

МИНОБРНАУКИ РОССИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»  
(ФГБОУ ВО «ВГУ»)

Факультет Компьютерных наук  
Кафедра программирования и информационных технологий

Техническое задание  
на разработку веб-приложения  
«<Платформа для создания и распространения интерактивных обучающих  
курсов для детей и взрослых: Unity >»

Исполнители

\_\_\_\_\_ Д.А. Зазвонных

\_\_\_\_\_ Р.А. Легкунов

\_\_\_\_\_ А.Е. Ускова

Заказчик

\_\_\_\_\_ В.С. Тарасов

Воронеж 2024

## СОДЕРЖАНИЕ

ИСПОЛЬЗУЕМЫЕ ТЕРМИНЫ .....	4
1 Общие сведения.....	6
1.1 Полное наименование системы и ее условное обозначение .....	6
1.2 Наименование исполнителя и заказчика приложения .....	6
1.2.1 Наименование заказчика .....	6
1.2.2 Наименование исполнителя .....	6
1.3 Перечень документов, на основании которых создается приложение .	7
1.4 Состав и содержание работ по созданию системы .....	7
1.5 Порядок оформления и предъявления заказчику результатов работ по созданию приложения .....	8
2 Цели и назначение создания автоматизированной системы.....	9
2.1 Назначение АС .....	9
2.2 Цели создания сайта .....	9
3 Требования к приложению и программному обеспечению .....	10
3.1 Требования к архитектуре .....	10
3.2 Требования к технологиям .....	10
3.3 Требования к функциям, выполняемые системой .....	11
3.4 Требования к надежности.....	12
3.5 Общие требования к оформлению и верстке страниц .....	12
4 Структура сайта .....	13
5 Языковые версии сайта .....	15
6 Описание страниц сайта.....	16
6.1 Описание динамических страниц .....	16
6.1.1 Страница входа .....	16
6.1.2 Страница регистрации .....	16
6.1.3 Страница личного кабинета .....	18
6.1.4 Главная страница .....	19

6.1.5	Поисковая страница .....	20
7	Дизайн приложения .....	22
ПРИЛОЖЕНИЕ .....		23

## ИСПОЛЬЗУЕМЫЕ ТЕРМИНЫ

**Проект, сайт** – веб-платформа, над которой ведется разработка.

**Frontend** – клиентская часть веб-приложения, задачей которой является представление данных, полученных от сервера, в удобном для пользователя виде на его устройстве.

**Сервер** (серверная часть) – компьютерная система, предназначенная для обработки запросов от клиентских устройств и предоставления им необходимых ресурсов для выполнения задач.

**Клиентская сторона** – устройство пользователя, использующее сервисы и ресурсы сервера для взаимодействия с веб-приложением.

**Backend** – программная и аппаратная части веб-платформы, отвечающие за логику работы приложения и обработку данных.

**GitHub** – платформа для размещения IT-проектов, обеспечивающая возможности для их совместной разработки и управления версиями.

**Фреймворк** – инструментарий для упрощения процесса разработки программ, предоставляющий готовые компоненты и шаблоны для создания структуры проекта.

**Пользователь** – лицо, имеющее доступ к функционалу веб-приложения через авторизацию в системе.

**REST API** – методология создания веб-приложений, предусматривающая взаимодействие клиента и сервера посредством HTTP-запросов для обмена данными.

**Python** – язык программирования с широким спектром применения, поддерживающий динамическую типизацию и автоматическое управление памятью.

**Django** – фреймворк для разработки веб-приложений на Python, обеспечивающий инструменты для создания безопасных и масштабируемых веб-сайтов.

**Система управления базами данных (СУБД)** – комплекс программного обеспечения, предназначенный для создания, хранения и управления данными в базах данных.

**PostgreSQL** – открытая реляционная система управления базами данных, поддерживающая транзакции и сложные запросы.

**JavaScript** – язык программирования для создания интерактивных элементов на веб-страницах, используемый в разработке пользовательских интерфейсов.

**CSS** – язык стилей, применяемый для задания внешнего вида документов, написанных на языке разметки.

**JavaScript-библиотека** – набор предварительно написанных кодов на JavaScript, предназначенных для упрощения разработки веб-приложений.

**HTML** – язык разметки, используемый для структурирования и представления содержимого на веб-страницах.

## **1 Общие сведения**

### **1.1 Полное наименование системы и ее условное обозначение**

Полное наименование сайта: «Платформа для создания и распространения интерактивных обучающих курсов для детей и взрослых Unity».

Условное обозначение сайта: «Unity».

### **1.2 Наименование исполнителя и заказчика приложения**

#### **1.2.1 Наименование заказчика**

Заказчик: Старший преподаватель Тарасов Вячеслав Сергеевич.  
Воронежский Государственный Университет, Факультет компьютерных наук, кафедра Программирования и Информационных Технологий.

#### **1.2.2 Наименование исполнителя**

Разработчик: ВГУ ФКН 3 курс 3 группа 1 команда

Состав команды разработчика:

- Зазвонных Денис Александрович – Frontend Разработчик, Системный аналитик, Тестировщик;
- Легкунов Роман Александрович – Teamlead, Backend разработчик, Project Manager, Тестировщик, Системный аналитик, Архитектор;
- Ускова Анна Евгеньевна – Бизнес Аналитик, Дизайнер, Разработчик БД, Тестировщик, Владелец продукта.

### **1.3 Перечень документов, на основании которых создается приложение**

Данный сайт будет создаваться на основании следующих документов:

- Федеральный закон "Об информации, информационных технологиях и о защите информации" от 27.07.2006 N 149–ФЗ;
- Федеральный закон "О персональных данных" от 27.07.2006 N 152–ФЗ.

### **1.4 Состав и содержание работ по созданию системы**

Состав и содержание работ по созданию сайта включают в себя следующие этапы:

- Сбор необходимой информации, постановка целей, задач системы, которые в будущем должны быть реализованы 20.02.24 – 01.03.24;
- Анализ предметной области, анализ системы и построение структуры требований, ведущих к решению поставленных задач и целей 01.03.24 – 13.03.24;
- Построение модели программы, описание спецификаций данных, определение связей между сущностями, разработка модели БД 01.03.24 – 13.03.24;
- Построение рабочего проекта, состоящего из написания кода, отладки и корректировки кода программы 13.04.24 – 15.05.24;
- Проведение тестирования и доработка информационного программного обеспечения 01.05.24 – 01.06.24.

## **1.5 Порядок оформления и предъявления заказчику результатов работ по созданию приложения**

Предварительные отчёты по работе будет проводиться во время рубежных аттестаций:

- 1 аттестация (середина-конец марта 2024) – создан репозиторий проекта на GitHub, распределены задачи проекта в каналах связи (Telegram, Discord, в перспективе – таск-менеджере YouTrack), налажено проведение онлайн и оффлайн собраний, создан проект Miro с общей логикой системы, предоставлены готовое техническое задание, сопроводительное письмо и презентация проекта;
- 2 аттестация (середина-конец апреля 2024) – написана основополагающая часть кода приложения, реализована БД и ее взаимодействие с сервером, проведена отладка и доработка кода, проведено предварительное тестирование по работе системы;
- 3 аттестация (середина-конец мая 2024) – разработан курсовой проект, выполнены завершающие работы по доработке приложения, предоставлена готовая система.

Результаты работы должны быть представлены в электронном виде в формате pdf и размещены на GitHub.



## **2 Цели и назначение создания автоматизированной системы**

### **2.1 Назначение АС**

Сайт позволяет решать следующие задачи:

- Искать курсы по категориям и ключевым словам;
- Просматривать контент курсов и уроков;
- Взаимодействовать с сообществом посредством комментариев;
- Модерировать контент;
- Редактировать данные аккаунта.

### **2.2 Цели создания сайта**

К целям создания сайта относятся:

- Предоставление качественного и доступного образовательного контента для детей и взрослых, способствующего развитию новых навыков и знаний;
- Создание условий для совместного проведения времени родителей и детей, что способствует укреплению семейных отношений и обеспечивает общий образовательный процесс;
- Внедрение интерактивных элементов (комментариев, оценивания уроков) для повышения мотивации и удержания интереса к обучению среди пользователей всех возрастов;
- Простой и лаконичный дизайн веб-приложения, который будет понятен не только родителям, но и их детям.

## **3 Требования к приложению и программному обеспечению**

### **3.1 Требования к архитектуре**

В архитектуре приложения клиент взаимодействует с серверной частью через веб-интерфейс. Обмен данными между клиентом и сервером осуществляется через REST API, который сервер использует для предоставления необходимой информации клиенту. Сервер состоит из нескольких ключевых компонентов: сервис для регистрации и входа в систему, API для обработки запросов, компонент для взаимодействия с базой данных и модуль для обработки данных. Для сохранения информации о пользователях в веб-приложении предусмотрена база данных.

### **3.2 Требования к технологиям**

Для реализации серверной части приложения будут использоваться следующие технологии:

- Фреймворк Django;
- Язык программирования Python;
- СУБД PostgreSQL.

Для реализации клиентской части приложения будут использоваться следующие технологии:

- Фреймворк Vue.js;
- Язык программирования JavaScript;
- Язык стилей CSS;
- Язык гипертекстовой разметки HTML.

### **3.3 Требования к функциям, выполняемые системой**

Приложение должно содержать функциональные требования для следующих групп пользователей:

- Неавторизованные пользователи (гости);
- Авторизованные пользователи;
- Модераторы.

#### **Неавторизованные пользователи**

В возможности неавторизованных пользователей входят:

- Поиск курсов;
- Просмотр курсов и уроков.

#### **Авторизированные пользователи**

В возможности авторизованных пользователей входят:

- Поиск курсов;
- Просмотр курсов и уроков;
- Возможность оставлять комментарии под уроками;
- Возможность оценивания уроков.

#### **Модераторы**

В возможности модераторов входят:

- Поиск курсов;
- Просмотр курсов и уроков;
- Возможность оставлять комментарии под уроками;

- Возможность оценивания уроков;
- Возможность удалять курсы, уроки и комментарии.

### **3.4 Требования к надежности**

Приложение должно содержать следующие требования к надежности:

- Защита от SQL-инъекций;
- Масштабируемость;
- Гибкость.

### **3.5 Общие требования к оформлению и верстке страниц**

Приложение должно содержать требования к оформлению и верстке страниц:

- Единообразие стиля: дизайн всех страниц приложения должен быть выполнен в едином стиле, что включает цветовую схему, типографику, размеры и стили кнопок, поля ввода и другие элементы интерфейса;
- Соответствие техническому заданию: все элементы интерфейса и визуальное оформление должны соответствовать требованиям, изложенным в техническом задании;
- Видимость элементов управления: элементы управления (кнопки, ссылки, поля ввода) должны быть легко заметны и понятны пользователю на фоне страницы;
- Кроссбраузерность: дизайн должен быть адаптивным, корректно отображаться и функционировать в различных браузерах (Microsoft Edge, Google Chrome, Yandex Browser, Safari, Mozilla Firefox);

- Логотип: разработка и внедрение уникального логотипа «Unity», который будет использоваться на всех страницах приложения для укрепления бренда и узнаваемости платформы;
- Интуитивно понятная навигация: создание логической и интуитивно понятной структуры навигации, позволяющей пользователям легко перемещаться по сайту без дополнительных пояснений;
- Адаптация под целевую аудиторию: дизайн должен быть адаптирован под целевую аудиторию веб-приложения, включая детей и взрослых, с использованием соответствующих цветов, шрифтов и элементов;
- Контрастность и размер шрифтов: обеспечение достаточной контрастности текста и фона, а также адекватных размеров шрифтов для обеспечения лучшего восприятия информации.

Помимо этих требований, будет проводиться тестирование дизайна с реальными пользователями для выявления и исправления проблем удобства использования и интерфейса.

#### **4 Структура сайта**

Приложение должно содержать следующие страницы:

- Приветственная страница;
- Страница «Регистрация»;
- Страница «Авторизация»;
- Страница «Восстановление доступа к аккаунту»;
- Главная (поисковая) страница;
- Страница с результатами поиска;

- Страница курса;
- Страница урока;
- Страница «Личный кабинет»;
- Страница «Настройки аккаунта»;
- Страница «Создание курса»;
- Страница «Редактирование курса»;
- Страница «Создание урока»;
- Страница «Редактирование урока».

## **5 Языковые версии сайта**

Приложение должно быть предоставлено для взаимодействия с пользователем на русском языке.

## **6 Описание страниц сайта**

### **6.1 Описание динамических страниц**

#### **6.1.1 Страница входа**

Страница входа доступна неавторизованным пользователям после нажатия кнопки «Войти» в основном меню главной или поисковой страницы сайта.

На странице находятся:

- Кнопка «Продолжить как гость», переносящая пользователя на главную поисковую страницу;
- Кнопка регистрации «Регистрация», переносящая пользователя на страницу регистрации;
- Кнопка входа в аккаунт «Войти», переносящая пользователя на страницу авторизации.

#### **6.1.2 Страница регистрации**

Страница регистрации доступна всем пользователям сайта после выбора кнопки регистрации на странице входа.

На странице находятся:

- Строка «Придумайте логин» с кликабельным полем (пустым) для ввода логина;
- Строка «Придумайте пароль» с кликабельным полем (пустым) для ввода пароля;
- Строка «Повторите пароль» с кликабельным полем для ввода (пустым) пароля;



- Строка «Введите контрольный вопрос» с кликабельным полем (пустым) для ввода вопроса;
- Строка «Введите контрольный ответ» с кликабельным полем (пустым) для ввода ответа;
- Строка «Введите название семьи» с кликабельным полем (пустым) для ввода названия семьи;
- Строка «Выберите роль члена семьи» с кликабельным выпадающим списком, состоящим из основных ролей: мама, папа, сын, дочь (Обязательно выбрать минимум 1 роль для продолжения регистрации).

Далее данные заполняются для выбранной роли члена семьи:

- Строка «Введите вашу Фамилию» с кликабельным полем (пустым) для ввода Фамилии;
- Строка «Введите ваше имя» с кликабельным полем (пустым) для ввода имени;
- Строка «Введите ваше отчество (если оно есть)» с кликабельным полем (пустым) для ввода Отчества;
- Строка «Введите вашу дату рождения» с кликабельным полем с шаблоном для ввода даты.

Ниже будет:

- Кнопка «Зарегистрироваться», которая завершает регистрацию нового пользователя (в случае верно введенного поля "Повторите пароль") и переносит его на главную поисковую страницу.

### 6.1.3 Страница личного кабинета

Страница личного кабинета доступна авторизованным пользователям в основном меню главной или поисковой страницы сайта.

На странице находятся:

- Строка «Профиль» сверху страницы, уведомляющая пользователя о том, где он находится;
- Кнопка настроек аккаунта «Настройки», переносящая пользователя на страницу настроек;
- Кнопка главной страницы «Главная», переносящая пользователя на главную страницу;
- Строка «Название семьи» с последующей строкой с названием семьи, указанном в настройках;
- Строка «Члены семьи» с информацией о членах семьи под ней в виде информационных сводок (см. ниже);
- Строка с ролью члена семьи, строка с ФИО члена семьи и строка с датой рождения члена семьи, образующие информационную сводку об одном члене семьи;
- Строка «Начатые курсы» информацией о начатых пользователем курсах под ней в виде информационных сводок (см. ниже);
- Строка «Название» с последующей строкой с названием курса;
- Строка «Автор» с последующей строкой с указанием автора курса;
- Строка «Выполнен на n%» с указанием информации, на сколько % пользователем пройден курс;
- Кнопки перехода на следующую и предыдущую страницы со списком начатых пользователем курсов, переносящая

пользователя на следующую и предыдущую страницы со списком начатых пользователем курсов;

- Строка «Созданные курсы» информацией о созданных пользователем курсах под ней в виде информационных сводок (см. ниже);
- Строка «Название» с последующей строкой с названием курса;
- Строка «Автор» с последующей строкой с указанием автора курса;
- Строка «Дата создания» с последующей строкой с датой создания курса;
- Кнопка редактирования курса «Редактировать курс», переносящая пользователя на страницу редактирования курса;
- Кнопки перехода на следующую и предыдущую страницы со списком созданных пользователем курсов, переносящая пользователя на следующую и предыдущую страницы со списком созданных пользователем курсов;
- Кнопка создания курса «Создать курс», переносящая пользователя на страницу создания нового курса.

#### **6.1.4 Главная страница**

Главная страница доступна всем пользователям сайта, на которую попадают все пользователи, когда заходят на сайт.

Страница содержит логотип сайта.

На странице находятся:

- Кликабельное поле поиска в верхней части страницы (после введения запроса в поле поиска и его отправки, переносит на поисковую страницу);

- Кликабельная кнопка «Войти» в верхней части страницы (для зарегистрированных пользователей меняется на кликабельную кнопку «Профиль»);
- Строка «Категории», располагающаяся над блоками категорий;
- Кликабельный блок категории курсов «Рисование»;
- Кликабельный блок категории курсов «Хендмэйд»;
- Кликабельный блок категории курсов «Развлечения»;
- Кликабельный блок категории курсов «Активный отдых».

После нажатия 1 или нескольких блоков категорий ниже блоков категорий появляется:

- Кликабельная кнопка «Начать просмотр курсов», которая переносит на поисковую страницу.

### **6.1.5 Поисковая страница**

Поисковая страница доступна всем пользователям, после нажатия кнопки «Начать просмотр курсов» или после введения и отправки запроса в поле поиска на главной странице.

Страница содержит логотип сайта.

На странице находятся:

- Кликабельное поле поиска в верхней части страницы;
- Кликабельная кнопка «Войти» в верхней части страницы (для зарегистрированных пользователей меняется на кликабельную кнопку «Профиль»);
- Табло с кликабельными блоками курсов 1 или нескольких категорий или по словам из названия курсов;

— Кликабельный блок курса состоит из названия курса, автора и оценки курса.

## **7 Дизайн приложения**

- Дизайн всех страниц приложения должен следовать единому стилевому решению;
- Разработка интерфейса должна соответствовать требованиям, изложенным в техническом задании;
- Элементы управления на сайте должны быть хорошо заметны на фоне страницы.

## ПРИЛОЖЕНИЕ

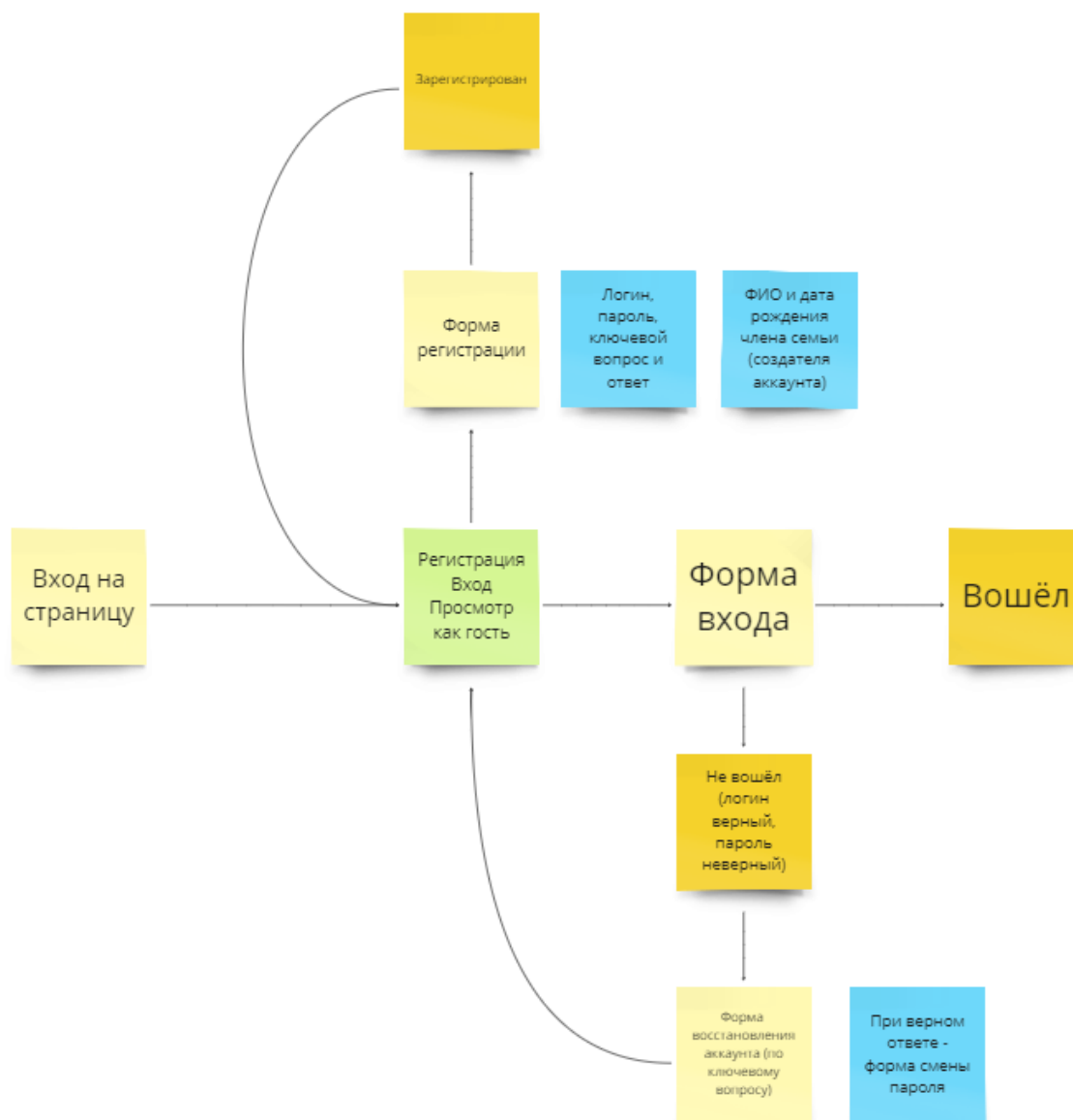


Рисунок 1 - Функциональная схема (вход и регистрация)

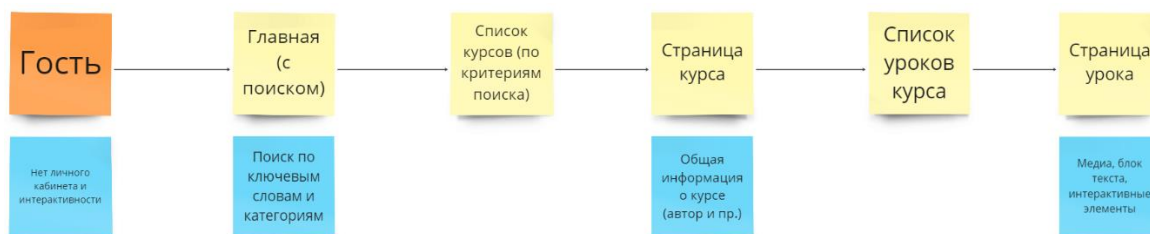


Рисунок 2 - Функциональная схема (гость)



Рисунок 3 - Функциональная схема (личный кабинет)

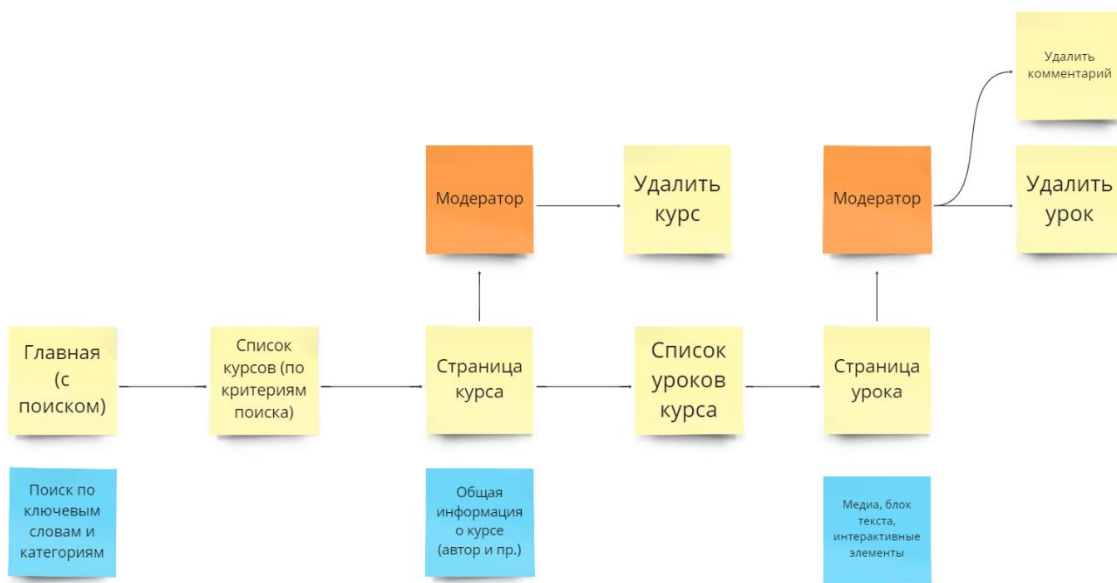


Рисунок 4 - Функциональная схема (поиск и просмотр курсов)



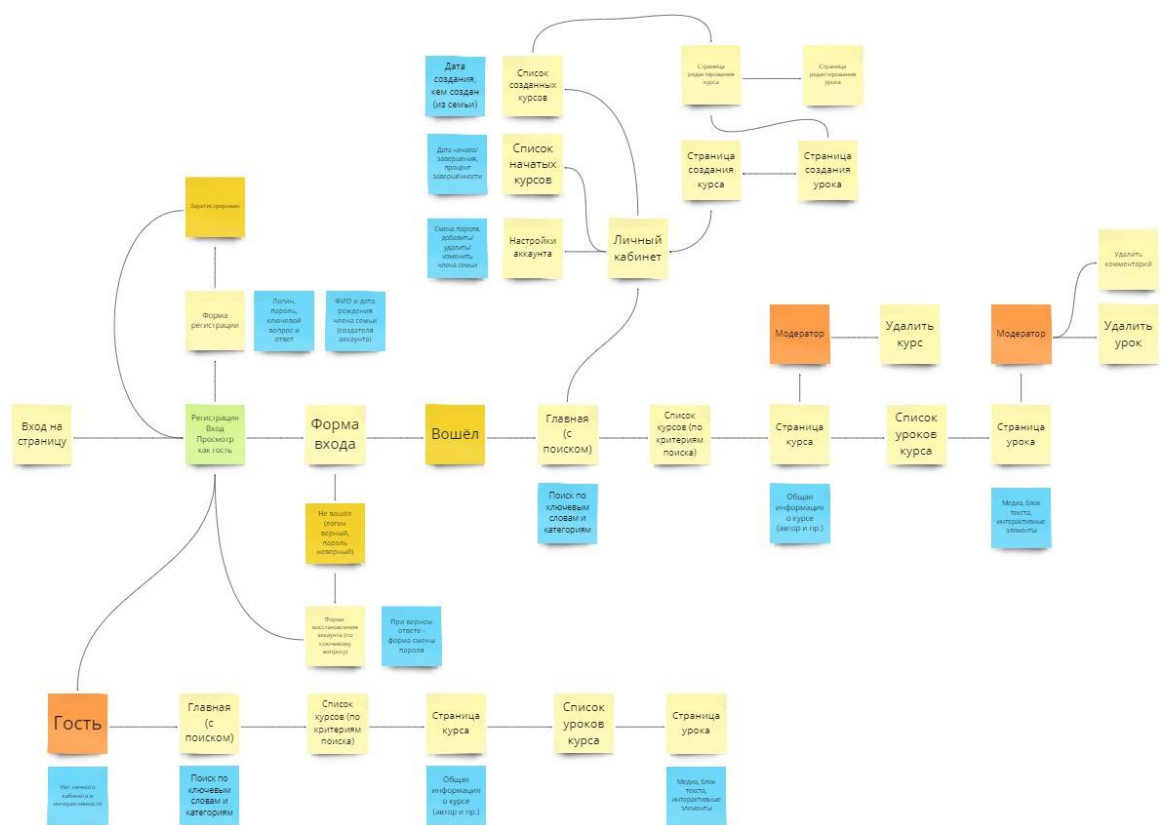


Рисунок 5 - Функциональная схема веб-приложения

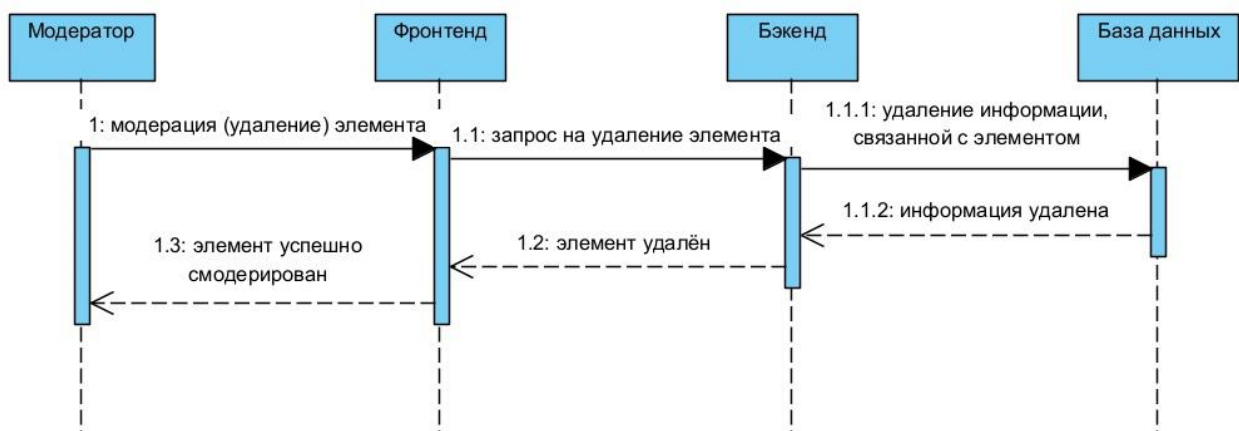


Рисунок 6 - Диаграмма последовательностей "Модератор удаляет элементы"

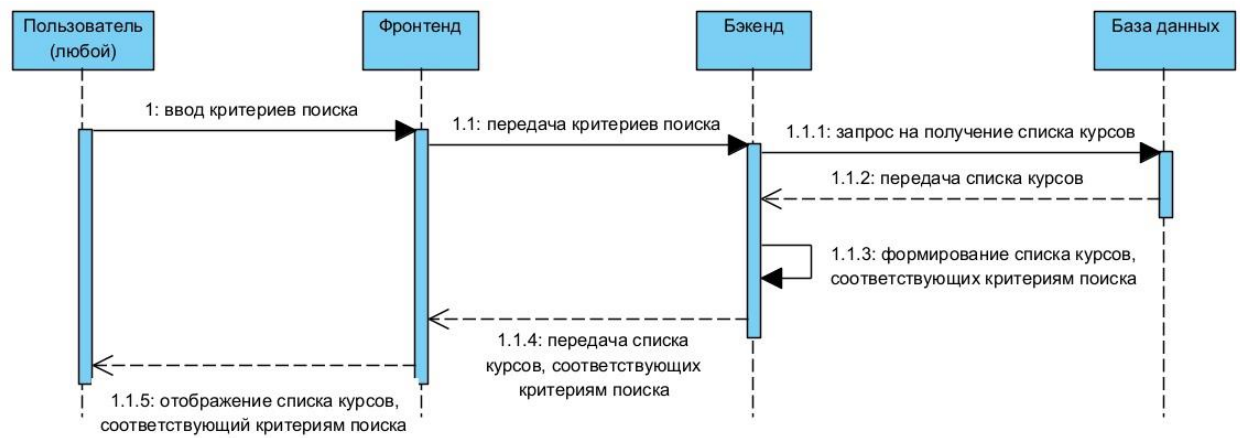


Рисунок 7 - Диаграмма последовательностей "Любой пользователь ищет курсы"

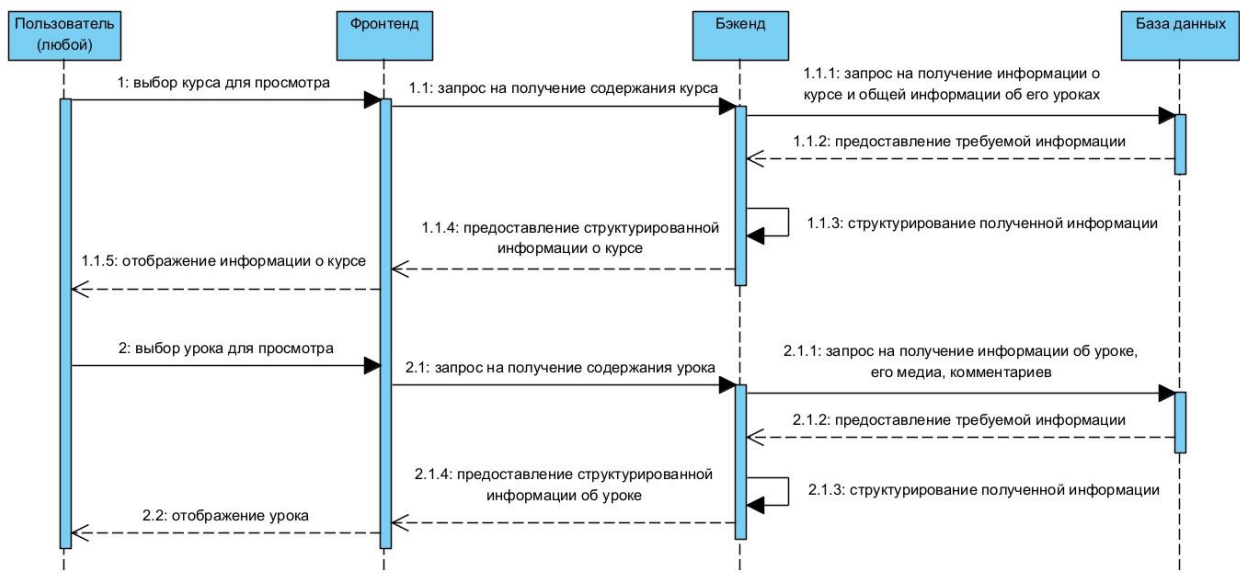


Рисунок 8 - Диаграмма последовательностей "Любой пользователь смотрит курсы и уроки"

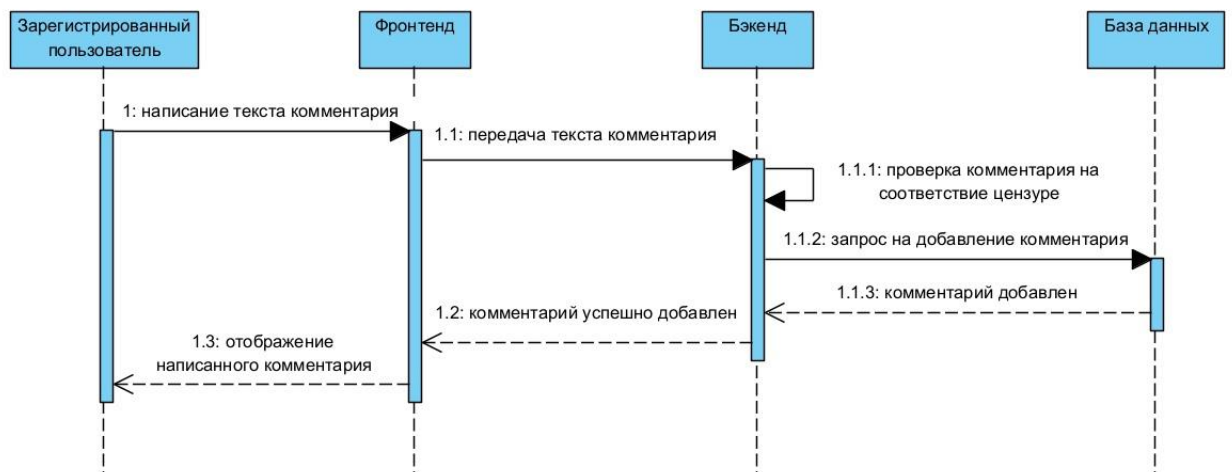


Рисунок 9 - Диаграмма последовательностей "Зарегистрированный пользователь пишет комментарии"

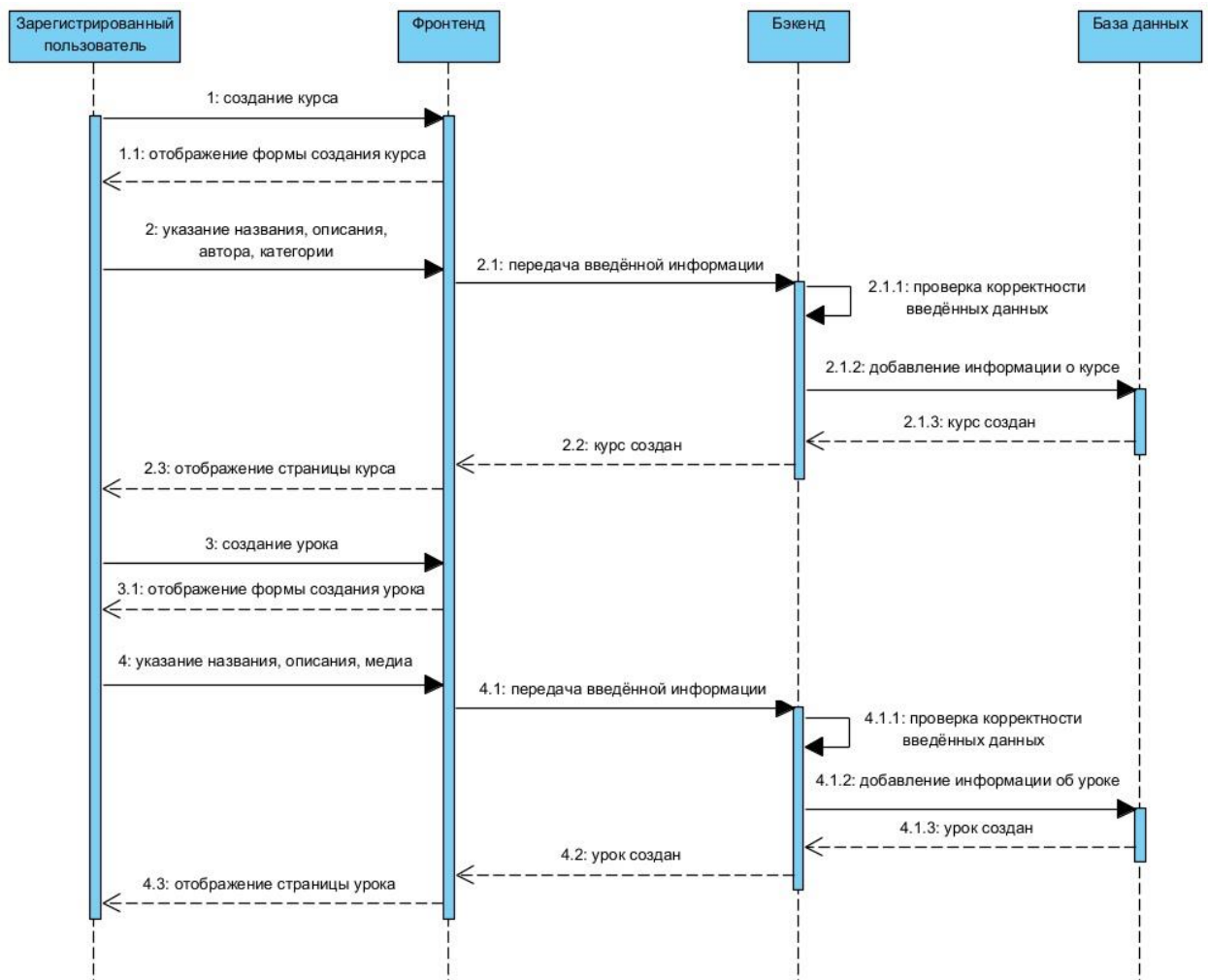


Рисунок 10 - Диаграмма последовательностей "Зарегистрированный пользователь создаёт курсы и уроки"

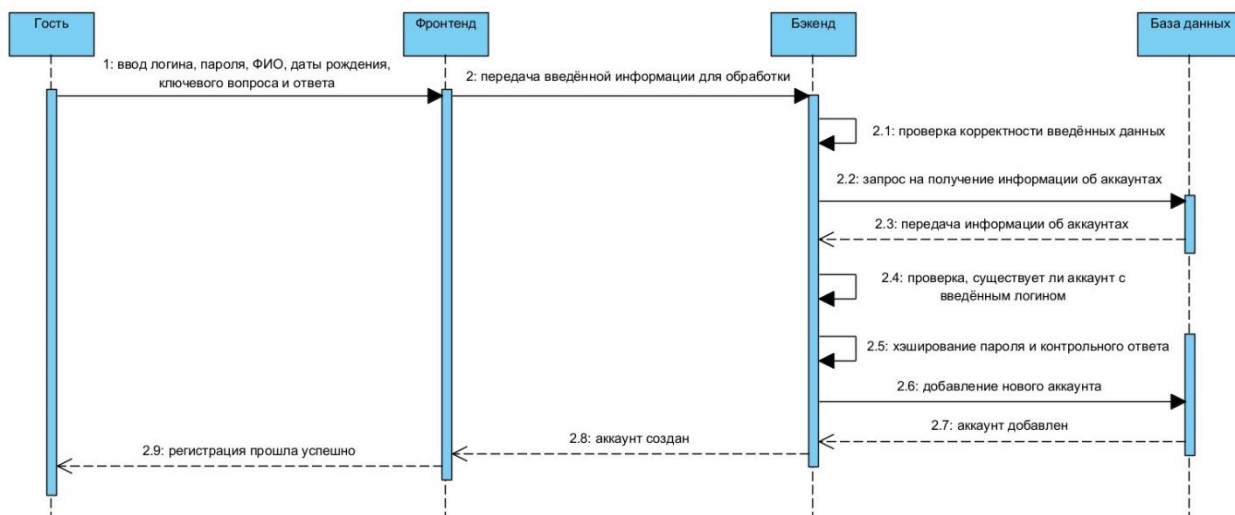


Рисунок 11 - Диаграмма последовательностей "Пользователь создаёт аккаунт"

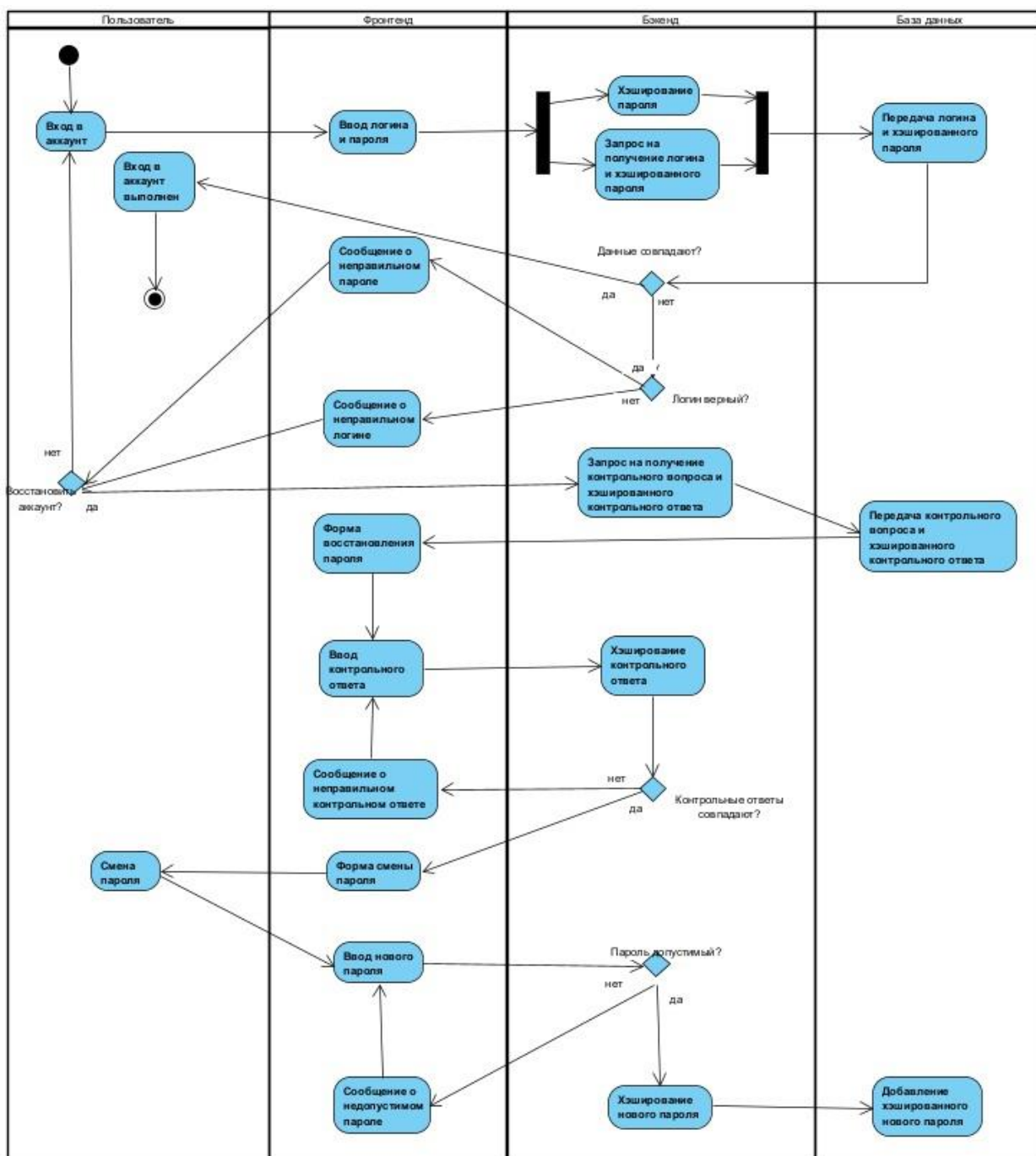


Рисунок 12 - Диаграмма активностей "Пользователь заходит в аккаунт"

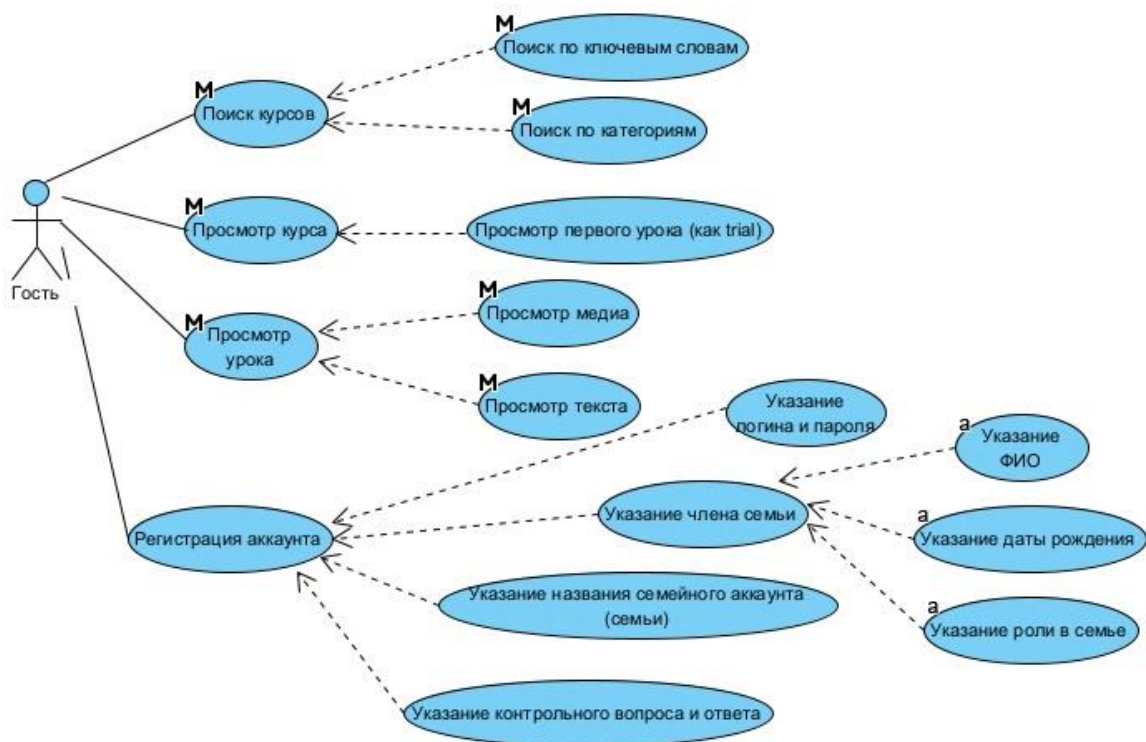


Рисунок 13 - Диаграмма взаимодействий USE CASE для актёра "Гость"

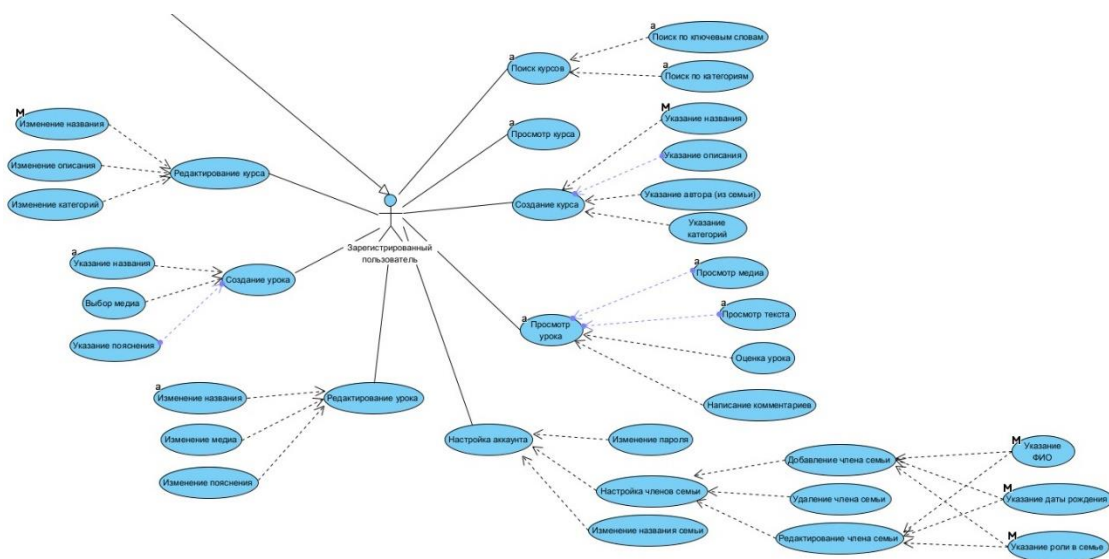


Рисунок 14 - Диаграмма взаимодействий USE CASE для актёра "Зарегистрированный пользователь"





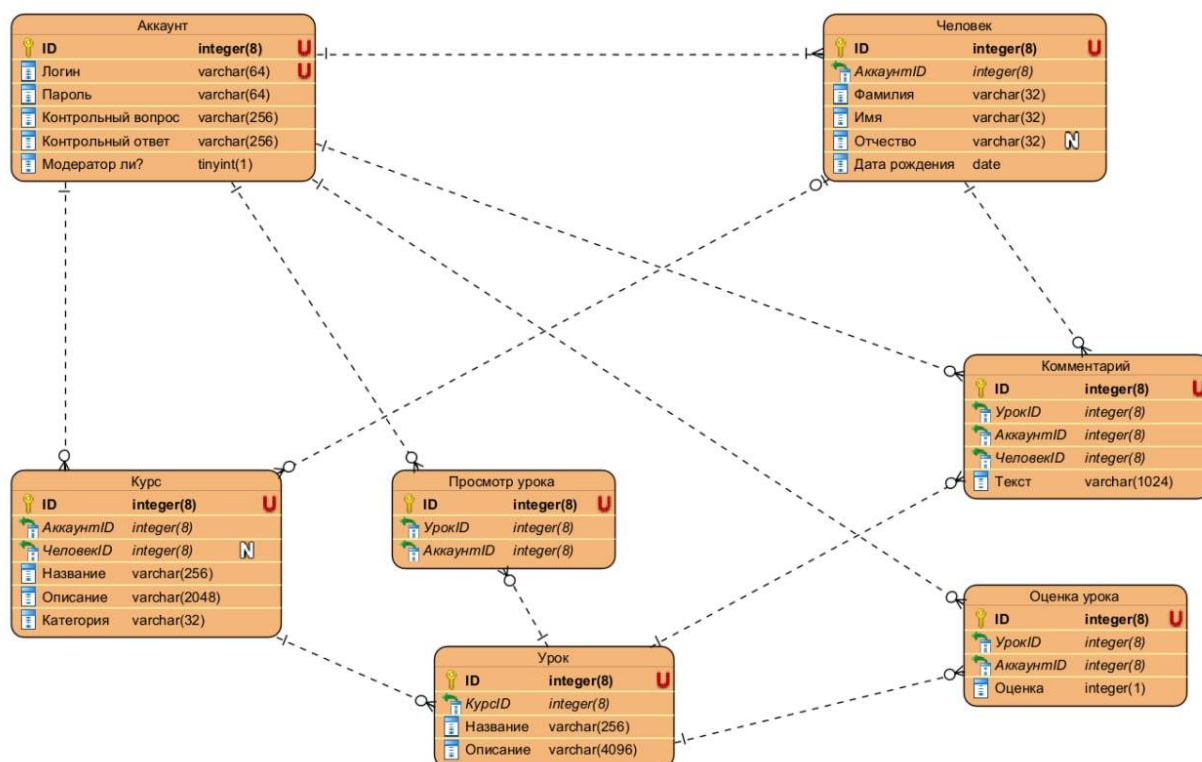


Рисунок 17 - ER-диаграмма базы данных веб-приложения