

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 удаленного изучения музыки»

Исполнители

\_\_\_\_\_ Ю.Г. Пьянин

\_\_\_\_\_ Р.Э. Мамедов

Заказчик

\_\_\_\_\_ В.С. Тарасов

## Содержание

1 Терминология .....	4
2 Общие сведения.....	7
1.1 Полное наименование системы и название приложения .....	7
1.2 Заказчик приложения .....	7
1.1 Исполнитель .....	7
1.2 Перечень документов, на основании которых создается система.....	7
1.3 Плановые сроки начала и окончания работы по созданию системы.....	8
1.4 Порядок оформления и предъявления заказчику результатов работы.....	9
3 Назначение и цели создания .....	10
3.1 Назначение проекта.....	10
3.2 Цели проекта.....	10
3.3 Задачи проекта.....	10
3.4 Предметная область .....	11
4 Требования к сайту и программному обеспечению .....	12
4.1 Требования к программному обеспечению сайта .....	12
5 Требования к оформлению и верстке.....	12
6 Структура сайта.....	14
6.1 Динамические страницы .....	14
7 Языковые версии сайта .....	15
8 Группы пользователей .....	16
8.1 Пользователь .....	16
9 Дизайн сайта.....	17
10 Навигация по сайту .....	17
10.1 Основное навигационное меню .....	18

11	Описания страниц сайта .....	19
11.1	Описание динамических страниц .....	19
12	Функциональность сайта .....	21
12.1	Формат предоставления материалов для сайта .....	21
13	Порядок контроля и приемки работ .....	21
ПРИЛОЖЕНИЕ .....		23

## 1 Терминология

Термин	Определение
Проект, система	Разрабатываемое веб-приложение
Личный кабинет клиента	Веб-страница, где отражаются персональная информация пользователя, введенная на этапе регистрации и/или отредактированная позже, переход к каталогу и переход к странице редактирования личных данных.
Frontend	Клиентская часть приложения. Отвечает за получение информации с программно–аппаратной части и отображение ее на устройстве пользователя.
Сервер, серверная часть	Компьютер, обслуживающий другие компьютеры (клиентов) и предоставляющий им свои ресурсы для выполнения определенных задач.
Клиентская сторона	Компьютер, использующий ресурсы сервера и предоставляющий пользователю возможность взаимодействия с системой.
Backend	Программно–аппаратная часть веб-приложения. Отвечает за

	функционирование внутренней части веб-приложения.
<b>GitHub</b>	Веб-сервис для хостинга IT-проектов и их совместной разработки.
<b>Фреймворк</b>	Программное обеспечение, облегчающее разработку и объединение разных компонентов большого программного проекта.
<b>Пользователь (Клиент)</b>	Авторизованный в системе человек, пользующийся функционалом веб-приложения.
<b>REST API (REST)</b>	Стиль архитектуры программного обеспечения для построения масштабируемых веб-приложений.
<b>Python</b>	Высокоуровневый язык программирования общего назначения с динамической строгой типизацией и автоматическим управлением памятью.
<b>Django</b>	Это высокоуровневый Python веб-фреймворк, который позволяет быстро создавать безопасные и поддерживаемые веб-сайты.
<b>База Данных (БД)</b>	Это упорядоченный набор структурированной информации или данных, которые обычно хранятся в

	электронном виде в компьютерной системе.
<b>Система управления базами данных (СУБД)</b>	Это набор программ, которые управляют структурой БД и контролируют доступ к данным, хранящимся в БД.
<b>SQLite</b>	Компактная встраиваемая БД для локальной разработки.
<b>PostgreSQL</b>	Это реляционная СУБД с открытым кодом.
<b>CSS</b>	Формальный язык описания внешнего вида веб-страницы, написанного с использованием языка разметки (HTML).
<b>HTML</b>	Стандартизированный язык разметки для просмотра веб-страниц в браузере.
<b>Bootstrap 5</b>	Свободный набор инструментов для создания сайтов и веб-приложений. Включает в себя html- и css-шаблоны.
<b>Dart</b>	Язык программирования созданный Google в качестве замены JS.
<b>Flutter</b>	Комплект средств разработки и фреймворк с открытым исходным кодом для создания мобильных приложений под Android и iOS, веб-приложений.

## **2 Общие сведения**

### **2.1 Полное наименование системы и название приложения**

Полное наименование: «Платформа для удаленного изучения музыки Music Trainee».

Название приложения: Music Trainee.

### **2.2 Заказчик приложения**

Факультет компьютерных наук

Кафедра программирования и информационных технологий

Тарасов Вячеслав Сергеевич – старший преподаватель

### **2.3 Исполнитель**

Факультет компьютерных наук

Кафедра информационных технологий управления

3 курс, 1 группа

— Состав команды разработчика:

— Студент Пьянин Юрий Геннадьевич;

— Студент Мамедов Ровшан Эльшанович.

### **2.4 Перечень документов, на основании которых создается система**

— 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;

— Федеральный закон "О коммерческой тайне" от 29.07.2004 N 98–ФЗ;

— Закон РФ от 07.02.1992 N 2300-1 (ред. от 11.06.2021) "О защите

прав потребителей";

— ГОСТ 34.602-89.

## 2.5 Плановые сроки начала и окончания работы по созданию системы

Плановый срок начала работ – март 2024

Плановый срок окончания работ – июнь 2024

Основные этапы работ, их содержание и их приблизительные сроки приведены в таблице ниже:

Номер этапа	Название этапа	Содержание работ	Порядок приемки документов	Сроки
1	Составление Технического Задания	Разработка функциональных и нефункциональных требований	Утверждение технического задания	До 19.03.24 23:59 по МСК
2	Техническое проектирование	Разработка дизайн-макета веб-приложения	Описание функциональной схемы в онлайн-сервисе Miro предоставление дизайна проекта на сервисе Figma	До 19.03.24 23:59 по МСК
3	Разработка программной части	Разработка серверной и клиентской частей и системы хранения данных	Приемка осуществляется во время испытаний	С 20.03.24 00:00 по МСК до 30.05.24



				23:59 по МСК
4	Разработка курсового проекта	Разработка Курсового проекта, содержащего аналитическую информацию о проекте	На основе Технического Задания	С 31.05.24 00:00 по МСК до 10.06.24 23:59 по МСК

## 2.6 Порядок оформления и предъявления заказчику результатов работы

Работы по созданию проекта сдаются Исполнителем поэтапно. По окончании каждого из этапов работ Исполнитель сдает Заказчику соответствующие отчетные документы этапа, а именно:

- Техническое задание по ГОСТ 34.602–89;
- Аналитику проекта;
- Исходный код системы;
- Исполняемые модули;
- Курсовой проект на основе Технического задания.

Вся документация должна быть подготовлена в печатном и электронном виде и размещена на GitHub.

### **3 Назначение и цели создания**

#### **3.1 Назначение проекта**

Создание веб-приложения по обучению музыки с возможностью публикации платных и бесплатных курсов в каталоге и их прохождение, а также с возможностью оставлять комментарии и оценки курсам.

#### **3.2 Цели проекта**

- Создание веб-приложение, которое предоставляет пользователю возможность выбора между созданием курса и его созданием;
- Реализация комментариев, и рейтинговой системы к занятиям и курсам;
- Реализация модели оплаты по qr-коду.

#### **3.3 Задачи проекта**

- Обеспечение возможности авторизации и регистрации пользователей;
- Обеспечение возможности восстановления доступа к аккаунту пользователя;
- Обеспечение возможности выбора возможности взаимодействия с платформой (преподавание или прохождение обучения);
- Обеспечение возможности оплаты по QR-коду;
- Обеспечение возможности оценивания и комментирования курсов;
- Обеспечение формы для создания курса (название, описание, содержание);
- Обеспечение возможности редактирования информации в личном кабинете;

### **3.4 Предметная область**

- Отношения, возникающие между потребителями (клиентами) и владельцами предоставляемых услуг регулируются настоящим законом РФ от 07.02.1992 N 2300-1 (ред. от 05.12.2022) «О защите прав потребителей»;
- Информация и персональные данные, размещающиеся на сайте и хранящиеся в базе данных веб-приложения защищены в соответствии с Федеральным законом от 27.07.2006 N 152-ФЗ «О персональных данных».

## **4 Требования к сайту и программному обеспечению**

- Приложение должно реализовывать основные задачи, стоящие перед проектом;
- Приложение должно иметь архитектуру, соответствующую шаблону Клиент-Серверного приложения и иметь разделение на backend и frontend, связанных между собой REST API.

### **4.1 Требования к программному обеспечению сайта**

Страницы сайта должны корректно отображаться в браузерах Google Chrome версии 90.0.4430.93 и выше.

Для реализации программно-аппаратной части были выбраны следующие технологии:

- Язык программирования Python;
- Фреймворк Django;
- СУБД SQLite – для локальной разработки;
- СУБД PostgreSQL – при развертывании на удаленном сервере.

Для реализации клиентской части были выбраны следующие технологии:

- Язык разметки HTML;
- Язык стилей CSS;
- Язык Dart;
- Фреймворк Flutter.

### **4.2 Требования к оформлению и верстке**

Оформление страниц веб-приложения должно быть выдержано в едином стиле и умеренной цветовой палитре, а также содержать удобочитаемые шрифты и размер текста.

При верстке страниц используются язык разметки HTML и каскадные таблицы стилей CSS.

#### **4.3 Требования к системе администрирования**

Для обслуживания системы необходимы администраторы, которые должны помогать пользователям с решением их проблем. Обращение администратор обрабатывает через почту и отвечает в индивидуальном порядке.

## **5 Структура сайта**

### **5.1 Статические страницы**

- Страница с условиями публикации курсов для преподавателей.

### **5.2 Динамические страницы**

- Личный кабинет для пользователей;
- Каталог курсов;
- Страница курса;
- Страница оплаты;
- Страница создания курса;
- Страница курсов пользователя;
- Страница выбора занятия;
- Страница занятия;
- Регистрация пользователей;
- Авторизация пользователей;
- Восстановление пароля пользователей;

## **6 Языковые версии сайта**

Все страницы сайта должны быть реализованы с поддержкой русской языковой версии.

## **7 Группы пользователей**

Разрабатываемая система должна содержать следующие виды пользователей:

— Пользователь;

### **7.1 Пользователь**

— Просмотр каталога;

— Просмотр информации о курсе;

— Просмотр комментариев и отзывов;

— Приобретение курса;

— Получение доступа к курсу;

— Оплата по qr-коду;

— Создание курса;

— Просмотр своих курсов;

— Выбор курса;

— Оценивание курсов и занятий;

— Написание комментариев к занятию;

— Прохождение обучения (теории, тестов, сдача домашнего задания);

— Авторизация;

— Просмотр и редактирование личных кабинетов;



## 8 Дизайн сайта

- Все страницы должны быть выполнены в одном стиле;
- Все страницы должны корректно отображаться на экранах;
- Верстка страниц должна соответствовать ТЗ;
- Все элементы управления сайтом должны быть заметны и выделены на общем фоне;
- Основное меню сайта должно находиться в верхней части каждой страницы;
- Макет страниц должен разрабатываться в Figma;
- Черновой вариант заглушек при разработке серверной части приложения должен представлять собой страницы, которые необходимо сверстать с помощью HTML + CSS. Необходимо для концептуального представления страниц сайта;
- Конечный вариант будет реализован на Flutter, основываясь на макетах в Figma.

## **9 Навигация по сайту**

### **9.1 Основное навигационное меню**

Логотип закреплен в верхнем левом углу. Основное меню навигации на экранах ПК расположено в верхней части экрана (в шапке). Оно позволяет переходить на страницы:

- Каталог (главная страница);
- Мои курсы;
- Преподавание;
- Личный кабинет.

## **10 Описания страниц сайта**

### **10.1 Описание статических страниц**

Страница «Преподавание»:

- Страница с пользовательским соглашением о публикации курса;
- Кнопка «Создать курс».

### **10.2 Описание динамических страниц**

Страница «Личный кабинет» пользователя включает в себя:

- Страница с формой редактирования личных данных;
- Страница «Каталог курсов» включает в себя:
- Список курсов доступных на платформе;
- Название курсов
- Краткое описание курсов;
- Стоимость каждого курсов;

Страница «Курс» включает в себя:

- Название курса;
- Описание курса;
- Цена курса;
- Кнопку «Купить»;

Страница «Оплаты» включает в себя:

- Форму для оплаты курса;
- Кнопку «Оплатить»

Страница «Преподавание» включает в себя:

- Текст с соглашением о предоставлении услуг;

— Кнопку «Создать курс»

Страница «Создания курса» включает в себя:

— Форму для создания курса;

— Кнопку «Создать»

Страница «Курсы пользователя» включает в себя:

— Список с кнопками всех курсов пользователя;

Страница «Выбора занятия» включает в себя:

— Список с кнопками занятий доступных в курсе;

Страница «Занятия» включает в себя:

— Тему занятия (теория, тесты или домашнее задание)

— Кнопку «Продолжить»

Страница «Авторизация пользователей» включает в себя:

— Форму авторизации;

— Кнопку «Регистрация»;

— Кнопку «Войти»;

Страница «Регистрация пользователей» включает в себя:

— Форму для регистрации;

— Кнопку «Зарегистрироваться»;

Страница «Восстановление пароля пользователей» в себя:

— Форму для ввода почты;

— Кнопку «Отправить код»;

— Форму для ввода кода;

— Кнопку «Отправить код»

## **11 Функциональность сайта**

Веб-приложение предоставляет возможности для:

- Просмотра каталога курсов;
- Приобретение курсов;
- Прохождение курсов;
- Оценивание курсов и написание комментариев к занятиям
- Авторизации;
- Доступ к личному кабинету пользователя;
- Контент и наполнение сайта

Основным наполнением сайта служит сущность «Каталог», с помощью которой пользователь имеет возможность перечень музыкальных курсов доступных на платформе, а также пополнять каталог своими курсами.

### **11.1 Формат предоставления материалов для сайта**

Все материалы, необходимые для реализации веб-приложения, должны быть размещены на GitHub.

### **11.2 Порядок контроля и приемки работ**

Контроль разработки веб-приложения осуществляется путем обсуждения текущего этапа разработки среди исполнителей, а также путем встреч с заказчиком.

Готовое веб-приложение со всей необходимой документацией предоставляется заказчику в обозначенный им срок.

За заказчиком остается осуществление проверки соответствия функциональности приложения согласно Техническому Заданию.

Исполнитель также обязан предоставить заказчику следующий комплект, необходимый для защиты проекта:

- Техническое Задание;
- Курсовой Проект;
- Демонстрационная версия приложения со всеми основными сценариями;
- Исходный код приложения.

Вся сопроводительная документация предоставляется в печатном виде и в электронном виде, размещенная на GitHub.

## ПРИЛОЖЕНИЕ

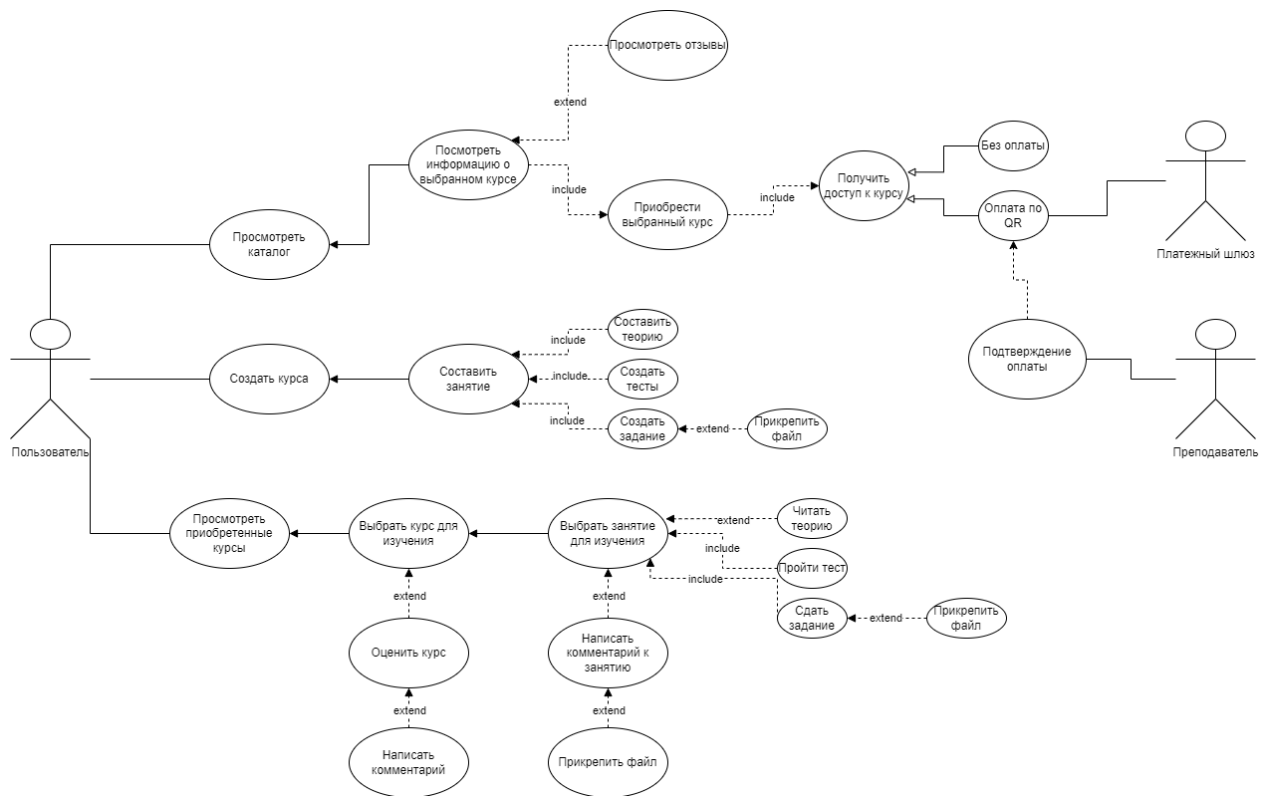


Рисунок 1 - Use Case Диаграмма Пользователя

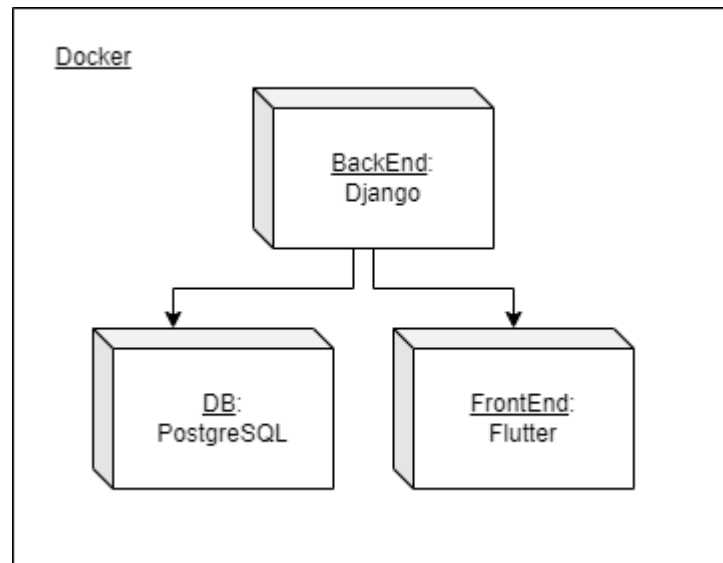


Рисунок 2 - Диаграмма развертывания

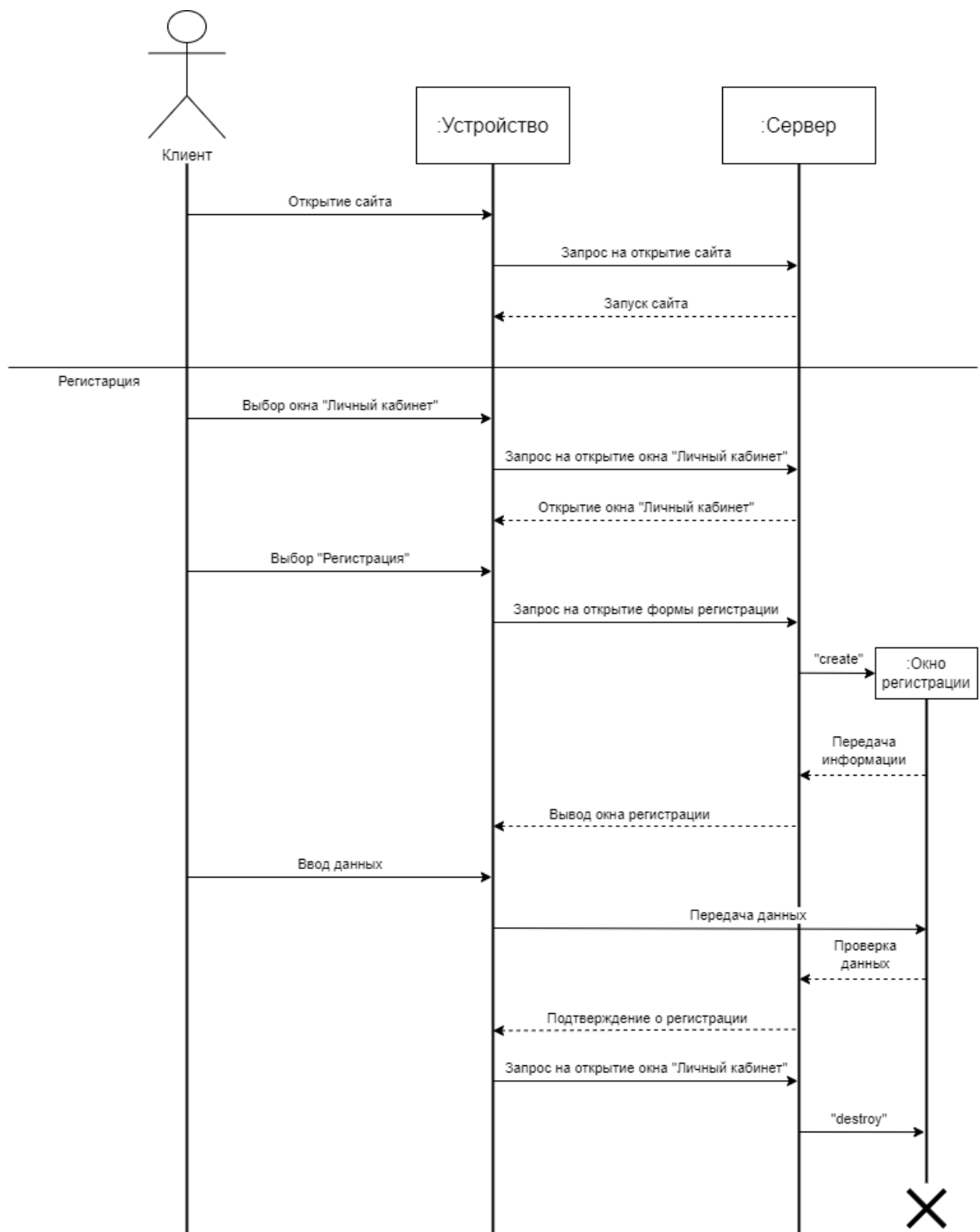


Рисунок 3 - Диаграмма Последовательности для регистрации



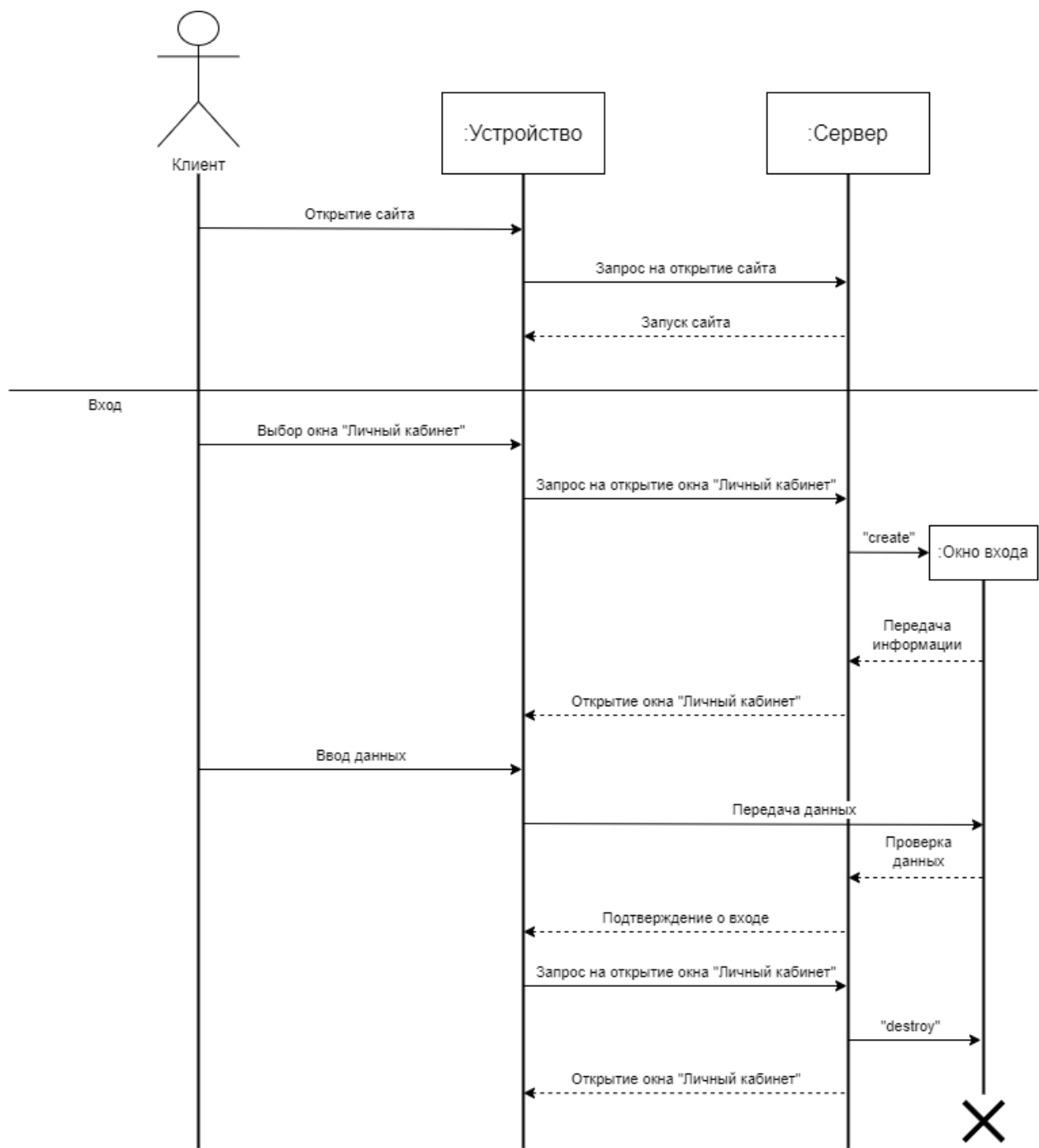


Рисунок 4 - Диаграмма Последовательности для входа

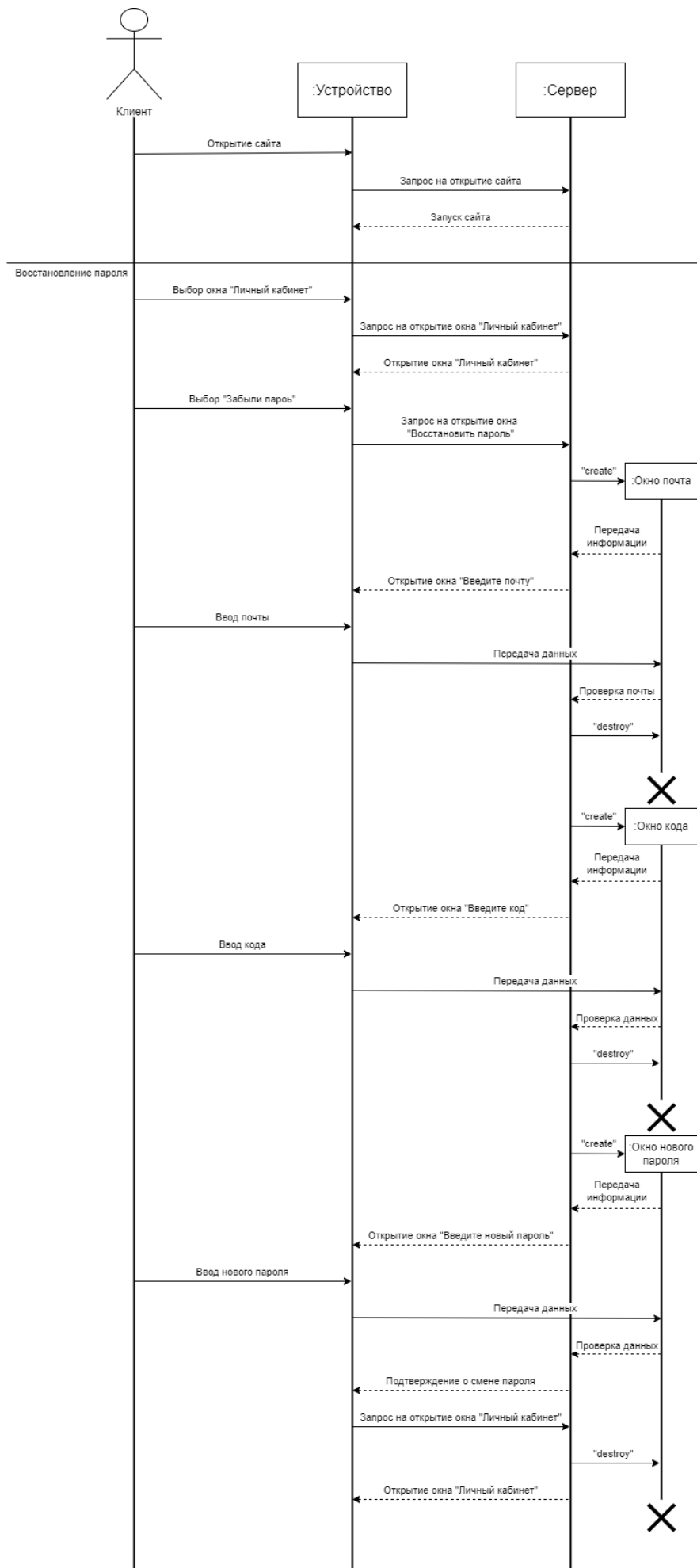


Рисунок 5 - Диаграмма Последовательности для восстановления пароля

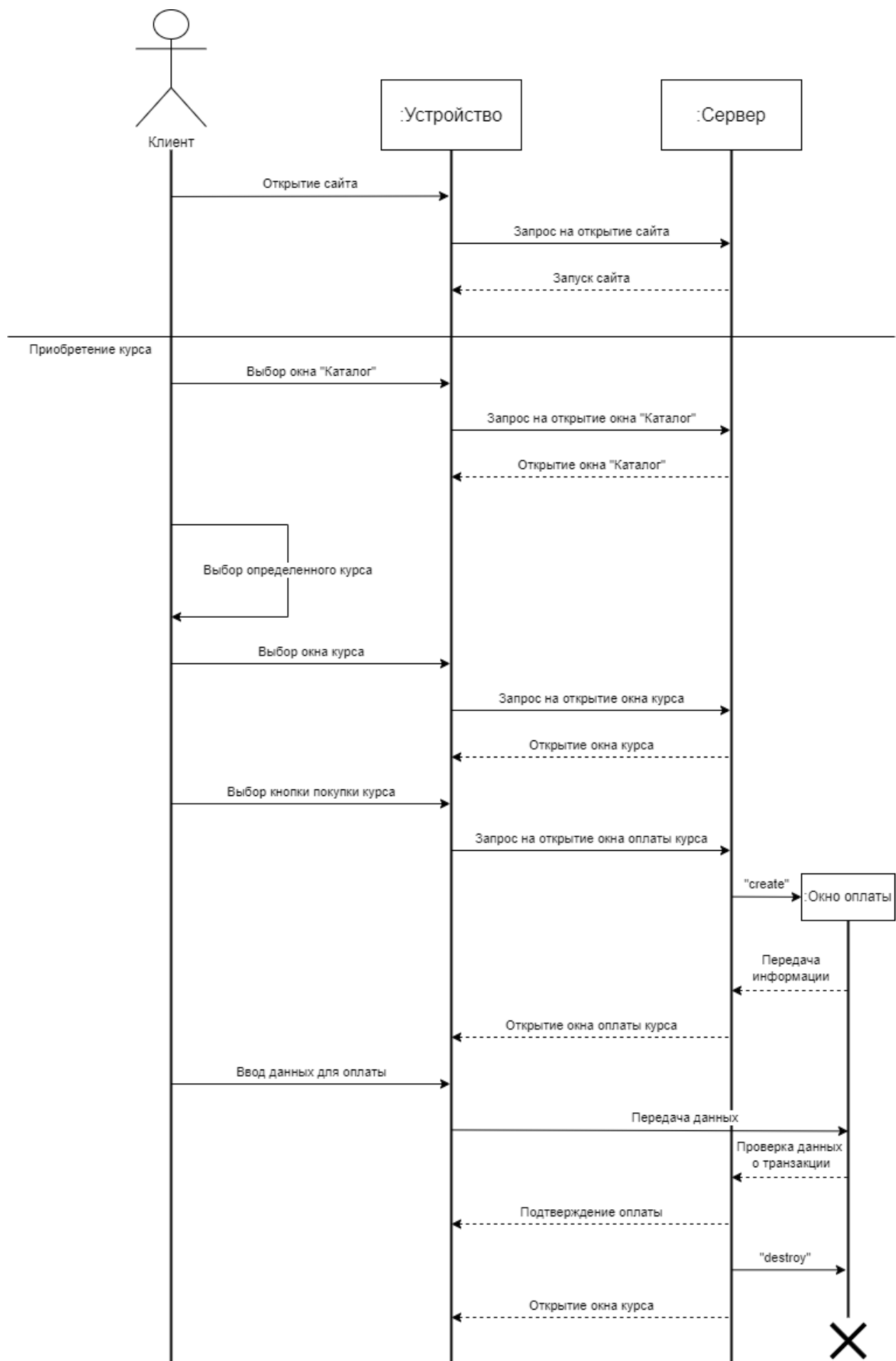


Рисунок 6 - Диаграмма Последовательности для приобретения курса

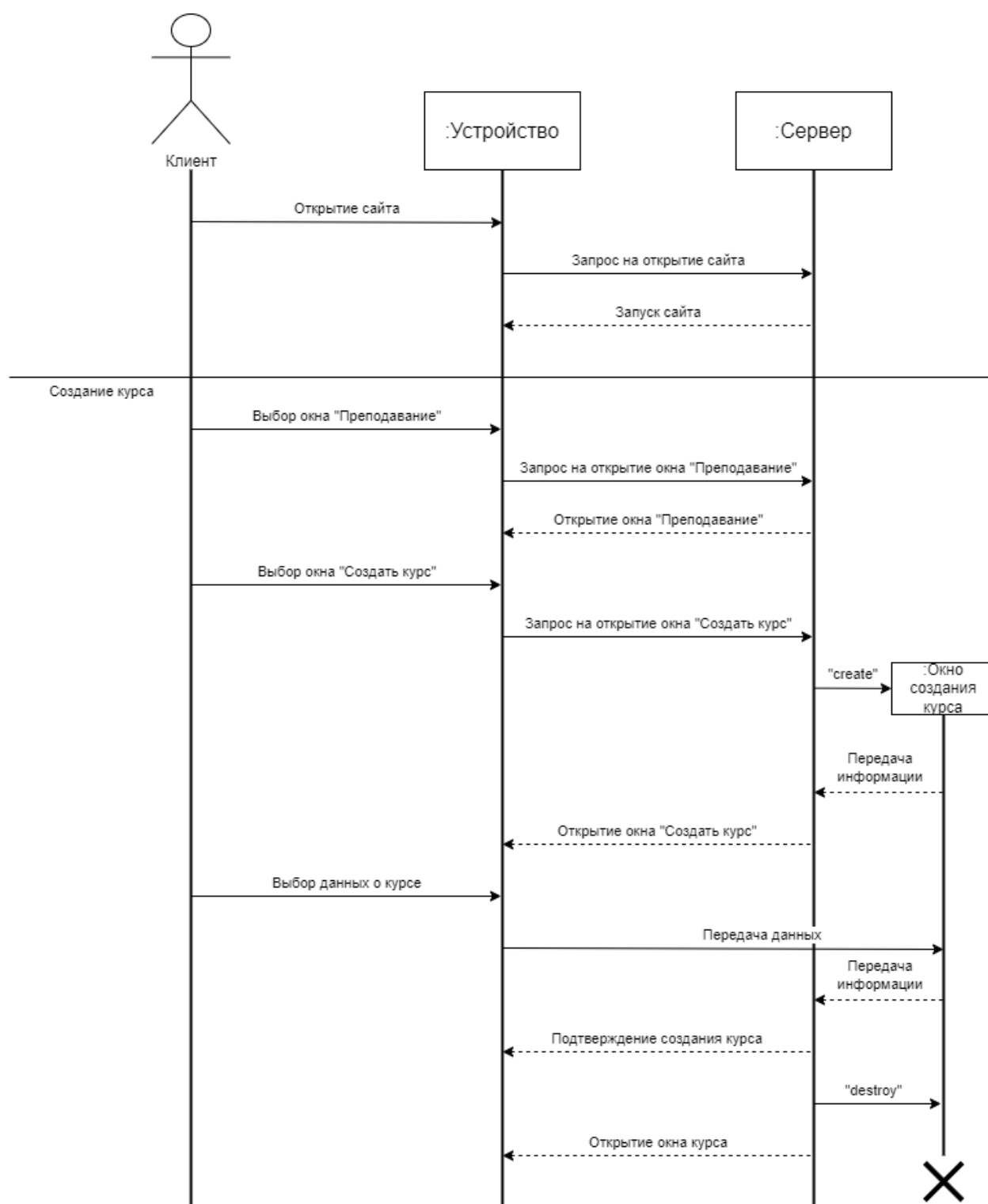


Рисунок 7 - Диаграмма Последовательности для создания курса

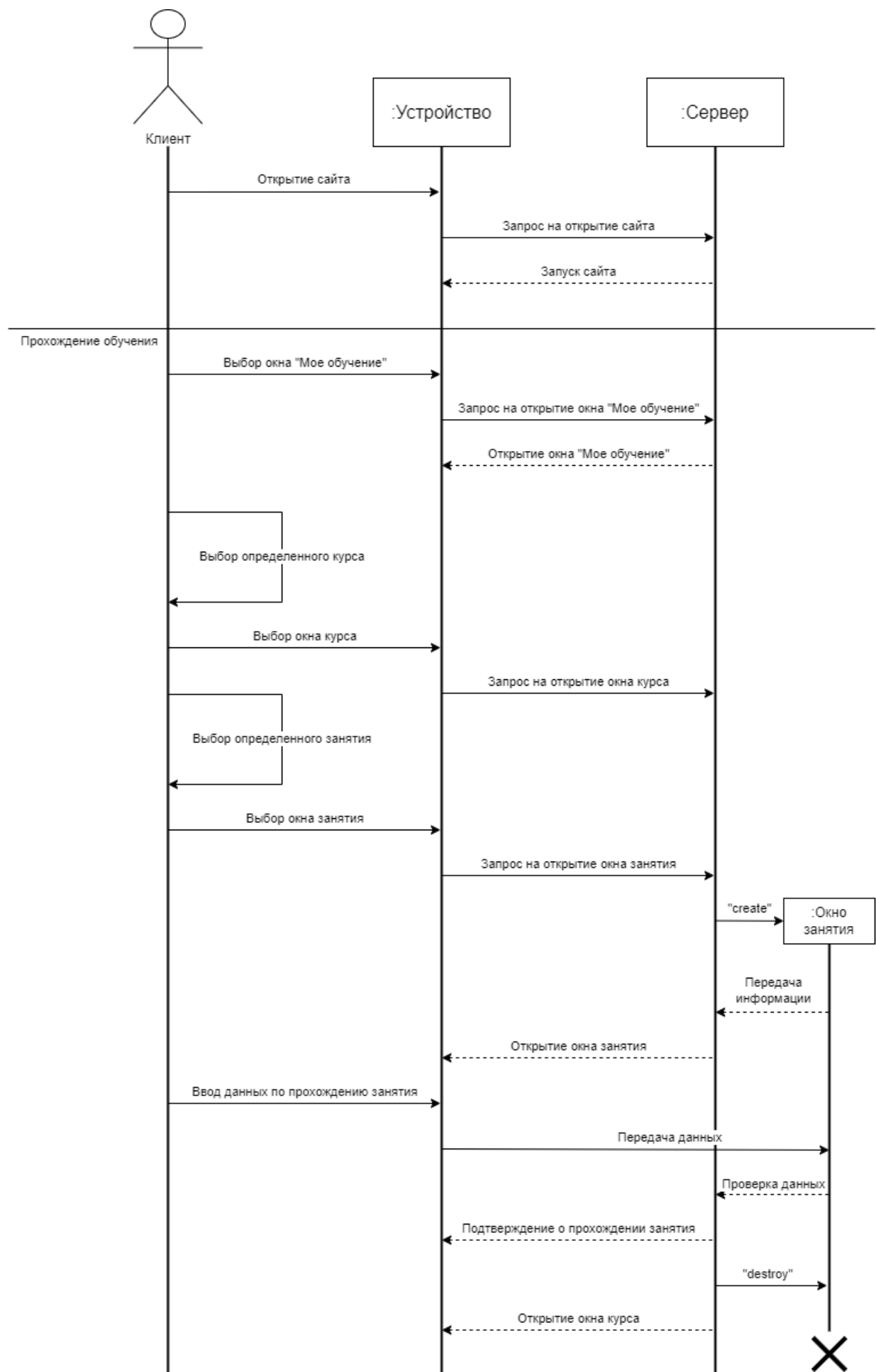


Рисунок 8 - Диаграмма Последовательности для прохождения обучения

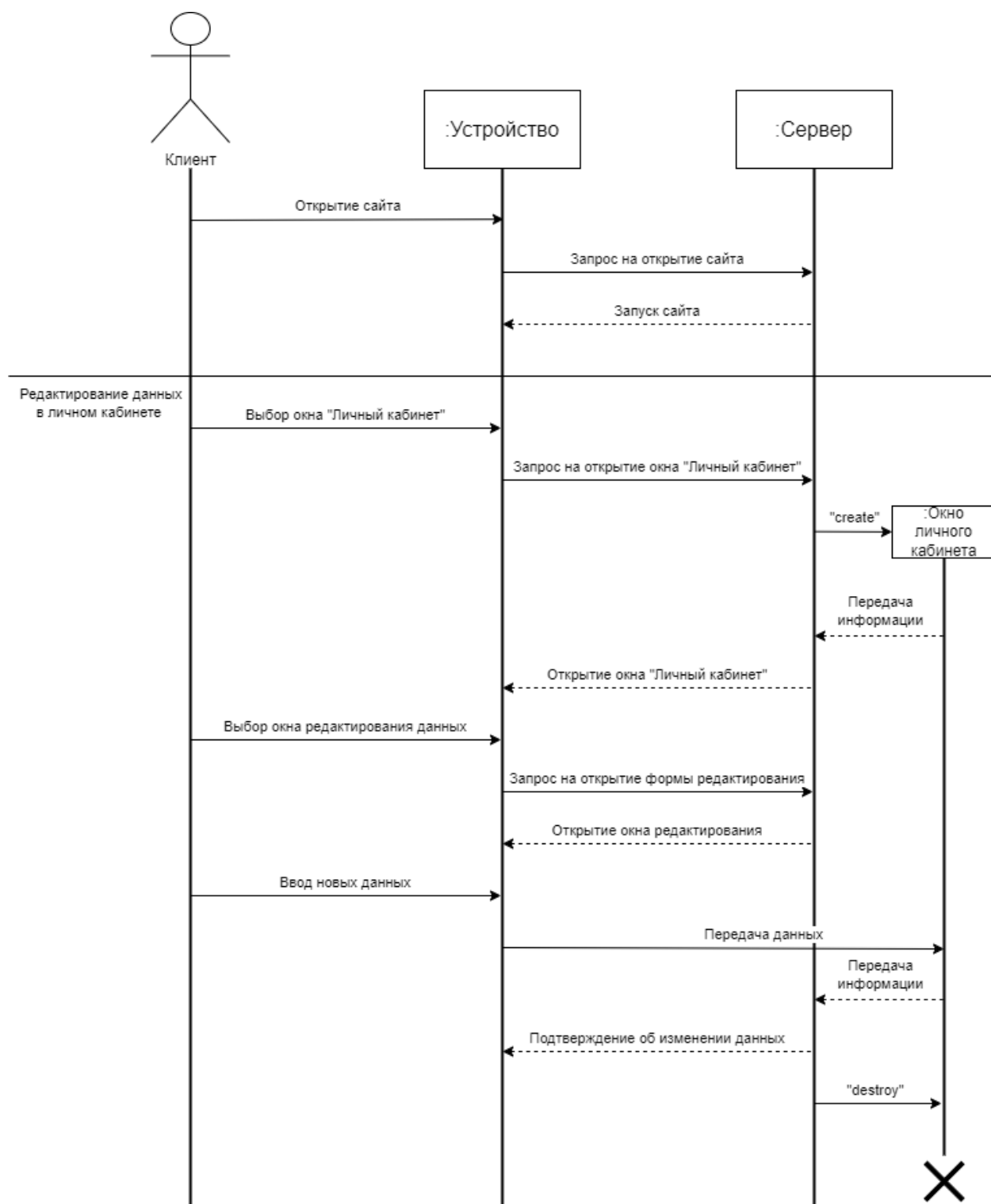


Рисунок 9 - Диаграмма Последовательности для редактирования информации в личном кабинете



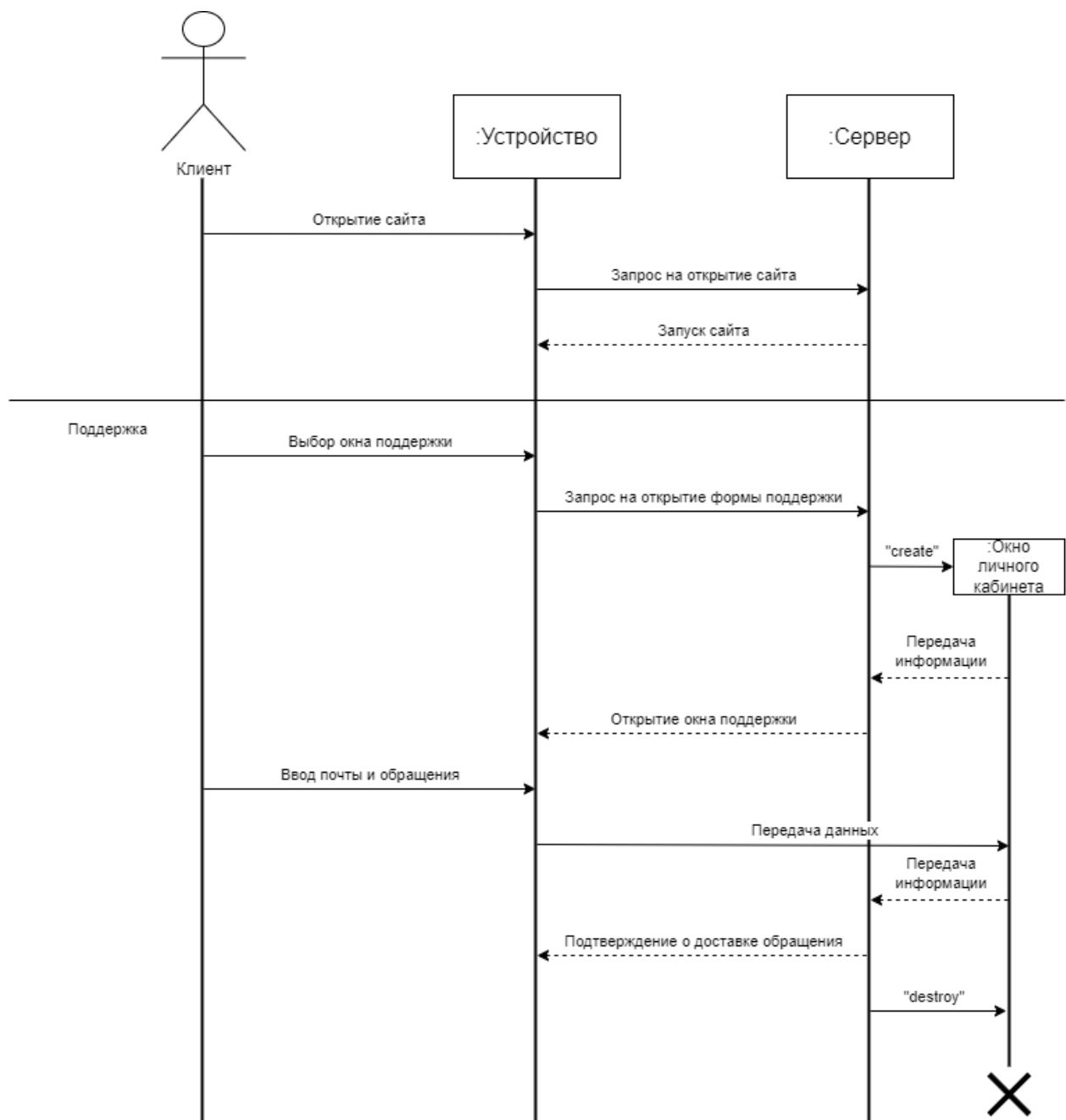


Рисунок 10 - Диаграмма Последовательности для связи с поддержкой

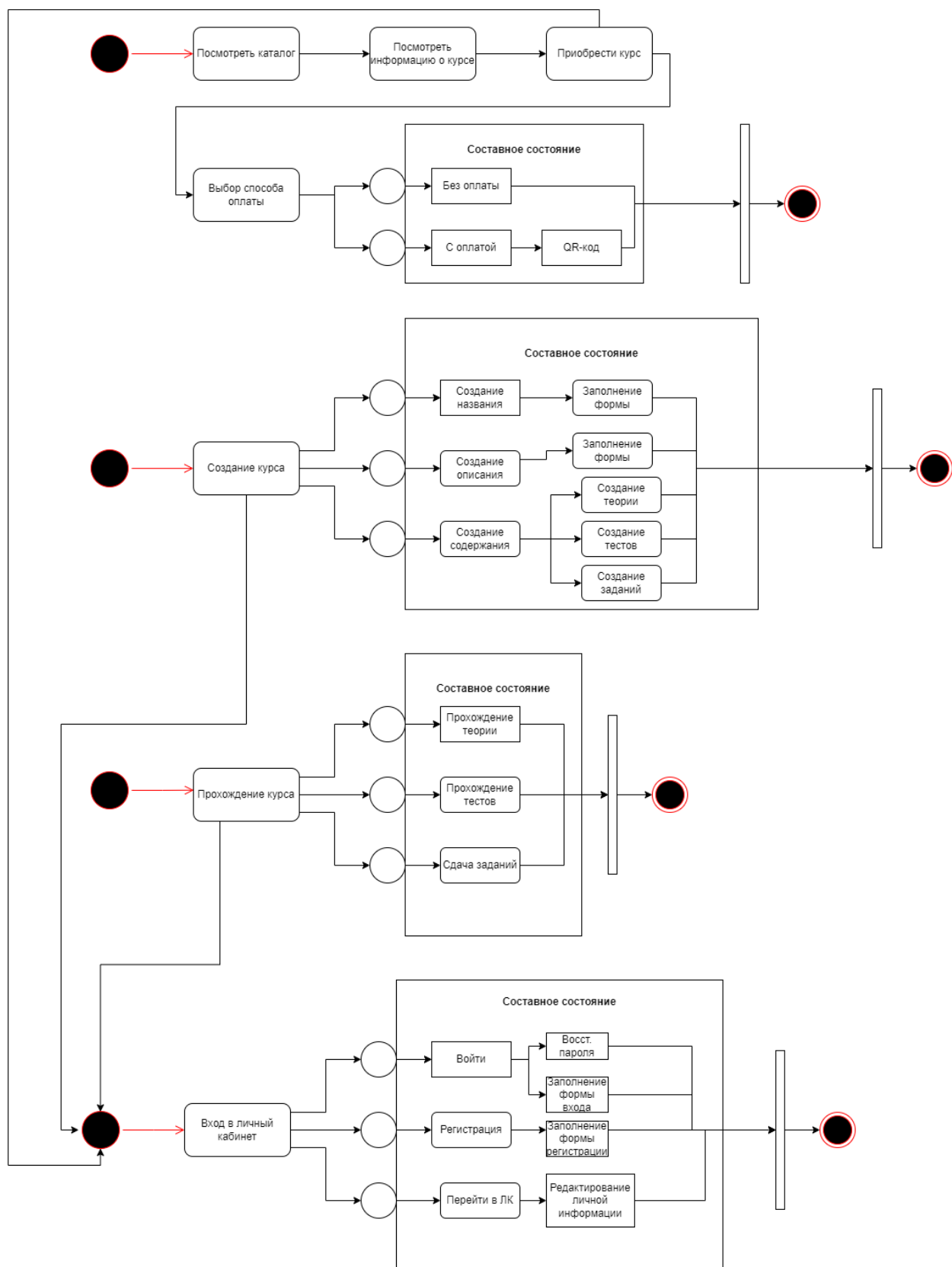


Рисунок 11 - Диаграмма Состояний

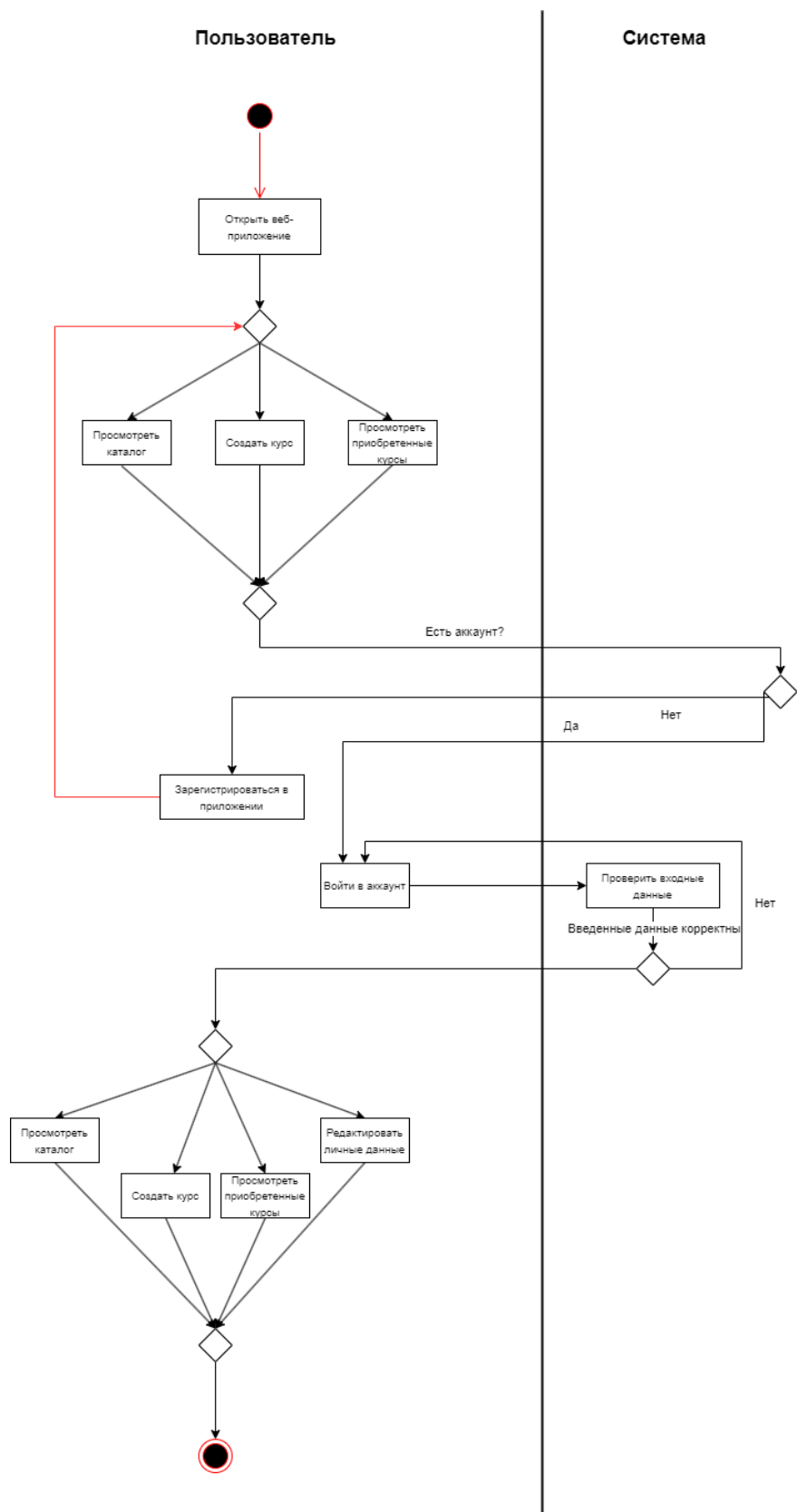


Рисунок 12 - Диаграмма Активности Пользователя

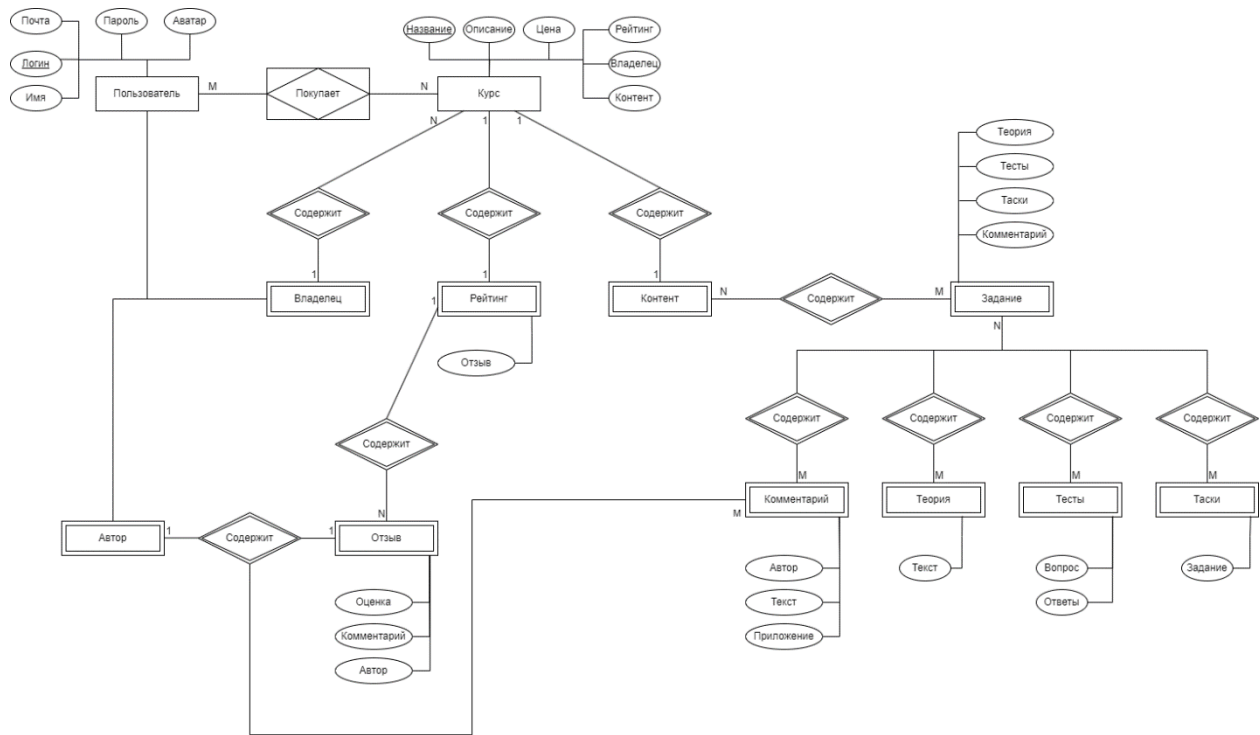


Рисунок 13 - Концептуальная ER диаграмма Пользователя

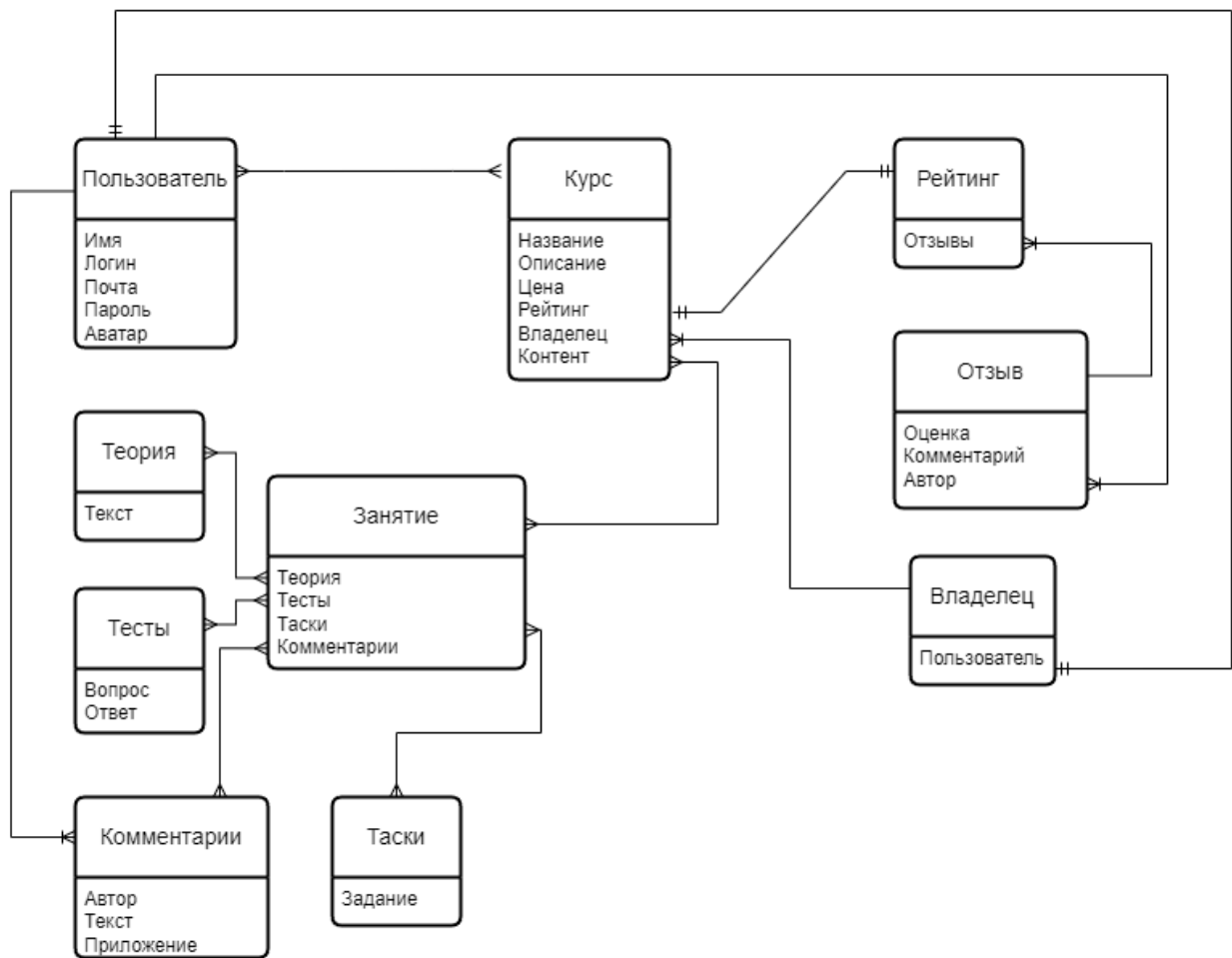


Рисунок 14 - Логическая ER диаграмма Пользователя