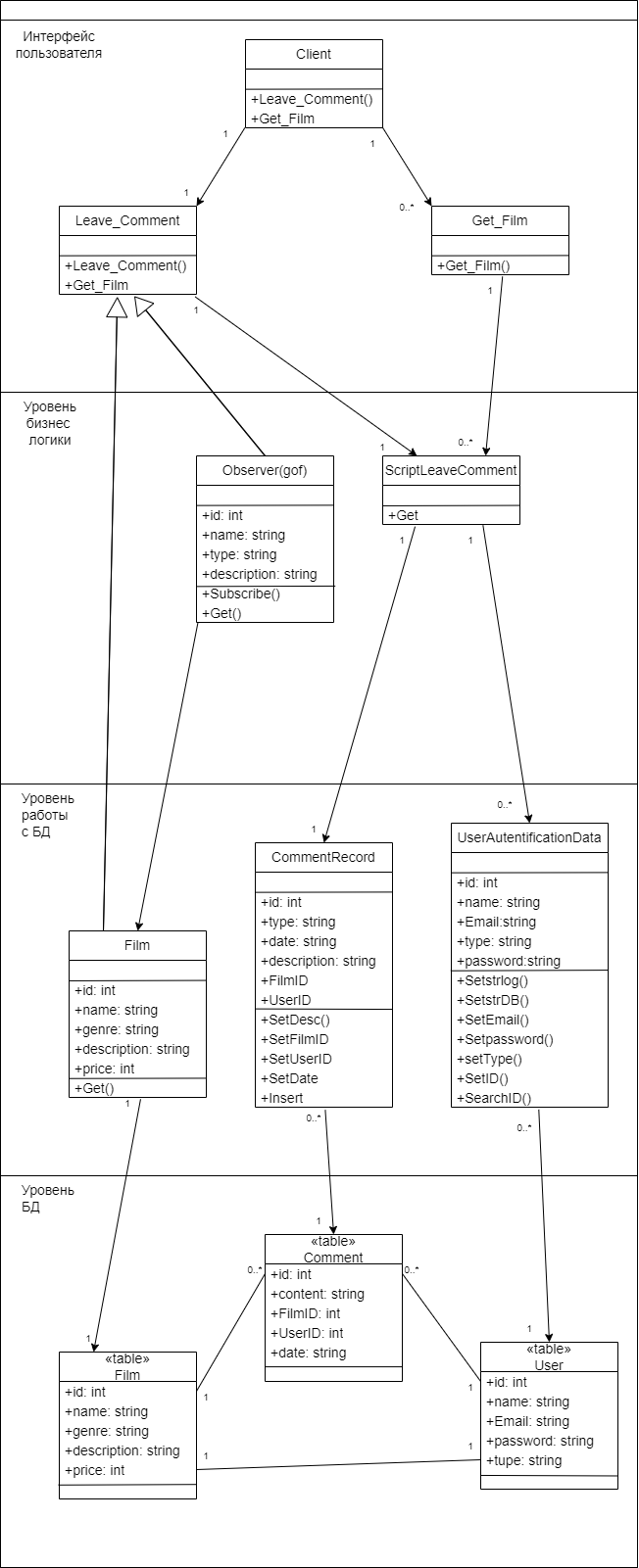


Рисунок 1. Диаграмма классов системы

На диаграмме классов системы видно использование паттерна бизнес-логики «service level» и паттерна работы с БД «table data gateway».

**Диаграмма последовательностей прецедента «Оставить комментарий»**

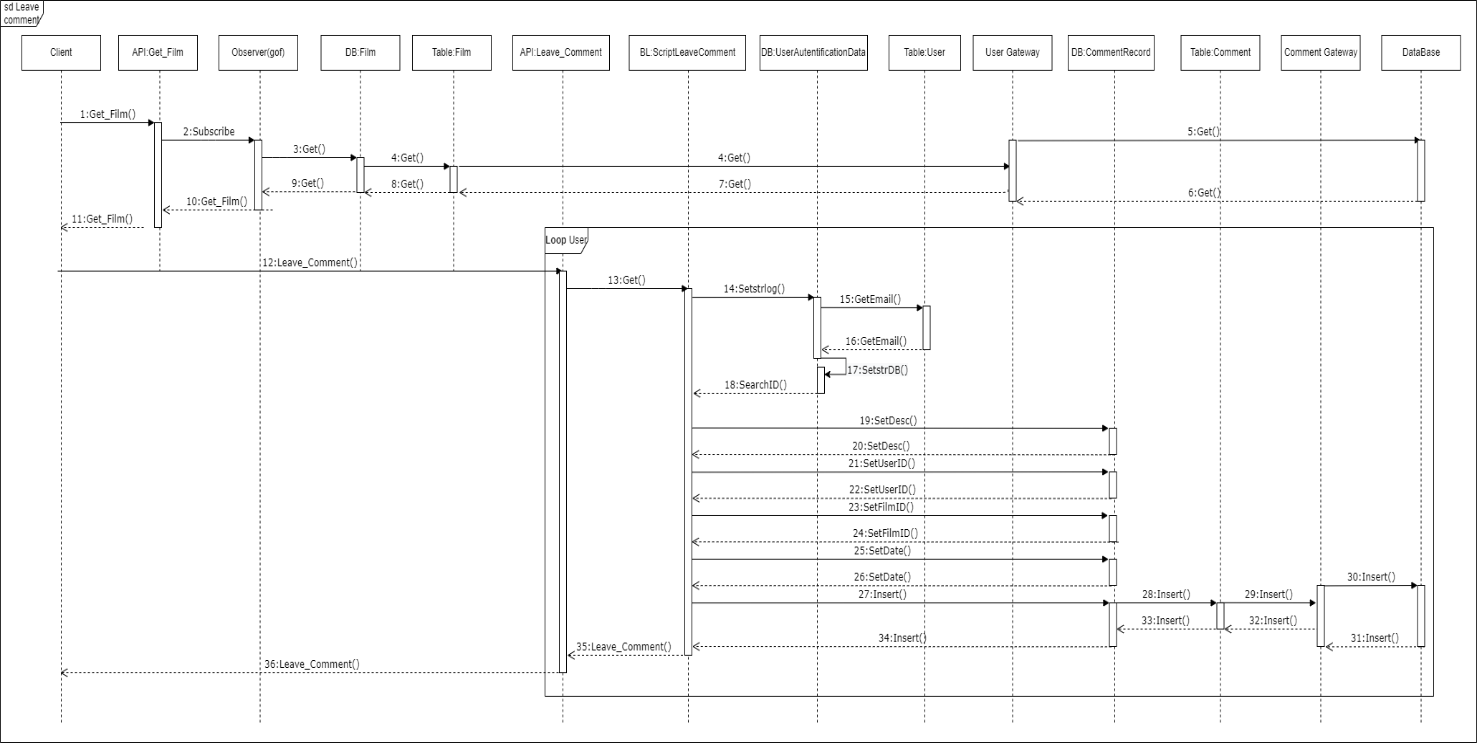


Рисунок 2. Диаграмма последовательностей прецедента «Оставить комментарий»

**Литература:**

1. Конспект лекций.

2. Фаулер М. Архитектура корпоративных приложений. - М.:Изд.дом Вильямс. - 2008 г.

3. Martin Fowler . Patterns of Enterprise Application Architecture.

https://martinfowler.com/books/eaa.html, https://martinfowler.com/eaaCatalog/

1.Технологии разработки программного обеспечения: Учебник/ С. Орлов. — СПб.: Питер, 2002. — 464 с.: ил. ISBN 5-94723-145-Х

2.Виноградова М.В., Белоусова В.И. Унифицированный процесс разработки программного обеспечения: учебное пособие / Виноградова М.В., Белоусова В.И. – М.: МГТУ им.Н.Э. Баумана. – 2015 г. – 82 с.

3.Орлов С.А., Цилькер Б.Я. Технологии разработки программного обеспечения. - СПб.: Питер. - 2012 г.