**Постановка задачи**

**1. Организационно-экономическая сущность задачи**

Наименование задачи: Сайт для репетиторства

Цель разработки: Разработать на примере аналогов функциональный сайт по репетиторству в Гродно

Назначение: Обучение

Периодичность использования: Круглосуточно, в любое время года

Источники и способы получения данных: grodno.profi-bel.by и edme.by

Обзор существующих аналогичных ПП: Рассмотрим сайт grodno.profi-bel.by, на нем мы можем увидеть удобный функционал, отзывы, рейтинг, что позволяет сделать более подходящий выбор. Есть поле регистрации и входа, контактные данные администрации сайта.

**2. Функциональные требования**

Описание перечня функций и задач, которые должен выполнять будущий ПП: Описание перечня функций и задач, которые должен выполнять

будущий ПП:

Гость:

-Регистрация

-Просмотр страниц сайта

-Поиск репетитора

-Использование фильтров

-Просмотр списка репетиторов

-Просмотр рейтинга репетиторов

-Просмотр отзывов

Пользователь:

Пользователь имеет все функции гостя, плюс:

-Просмотр контактных данных репетиторов

-Составление резюме

-Написание отзывов

Администратор:

Администратор имеет все функции пользователя и гостя

-Модерация резюме и отзывов

-Управление данными пользователя

**3. Описание процессов с входной, выходной и условно-постоянной информацией:**

Гость:

При регистрации в окне авторизации, будет создан новый аккаунт учащегося с настроенным профилем в соответствии с желанием гостя. Входная информация – данные нового пользователя (Имя, пароль, почта), выходная информация – сообщение “Вы зарегистрированы”, условно-постоянная информация – поля регистрации.

При входе в аккаунт входной информацией будет почта и пароль аккаунта, выходной – отсутствует, условно-постоянная информация – поля авторизации.

При просмотре информации сайта, входная информация – данные сайта, выходная информация – отсутствует, условно-постоянная информация – данные сайта.

Просмотр контактной информации, на главной странице в самом низу есть почта для связи с службой поддержки, входная информация – имя, сообщение, почта, выходная информация – отсутствует, условно-постоянная информация – почта службы поддержки.

Пользователь:

Составление резюме, нужно в профиле создать свое резюме, входная информация – данные, выходная информация – надпись “Резюме готово”, условно-постоянная информация – анкета для заполнения.

Написание отзывов, возможность оставить отзыв у конкретного репетитора, входная информация – отзыв, выходная информация – отсутствует, условно-постоянная информация – поле для заполнения, данные репетитора.

Администратор:

Модерация резюме и отзывов, переход на страницу “Резюме” и изменение резюме и отзывов у репетитора, входная информация – отзывы, резюме, выходная информация – измененные отзывы, условно-постоянная информация – отсутствует.

**4. Эксплуатационные требования**

Требования к применению: помощь в нахождении репетитора в определенной области

Требования к реализации: Для реализации статических страниц и шаблонов должны использоваться языки HTML и CSS. Для реализации интерактивных элементов клиентской части должны использоваться языки JavaScript.

Требования к надежности: Защита пользовательской информации, исправное состояние сайта и его устойчивость к нагрузке

Требования к интерфейсу: Возможность посмотреть интересующую пользователя информацию должна быть доступна с первой страницы. На главной странице будет доступна информация о сайте. Сайт должен адаптироваться под компьютер.

Требования к хостингу:

* Объем дискового пространство 8Гб,
* Эффективная защита от спама и взлома
* Базы данных хранятся на SSD-дисках.
* Максимальный размер базы данных - 2 Гб
* Все данные проходят через процедуру ежедневного резервного копирования. Копии хранятся в течение 20 дней.

**5 Выбор инструментов разработки**

Для разработки были выбраны языки программирования, которые наилучшим образом соответствует потребностям и возможностям проекта. Для такой задачи лучше всего подойдут языки программирования HTML и CSS.

Вот несколько причин почему были выбраны эти языки программирования:  
Простота и Надежность: HTML и CSS - это простые и легко понимаемые языки, что делает их идеальными для начинающих разработчиков. Вы сможете быстро освоить основы и приступить к созданию своего сайта.

Универсальность: HTML предоставляет структуру для ваших веб-страниц, определяя элементы и их взаимосвязи. CSS, в свою очередь, позволяет стилизовать и оформлять эти элементы. Оба языка плотно интегрированы, что обеспечивает гармоничное взаимодействие между структурой и дизайном.

Поддержка веб-стандартов: HTML и CSS являются стандартами для веб-разработки. Это обеспечивает совместимость вашего сайта с различными браузерами и устройствами, что важно для достижения оптимального пользовательского опыта.

Открытый исходный код: HTML и CSS являются открытыми исходными кодами, что означает, что вы можете свободно использовать их без лицензионных ограничений. Это также поддерживает принципы открытого сообщества, где разработчики активно обмениваются опытом и ресурсами.

Богатство Ресурсов и Документации: Для HTML и CSS существует обширная документация и большое количество обучающих ресурсов. Это облегчает процесс обучения и решения возможных проблем.

**6 UX-проектирование пользовательского интерфейса**

Ссылка на проект с дизайном интерфейса:

https://www.figma.com/file/w6wFIT0KNnLVWNCHabOvH8/LingoLearn?type=design&node-id=0%3A1&mode=design&t=oOrUDxJ0cdWjxAdG-1

**7 UI-проектирование пользовательского интерфейса**

Ссылка на проект с дизайном интерфейса:

https://www.figma.com/file/cN0MGWdmGXLoo5gUW389ZC/LingoLearn?type=design&node-id=0%3A1&mode=design&t=x1qR4ATlVnhxmkD2-1

**9 Реализация**

Web-ресурс уже находится на бесплатном хостинге, и его можно посмотреть

по ссылке: lingolearn-practice.netlify.app

**10 Тестирование**

Таблица 10.1 – Отчёт результатах тестирования

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Тест | Ожидаемый результат | Физический результат | Результат  тестирования |
| 1 | Проверка кнопки «Топ» | Переход на блок «Топ» на главной странице | Переход на блок «Топ» | Выполнено |
| 2 | Проверка кнопки «Кабинет» | Переход на экран «Авторизация» | Переход на экран «Авторизация» | Выполнено |
| 3 | Проверка полей ввода экрана «Авторизация» | Заполнение данных | Заполнение данных | Выполнено |
| 4 | Проверка кнопки «Регистрация» на экране «Авторизация» | Переход на экран «Регистрация» | Переход на экран «Регистрация» | Выполнено |
| 5 | Проверка полей ввода экрана «Регистрация» | Заполнение данных | Заполнение данных | Выполнено |
| 6 | Использование фильтров в поиске | Использование фильтров | Использование фильтров | Выполнено |

Продолжение таблицы 1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 7 | Переход на вкладки в подвале | Переход на блоки | Переход на блоки | Выполнено |

**11 Руководство пользователя**

При входе на сайт встречает главная страница (рисунок 1)

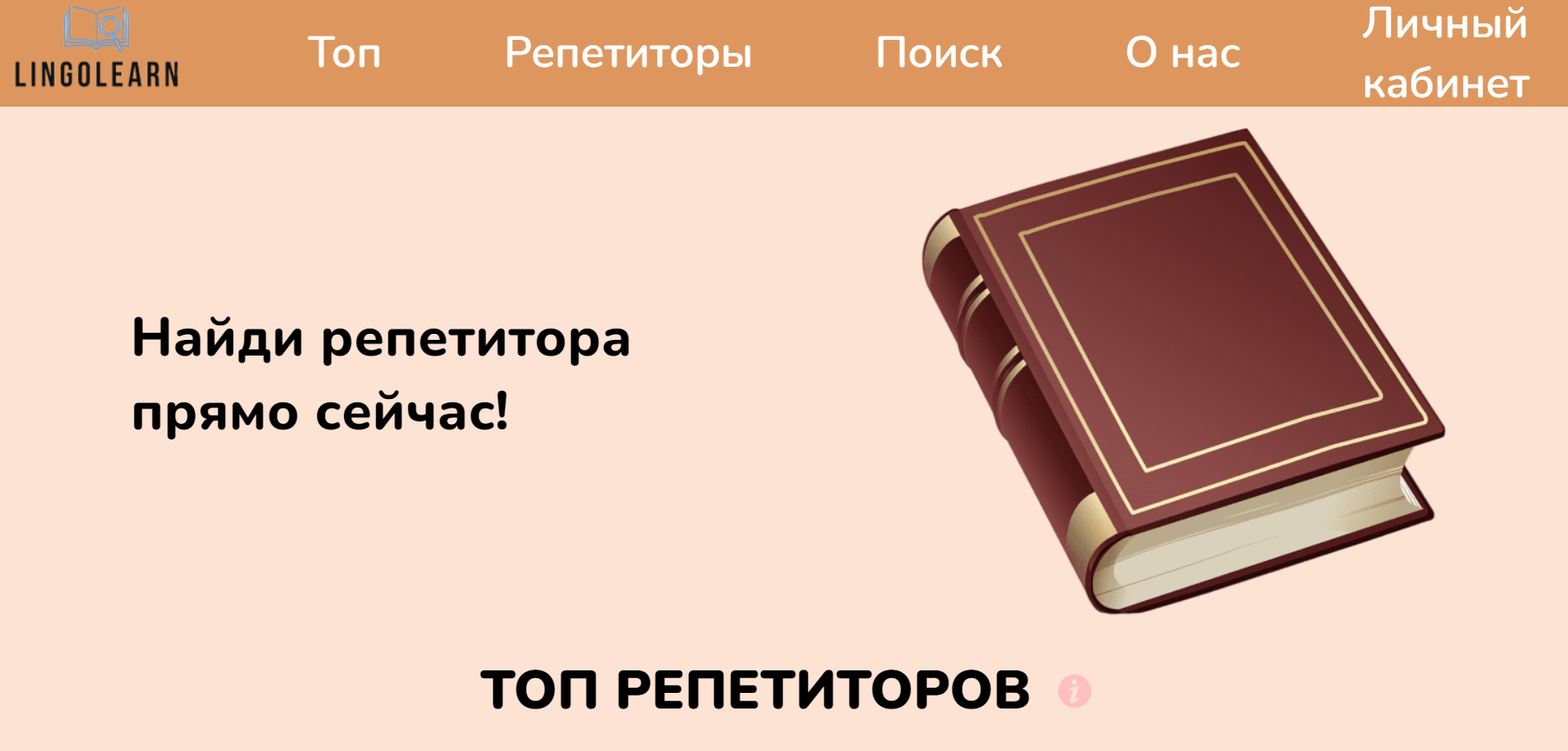


Рисунок 1 – главная страница

В хэдере у нас есть несколько пунктов, при нажатии которых нас переносят на нужный блок на главной странице (рисунок 2)

Рисунок 2 – вкладка «Репетиторы»

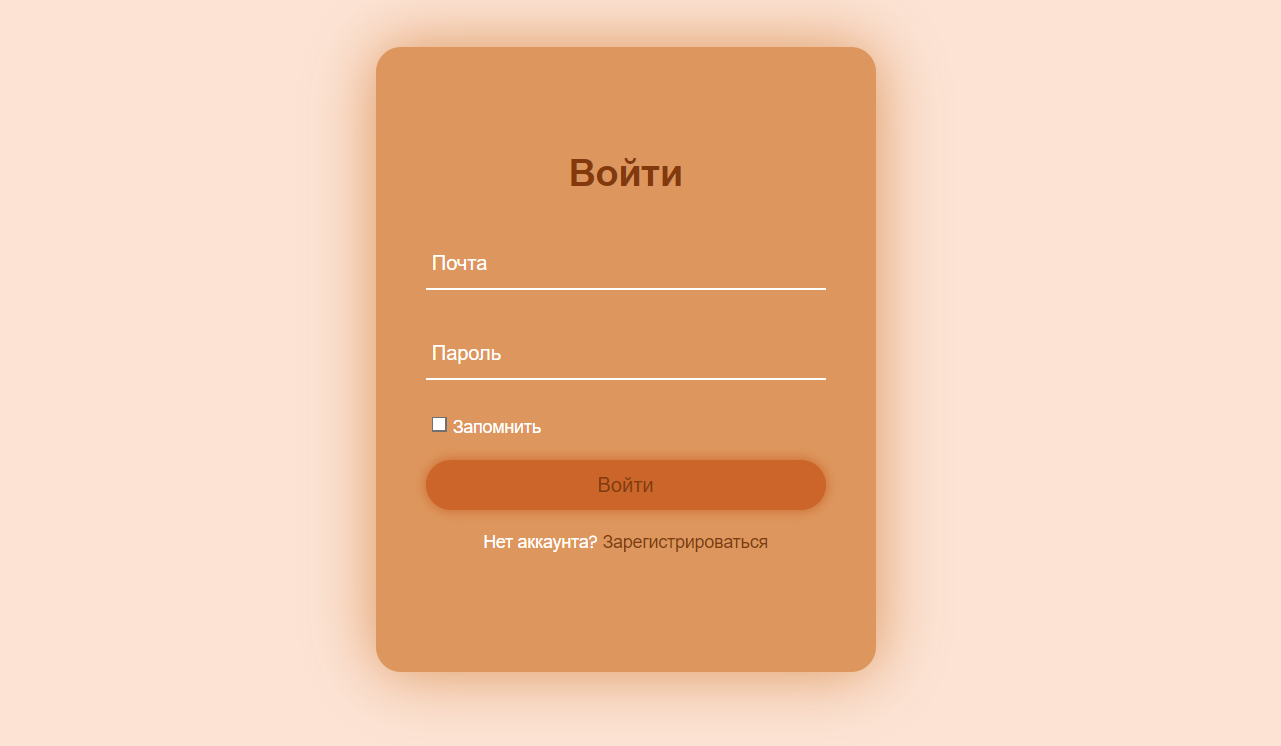
При нажатии на вкладку «Личный кабинет» нас переносит на вкладку авторизации (рисунок 3)

Рисунок 3 – вкладка авторизации

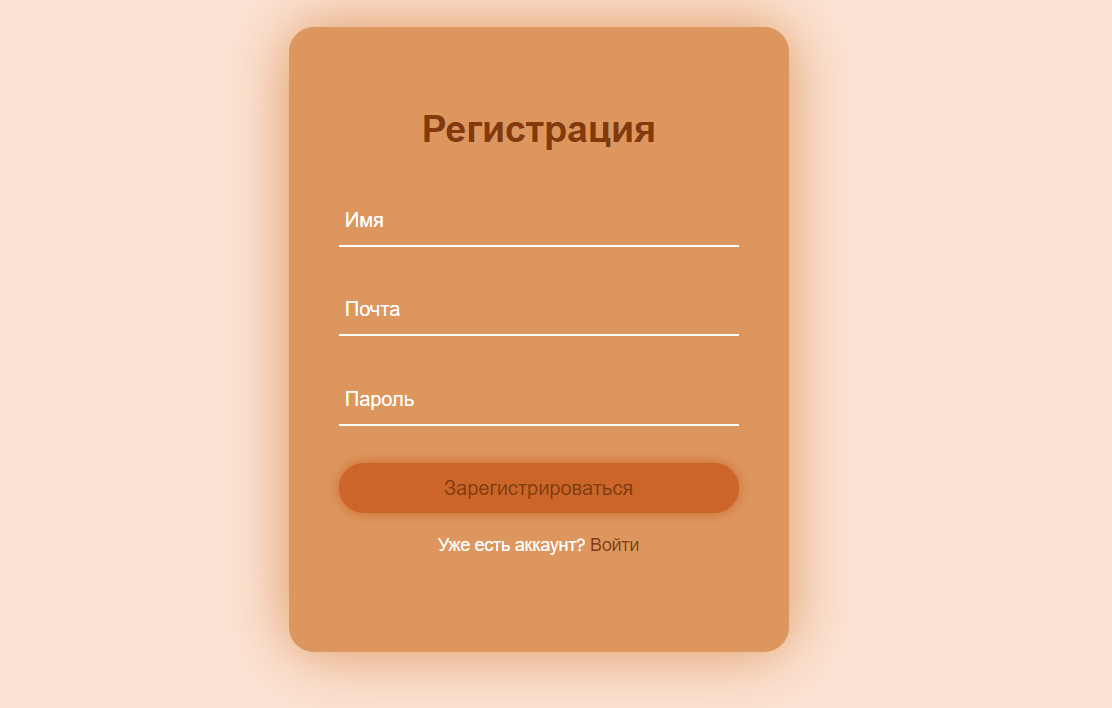
Если нету аккаунта, можно нажать на кнопку «Зарегистрироваться», которая переносит на вкладку регистрации (рисунок 4)

Рисунок 4 – вкладка регистрации

После авторизации нас переносит в кабинет (рисунок 5)



Рисунок 5 - кабинет

**11.2 Руководство программиста**

**11.2.1 Организация данных**

index.html – html разметка главной страницы;

style.css – css стиль главной страницы;

media\_style.css – css стиль масштабирования главной страницы;

authoriz.html – html разметка страницы авторизации;

authoriz.css – css разметка страницы авторизации;

registr.html – html разметка страницы регистрации;

registr.css – css разметка страницы регистрации;

cabiner.html – html разметка кабинета;

cabiner.css – css стиль кабинета;

cabinet\_meadia.css – css стиль масштабирования кабинета.

**Приложение А**

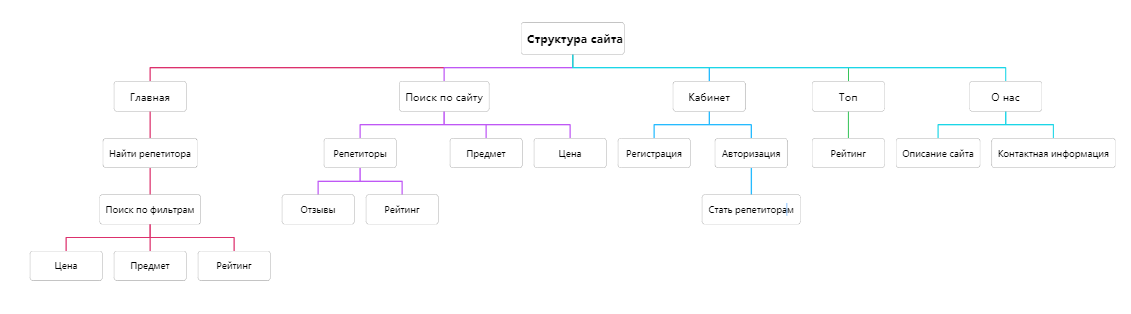


Рисунок А.1 – Структура сайта

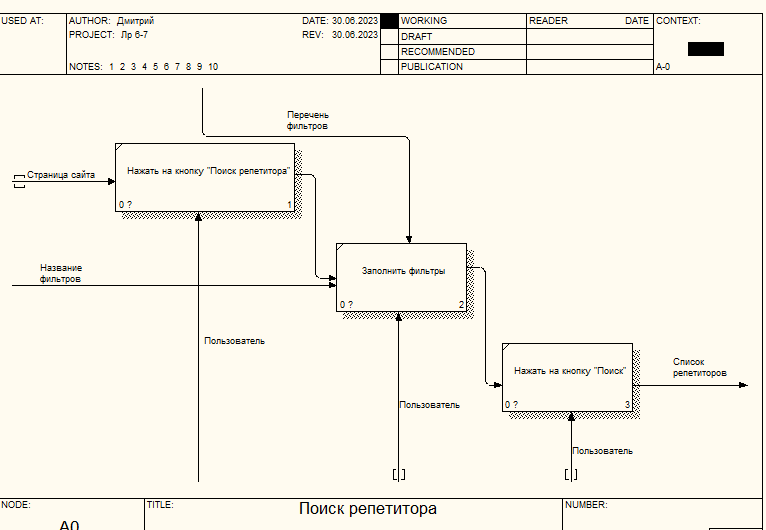
****

Рисунок А.2 – Функциональная модель

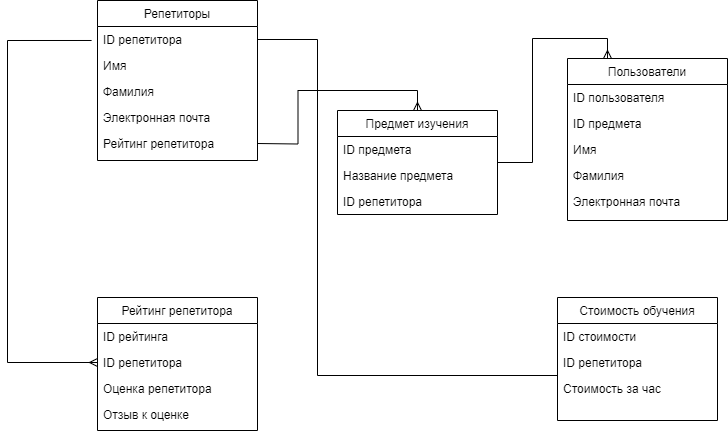


Рисунок А.3 – Модель данных

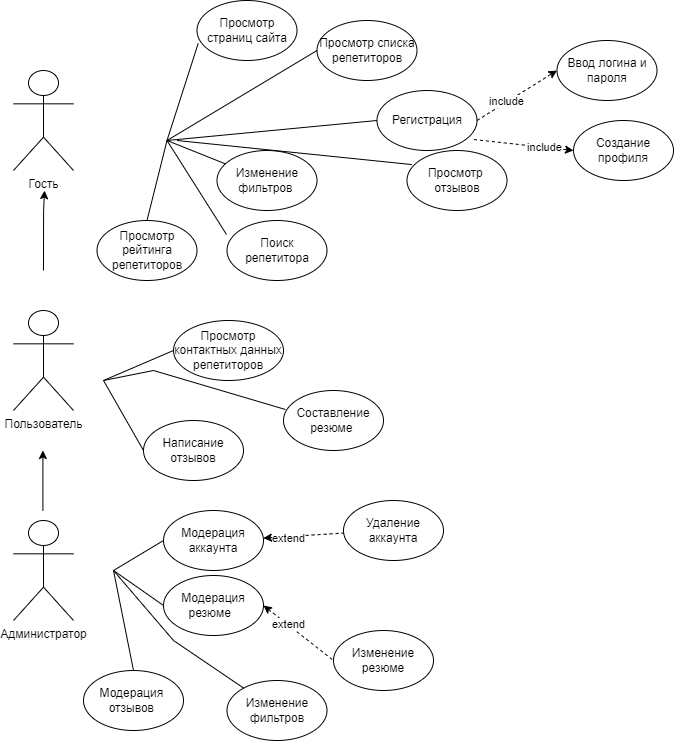


Рисунок А.4 – Диаграмма видов использования

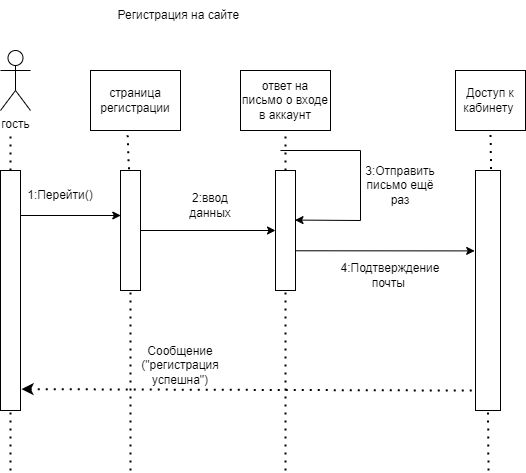


Рисунок А.5 – Диаграмма последовательности

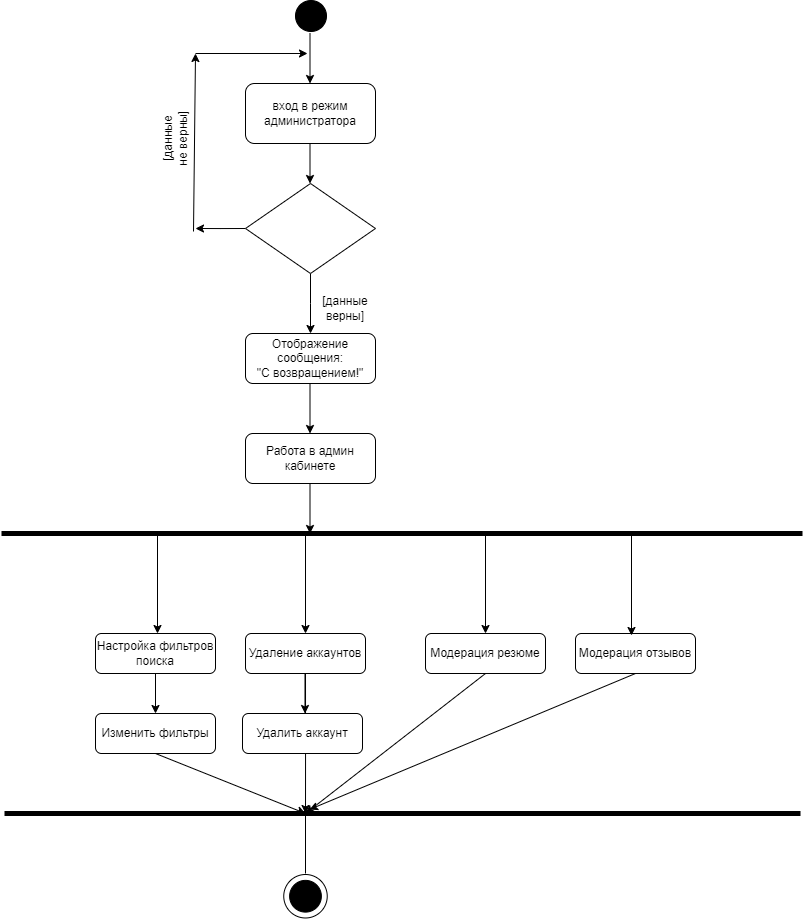


Рисунок А.6 – Диаграмма деятельности

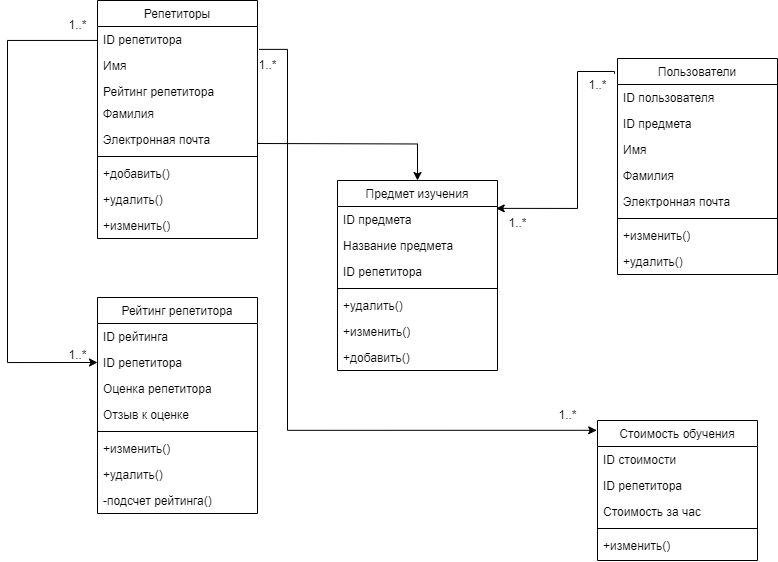


Рисунок А.7 – Диаграмма классов

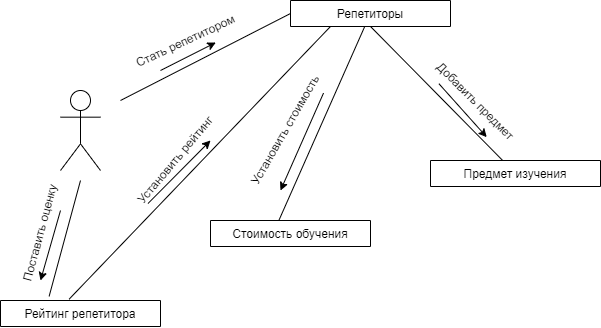


Рисунок А.8 – Диаграмма объектов