ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ОБРАЗОВАНИЯ МОСКОВСКОЙ ОБЛАСТИ

Государственный университет «Дубна»

Институт системного анализа и управления

Кафедра системного анализа и управления

КУРСОВАЯ РАБОТА

ПО ДИСЦИПЛИНЕ

**ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННЫХ СИСТЕМ**

«Автоматизация создания тестового окружения»

Выполнил: студент группы 3253

3 курса ИСАУ

**Егоров Александр Ильич**

**Руководитель: аспирант???**

**Йцу йцу йцу**

Дубна, 2021

ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ОБРАЗОВАНИЯ МОСКОВСКОЙ ОБЛАСТИ

Государственный университет «Дубна»

Институт системного анализа и управления

Кафедра системного анализа и управления

КУРСОВАЯ РАБОТА

ПО ДИСЦИПЛИНЕ

**ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННЫХ СИСТЕМ**

«Автоматизация создания тестового окружения»

Выполнил: студент группы 3253

3 курса ИСАУ

Егоров Александр Ильич

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

(подписи студентов)

Руководитель: старший преподаватель

Миловидова Анна Александровна

Оценка: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата защиты: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись преподавателя)

Дубна, 2021

Содержание

[Введение 4](#_Toc73613587)

[1. Анализ предметной области 5](#_Toc73613588)

[2. Постановка задачи 6](#_Toc73613589)

[3. Требования к информационной системе 8](#_Toc73613590)

[4. Функциональная модель системы 9](#_Toc73613591)

[**4.1.** **Диаграмма вариантов использования** 9](#_Toc73613592)

[**4.2.** **Детальное описание прецедентов** 11](#_Toc73613593)

[**4.3.** **Декомпозиция основных функций ИС** 12](#_Toc73613594)

[***Диаграммы деятельности*** 12](#_Toc73613595)

[***Диаграммы последовательности ИС*** 14](#_Toc73613596)

[5. Диаграмма классов 18](#_Toc73613597)

[6. Модель данных 20](#_Toc73613598)

[**6.1.** **Логический уровень модели данных** 20](#_Toc73613599)

[**6.2.** **Физический уровень модели данных** 22](#_Toc73613600)

[7. Интерфейс информационной системы 29](#_Toc73613601)

[Заключение 41](#_Toc73613602)

[Список литературы 42](#_Toc73613603)

# Введение

Данная курсовая работа представляет собой систему, где разработчики смогут сами поднимать и удалять тестовое окружение для своего кода. Модели информационной системы построены с учетом принципов объектно-ориентированного анализа и правил языка *UML* [1].

Актуальность данной работы заключается в уменьшении времени ожидания разработчика для тестирования своего кода в обстановке, приближенной к продакшену. Польза от данной систему увеличивается при росте штата сотрудников одной компании, где количество готовых стендов ограничено.

# Анализ предметной области

Предметная область – материальная система или система, характеризующая элементы материального мира, информация о которых хранится и обрабатывается [3].

В данной курсовой работе предметной областью является работа связанная с созданием тестового окружения компани WebBee. Компания занимается разработкой веб приложений и параллельно их тестированием. Для проведения своих тестов, а так же для проверки действия кода, разработчики и тестировщики используют тестовое окружение – стенды.

**Стенды** создаются DevOps-инженерами на основе данных, которые были получены от разработчиков. Процесс создания делится на несколько этапов: выделение требований и создание задач в Confluence и Jira соответственно, создание тестового окружения, заполнение отчетности. Для создания всего конвейера сборки и выкладки DevOps-инженеры создают у себя отдельную ветку, доступ к которой имеют только они.

После создания pipeline, разработчики заверяют соответствие работоспособности стенда с указанными ими требованиями по составленной документации. Если где-то есть несоответствия, идет доработка. В случае если стенд успешно собирается и правильно функционирует, то задача закрывается и стенд переходит к разработчикам и тестировщикам на постоянное использование.

В случае нахождения неисправностей или дополнения стенда исходя из личных предпочтений, создается отдельная задача в результате в результате которой, дорабатывается уже готовая документация.

# Постановка задачи

**Цель**

Разработка ИС по созданию и удалению тестового окружения в платформы облачных вычислений OpenNebula.

**Исходные данные**

Список сотрудников и их учетные данные, документация по используемым инструментам, рабочая платформа OpenNebula и Jenkins.

**Априорные модельные представления**

**Клиенту** в системе должны быть доступны следующие функции:

* вход в систему,
* создание виртуальной машины,
* удаление вируальной машины,
* изменение виртуальной машины,
* изменение количества ресурсов для машины.

**Администратор имеет возможности:**

* регистрация пользователей в системе,
* создание виртуальной машины,
* удаление виртуальной машины
* вход в систему,
* изменение виртуальной машины,
* изменение количества ресурсов для машины.

**Ожидаемый результат**

Проект и реализация ИС, соответствующие априорным модельным представлениям.

**Критерии оценки результата**

Результатами внедрения информационной системы являются:

* автоматическое создание и удаление окружения,
* отсутствие проблемы ожидания свободного стенда,
* скорость полного создания стенда при скорости интернета 5 Мб/с и размере Docker образа 1 Гб будет меньше 10 минут.

**Средства проектирования и реализации**

Для создания моделей выбрано программное средство, как *DrawIo* и *Process Modeler*. Для изображения графического интерфейса использовался инструмент *Figma.* Для написания кода используется редактор VS Code.

# Требования к информационной системе

**Функциональные требования**

См. априорные модельные представления.

**Требования к данным**

В системе должны храниться данные о пользователях и их созданных машинах, времени создания, ресурсах, используемых для стенда.

**Требования к интерфейсам**

Удобный интуитивно понятный интерфейс с подписанием всех форм для ввода и с защитой от фиктивных или заведомо ложных, опасных для системы данных.

**Требования к безопасности данных**

Система должна удовлетворять требованиям федерального закона № 152-ФЗ «О персональных данных» и ФЗ от 27.07.2006 N 149 «Об информации, информационных технологиях и о защите информации».

# Функциональная модель системы

## **Диаграмма вариантов использования**

Диаграммы вариантов прецендентов – это тип поведенческой диаграммы UML, который часто используется для анализа различных систем [2]. Они позволяют визуализировать различные типы ролей в системе и то, как эти роли взаимодействуют с системой.

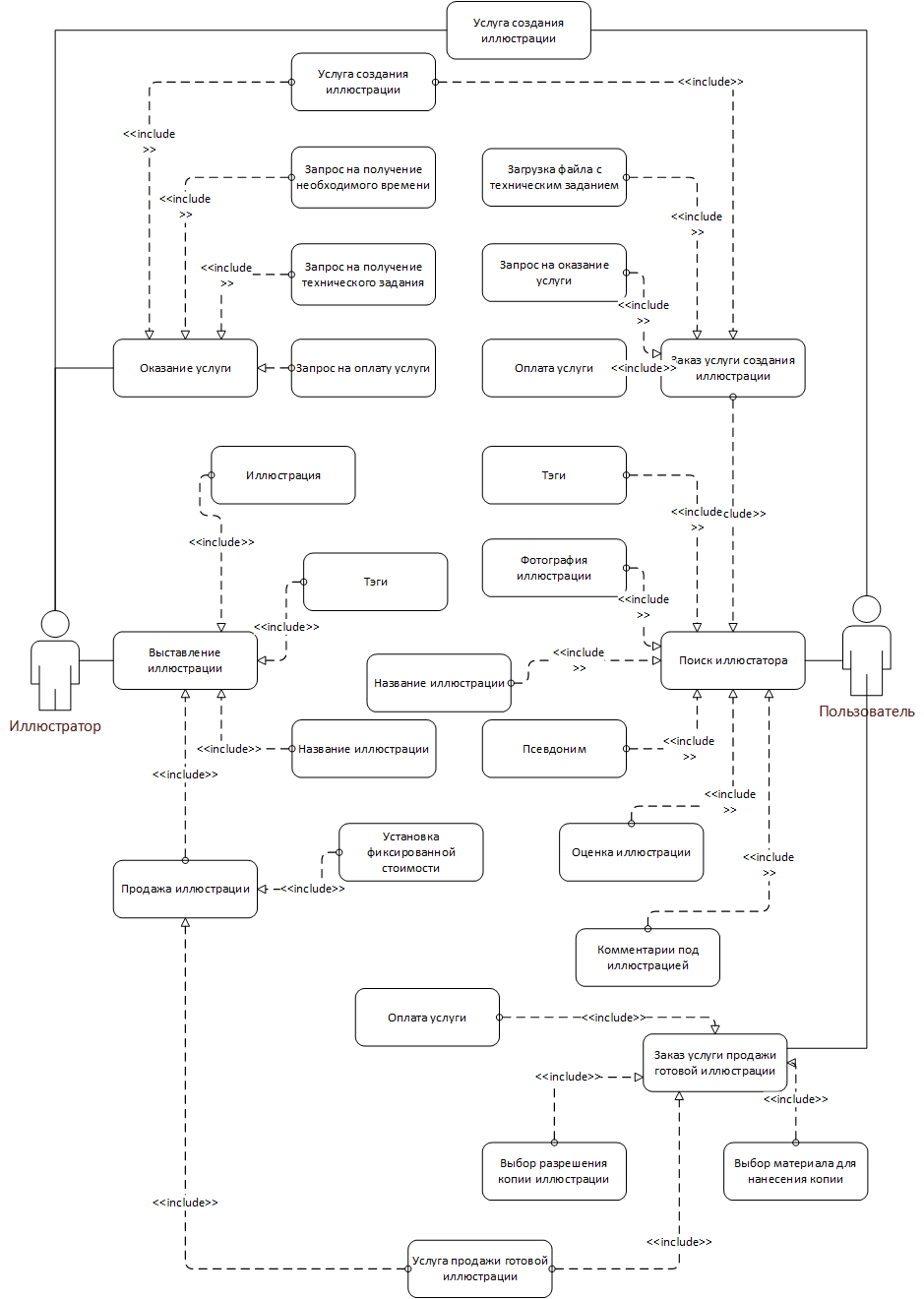


Рисунок 1. Use-case ИС

## **Детальное описание прецедентов**

**Регистрация и авторизация**

Таблица 1. Описание прецедента «Регистрация и авторизация»

|  |  |
| --- | --- |
| Название прецедента | Регистрация и авторизация |
| Основное действующее лицо | Пользователь и художник |
| Цель | Регистрация и авторизация |
| Триггер | Пользователь хочет получить доступ к системе |
| Результат | Пользователь получает доступ к системе |

**Основная последовательность**

1. Пользователь заполняет данные для регистрации.
2. Нажимает кнопку «Зарегистрироваться».
3. Система проверяет корректность данных и открывает форму для входа и отправляет данные в базу данных.
4. Пользователь вводит данные для входа.
5. Нажимает кнопку «Войти».
6. Система проверяет введенные данные и открывает раздел «главная страница».

**Выставление художественной работы**

Таблица 2. Описание прецедента «Выставление художественной работы»

|  |  |
| --- | --- |
| Название прецедента | Выставление художественной работы |
| Основное действующее лицо | Пользователь и художник |
| Цель | Выставление художественной работы |
| Триггер | Художник решает загрузить свою работу, указывает название, ставит тэги |
| Результат | Художник загружает свою работу, пользователь её просматривает, оценивает и оставляет комментарии |

**Основная последовательность**

1. Художник загружает работу, указывает её название, устанавливает хештеги
2. Система сохраняет рисунок в БД, создает в разделе «Портфолио» новую ссылку в виде иконки.
3. Пользователь нажимает на псевдоним художника и переходит в раздел «Портфолио».
4. Пользователь нажимает на иконку выложенной работы и
5. Ставит оценку данной работе.
6. Оставляет комментарий под работой.
7. Нажимает на кнопку «купить», расположенную ниже рисунка, и заказывает себе физическую копию.

## **Декомпозиция основных функций ИС**

Диаграмма деятельности — это технология, позволяющая описывать логику процедур, бизнес-процессы и потоки работ [3].

### ***Диаграммы деятельности***

Рассмотрим диаграммы информационной системы для творческой площадки.

На рисунке 2 изображена диаграмма деятельности, описывающая взаимодействие между пользователем и художником через систему и демонстрирующая процесс «Заказ иллюстрации».

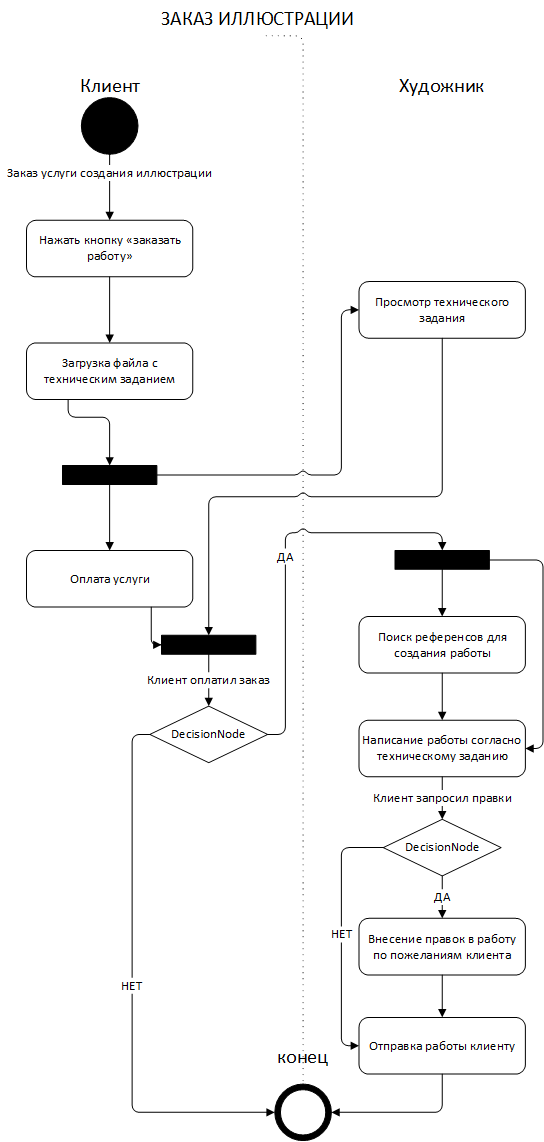


Рис. 2. Заказ иллюстрации

### ***Диаграммы последовательности ИС***

Диаграммы последовательности (*sequence diagram*) являются видом диаграмм взаимодействия языка *UML*, которые описывают отношения объектов в различных условиях. Условия взаимодействия задаются сценарием, полученным на этапе разработки диаграмм вариантов использования [4].

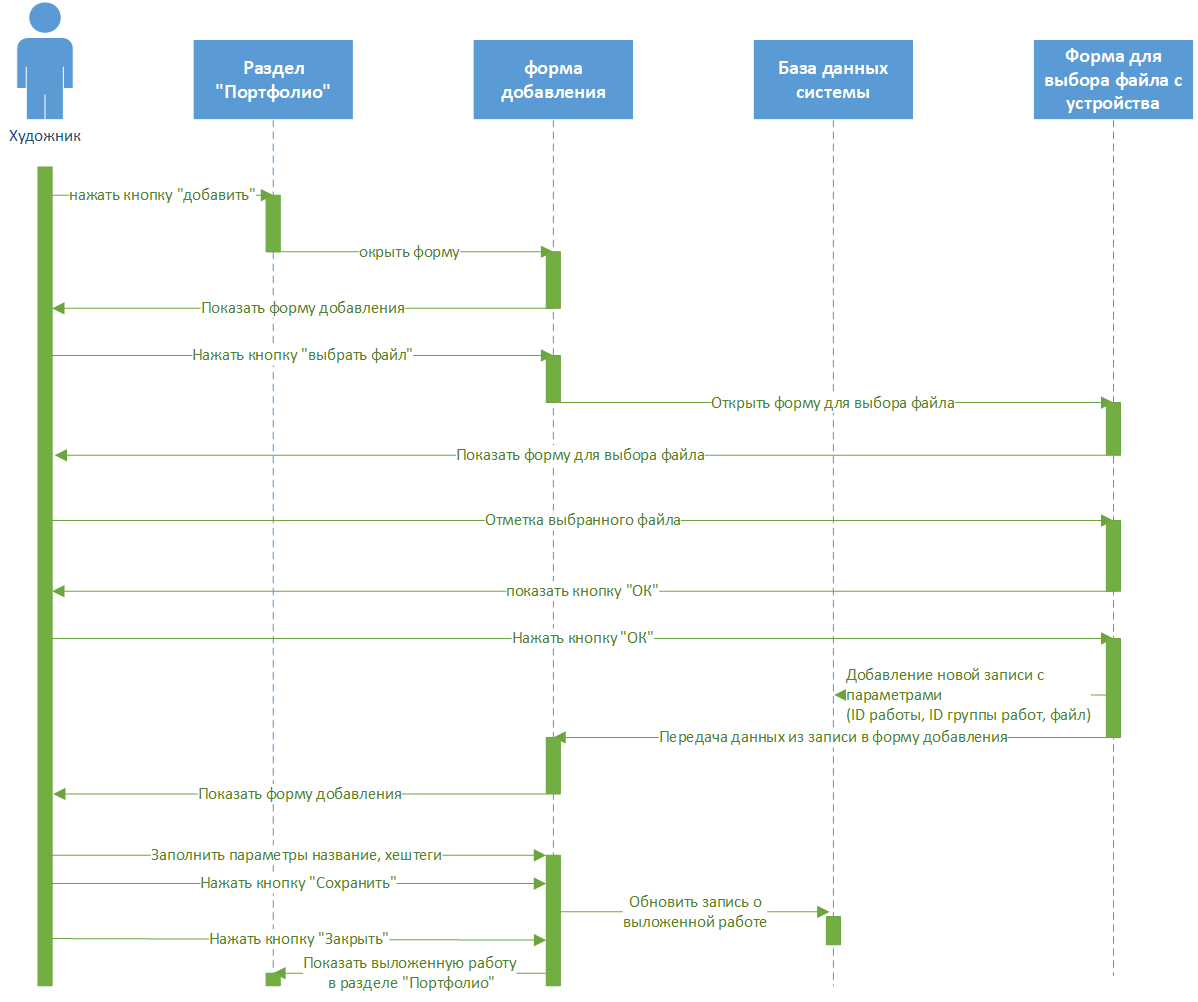
На рисунке 3 отображена последовательность действий при прохождении процесса «Добавление новой работы» со стороны художника.

Рис. 3. Добавление новой работы

Далее с помощью диаграммы последовательности описываем процесс «Поиск художника». В диаграмме используется блок *loop,* для отображения цикличности процессов создания записи о результате поиска и передаче результатов из БД на форму поиска. Также используется блок *alt,* который показывает несколько вариантов возможных способов для поиска автора (рисунок 4).

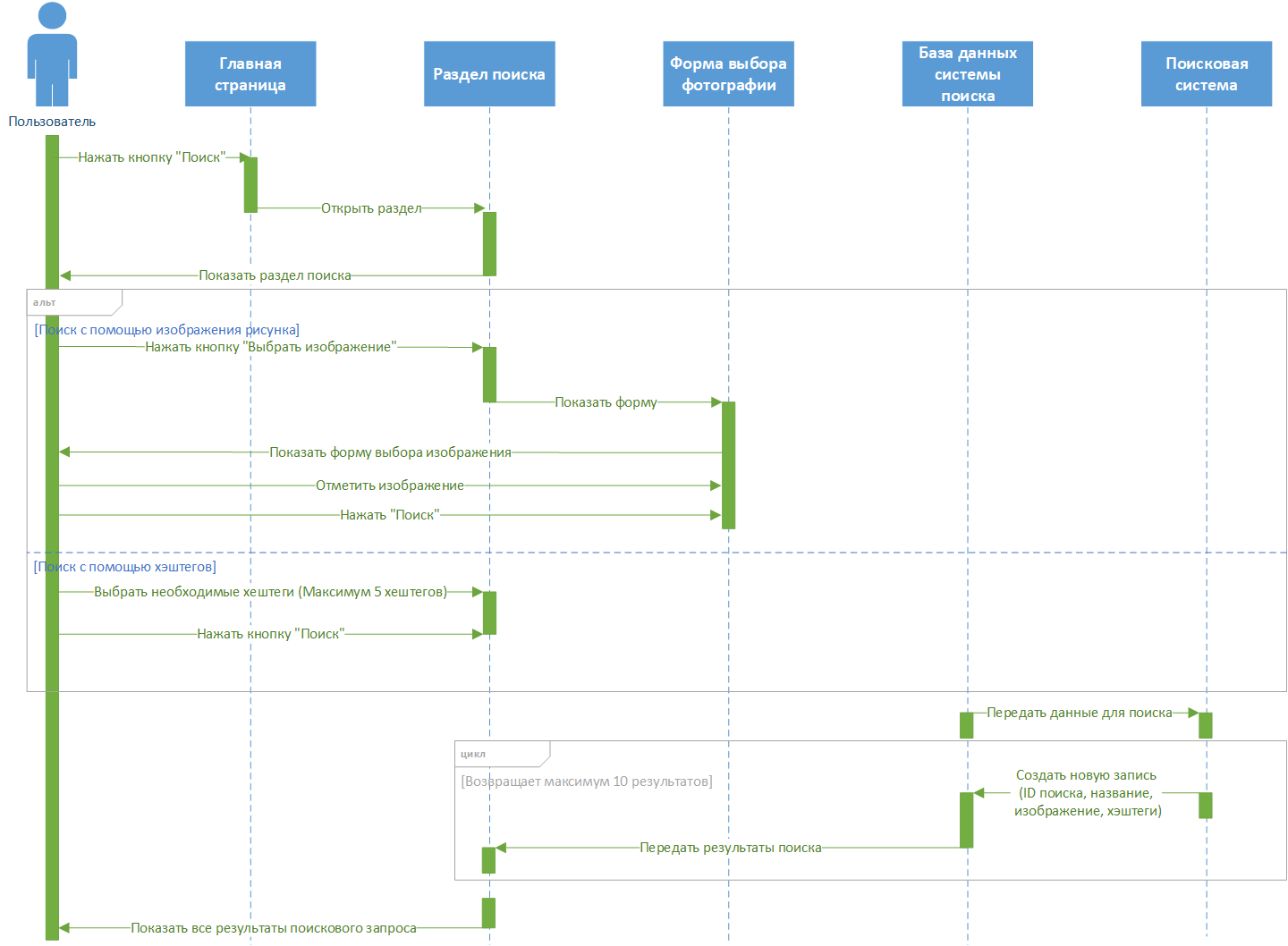


Рис. 4. Модель поиска автора

В случае добавления нового хештега используется обновляемая таблица. Для объяснения работы данной таблицы представлена диаграмма. В диаграмме используется блок *loop* для проверки существующих значений и добавления новых в случае отсутствия данного хештега (рисунок 5).

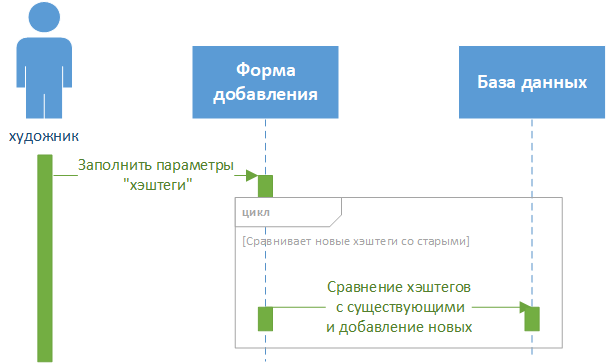


Рис. 5. Модель добавления новых хештегов

# Диаграмма классов

Диаграмма классов определяет типы классов системы и различного рода статические связи, которые существуют между ними. На диаграммах классов изображаются также атрибуты классов, операции классов и ограничения, которые накладываются на связи между классами [5]. На рисунке 6 изображена диаграмма классов информационной системы.

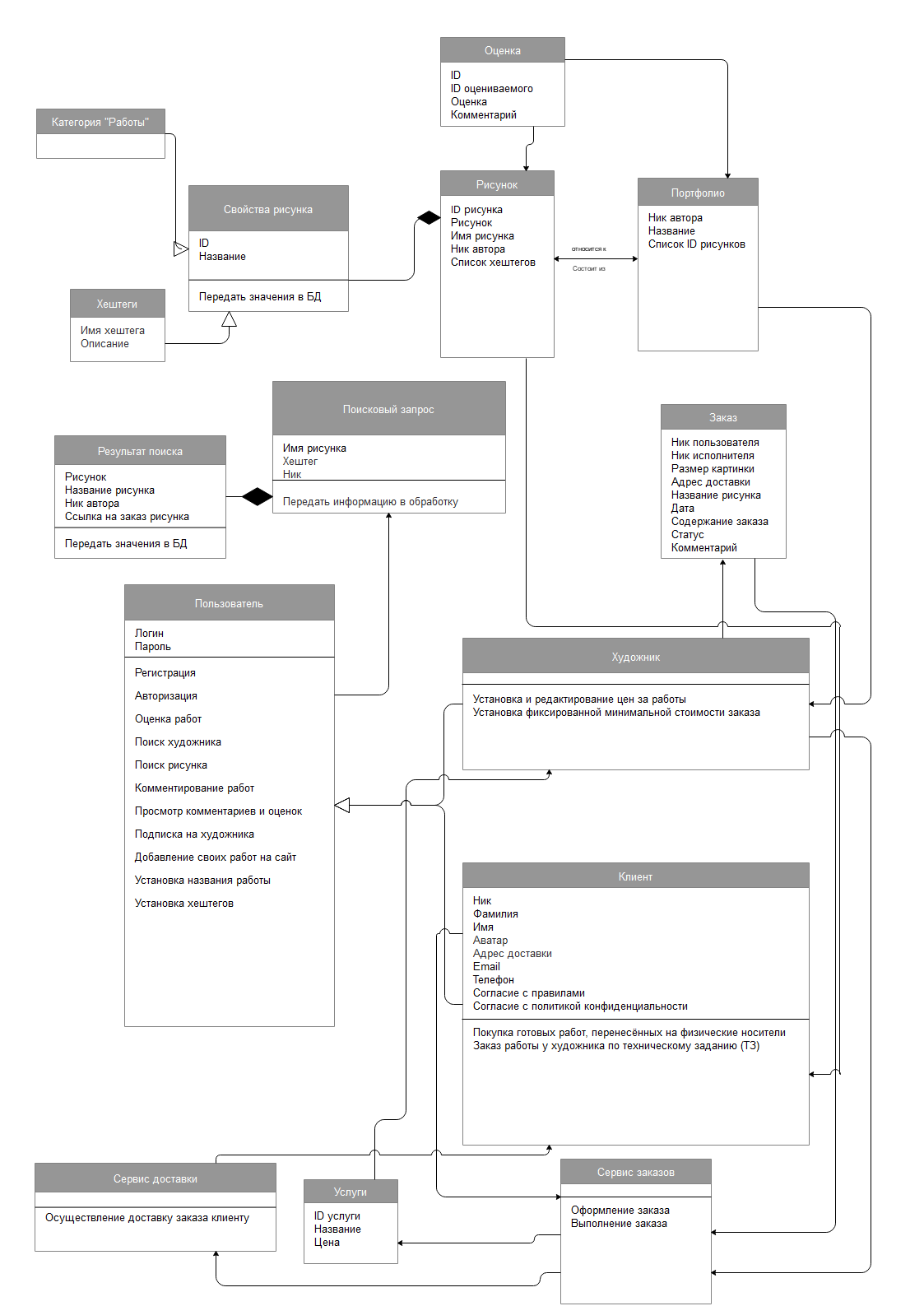


Рис. 6. Диаграмма классов ИС

# Модель данных

Модель данных – совокупность структуры информации и набора операций, определенных над элементами описания.

Для представления данных используют 2 уровня:

* Логический уровень – представляет информацию в формате удобном для пользователя и прикладных программ.
* Физический уровень – информация на конкретном носителе [8].

## **Логический уровень модели данных**

Вся информация, использующаяся в информационной системе, хранится в базе данных. Модель данных системы показана на рисунке 7.

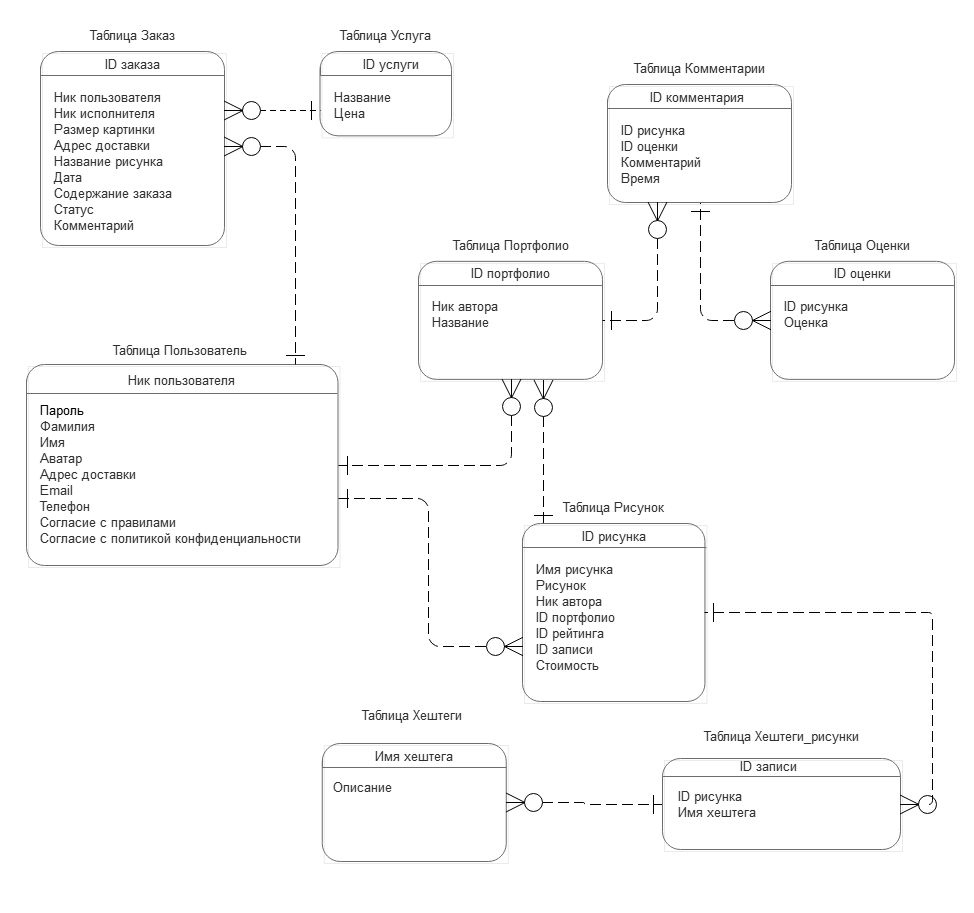


Рис. 7. Логический уровень одели данных

## **Физический уровень модели данных**

В качестве СУБД была выбрана *MySQL*, потому что она является открытым программным обеспечением.Все атрибуты реляционной модели указаны с соответствующими типами данных (Рисунок 8).

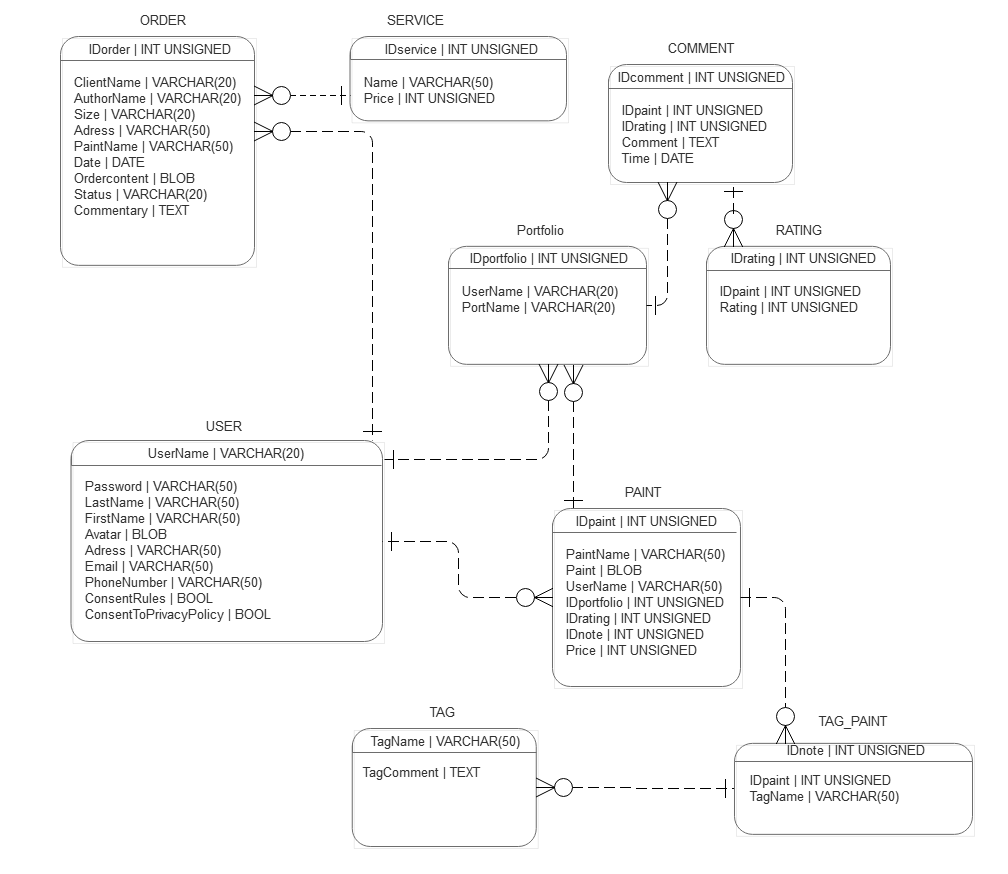


Рис. 8. Физическая уровень модели данных

Таблица «Пользователь» (таблица 3) предназначена для хранения информации обо всех пользователях системы.

Таблица 3 Пользователь

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Атрибут | Тип данных | Назначение | FK | PK |
| UserName | VARCHAR(20) | Логин для авторизации | ДА | ДА |
| Password | VARCHAR(50) | Пароль для авторизации |  |  |
| Lastname | VARCHAR(50) | Фамилия пользователя |  |  |
| Firstname | VARCHAR(50) | Имя пользователя |  |  |
| Avatar | BLOB | Аватар |  |  |
| Email | VARCHAR(50) | Электронная почта |  |  |
| PhoneNumber | VARCHAR(50) | Номер телефона |  |  |
| ConsentRules | BOOL | Согласие с правилами системы |  |  |
| ConsentToPrivacyPolicy | BOOL | Согласие с политикой конфиденциальности |  |  |

Таблица «Рисунок» (таблица 4) предназначена для хранения информации обо всех рисунках пользователей.

Таблица 4 Рисунок

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Атрибут | Тип данных | Назначение | FK | PK |
| IDpaint | INT UNSIGNED | Идентификатор записи | ДА | ДА |
| PaintName | VARCHAR(50) | Название рисунка |  |  |
| UserName | VARCHAR(20) | Ник автора | ДА |  |
| IDportfolio | INT UNSIGNED | Идентификатор портфолио | ДА |  |
| IDrating | INT UNSIGNED | Идентификатор оценки рисунка | ДА |  |
| IDnote | INT UNSIGNED | Идентификатор хештега к рисунку | ДА |  |

Таблица «Портфолио» (таблица 5) предназначена для хранения информации о рисунках конкретного пользователя.

Таблица 5 Портфолио

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Атрибут | Тип данных | Назначение | FK | PK |
| IDportfolio | INT UNSIGNED | Идентификатор записи | ДА | ДА |
| UserName | VARCHAR(20) | Ник автора | ДА |  |
| PortName | VARCHAR(20) | Название портфолио |  |  |

Таблица «Хештеги» (таблица 6) предназначена для хранения информации о хештегах.

Таблица 6 Хештеги

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Атрибут | Тип данных | Назначение | FK | PK |
| TagName | VARCHAR(50) | Имя хештега | ДА | ДА |
| TagComment | TEXT | Описание хештега |  |  |

Таблица «Хештеги\_Рисунки» (таблица 7) предназначена для хранения информации о привязанных к рисунку хештегах.

Таблица 7 Хештеги\_Рисунки

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Атрибут | Тип данных | Назначение | FK | PK |
| IDnote | VARCHAR(50) | Имя хештега | ДА | ДА |
| TagComment | TEXT | Описание хештега |  |  |

Таблица «Оценка» (таблица 7) предназначена для хранения информации об оценках рисунков.

Таблица 7 Оценка

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Атрибут | Тип данных | Назначение | FK | PK |
| IDrating | INT UNSIGNED | Идентификатор оценки | ДА | ДА |
| IDpaint | INT UNSIGNED | Идентификатор рисунка | ДА |  |
| Rating | INT UNSIGNED | Оценка рисунка по 5-ти бальной шкале |  |  |

Таблица «Комментарий» (таблица 8) предназначена для хранения информации о результатах поисковых запросов пользователей.

Таблица 8 Комментарий

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Атрибут | Тип данных | Назначение | FK | PK |
| IDcomment | INT UNSIGNED | Идентификатор записи |  | ДА |
| IDpaint | INT UNSIGNED | Идентификатор рисунка | ДА |  |
| IDrating | INT UNSIGNED | Идентификатор оценки | ДА |  |
| Comment | TEXT | Комментарий |  |  |
| Time | DATE | Время написания комментария |  |  |

Таблица «Заказ» (таблица 9) предназначена для хранения информации о заказах, совершенный пользователем.

Таблица 9 Заказ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Атрибут | Тип данных | Назначение | FK | PK |
| IDorder | INT UNSIGNED | Идентификатор записи | ДА | ДА |
| ClientName | VARCHAR(20) | Ник заказчика | ДА |  |
| AuthorName | VARCHAR(20) | Ник художника | ДА |  |
| Size | VARCHAR(20) | Размер рисунка |  |  |
| Adress | VARCHAR(50) | Адрес доставки | ДА |  |
| PaintName | VARCHAR(50) | Имя рисунка | ДА |  |
| Date | DATE | Дата заказа |  |  |
| Ordercontent | BLOB | Заказанный рисунок | ДА |  |
| Status | VARCHAR(20) | Статус заказа |  |  |
| Commentary | TEXT | Комментарий к заказу |  |  |

Таблица «Услуга» (таблица 10) предназначена для хранения информации о возможных услугах.

Таблица 10 Услуга

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Атрибут | Тип данных | Назначение | FK | PK |
| IDservice | INT UNSIGNED | Идентификатор записи | ДА | ДА |
| Name | VARCHAR(50) | Название услуги |  |  |
| Price | INT UNSIGNED | Цена услуги |  |  |

# Интерфейс информационной системы

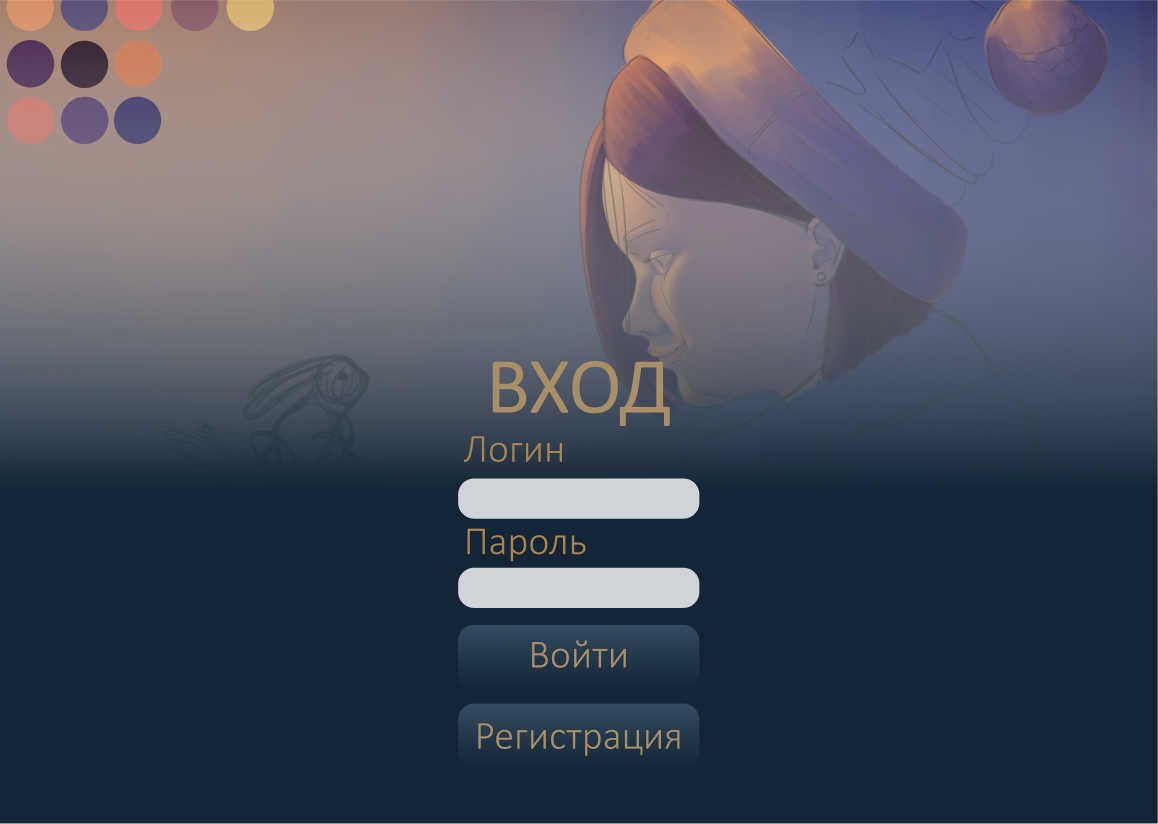
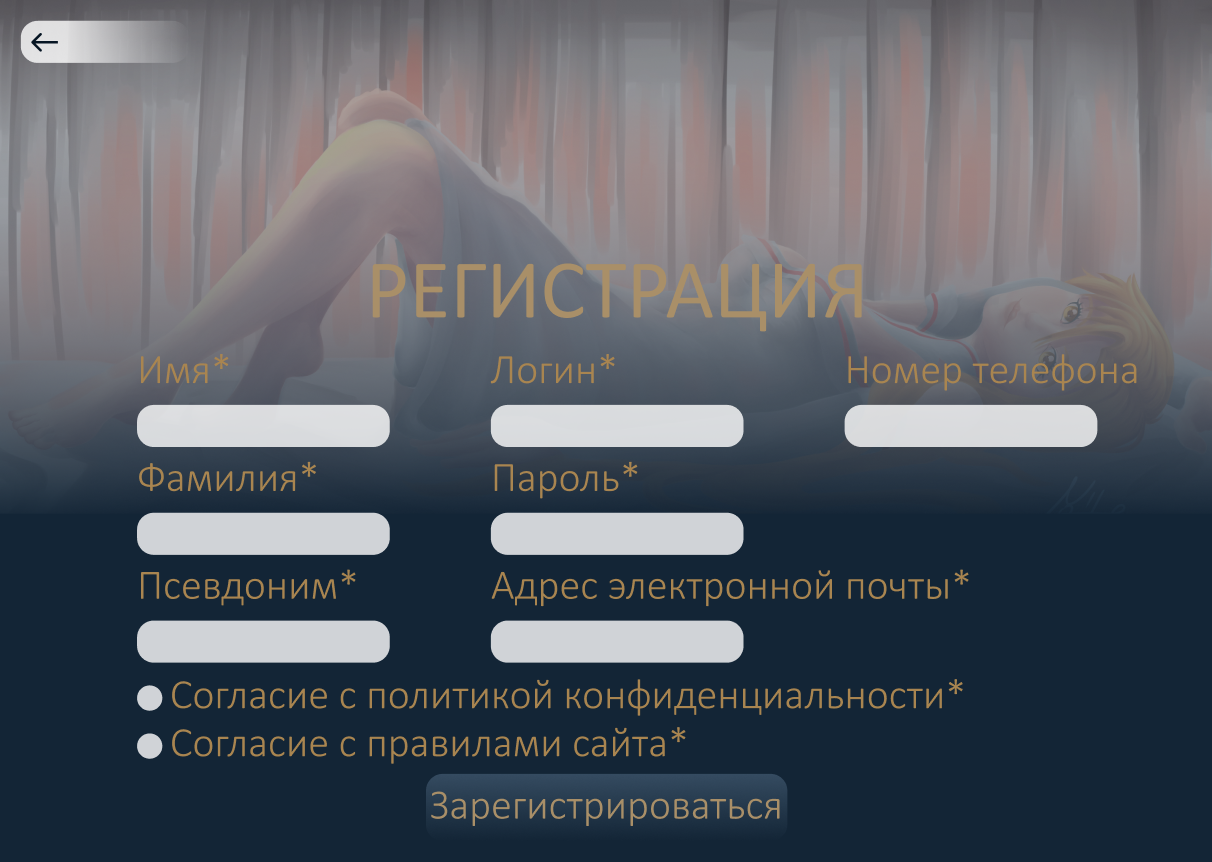
Прототип информационной системы для поиска художников для работы был спроектирован в среде *Figma*. После первого запуска приложения пользователь попадает на экран входа в систему (рисунок 9)

Рис. 9. Окно входа

Здесь пользователь должен ввести свои данные и нажать на кнопку «Войти», или, если он не зарегистрирован, то нажать кнопку «Регистрация». При нажатии на кнопку «Регистрация» происходит переход на форму регистрации, изображенную на рисунке 10. На форме «Регистрация» необходимо ввести свои имя и фамилию, псевдоним, который будет использоваться пользователем, логин и пароль, адрес электронной почты, а также в необязательном порядке номер телефона и номер карты.

Рис. 20. Форма регистрации



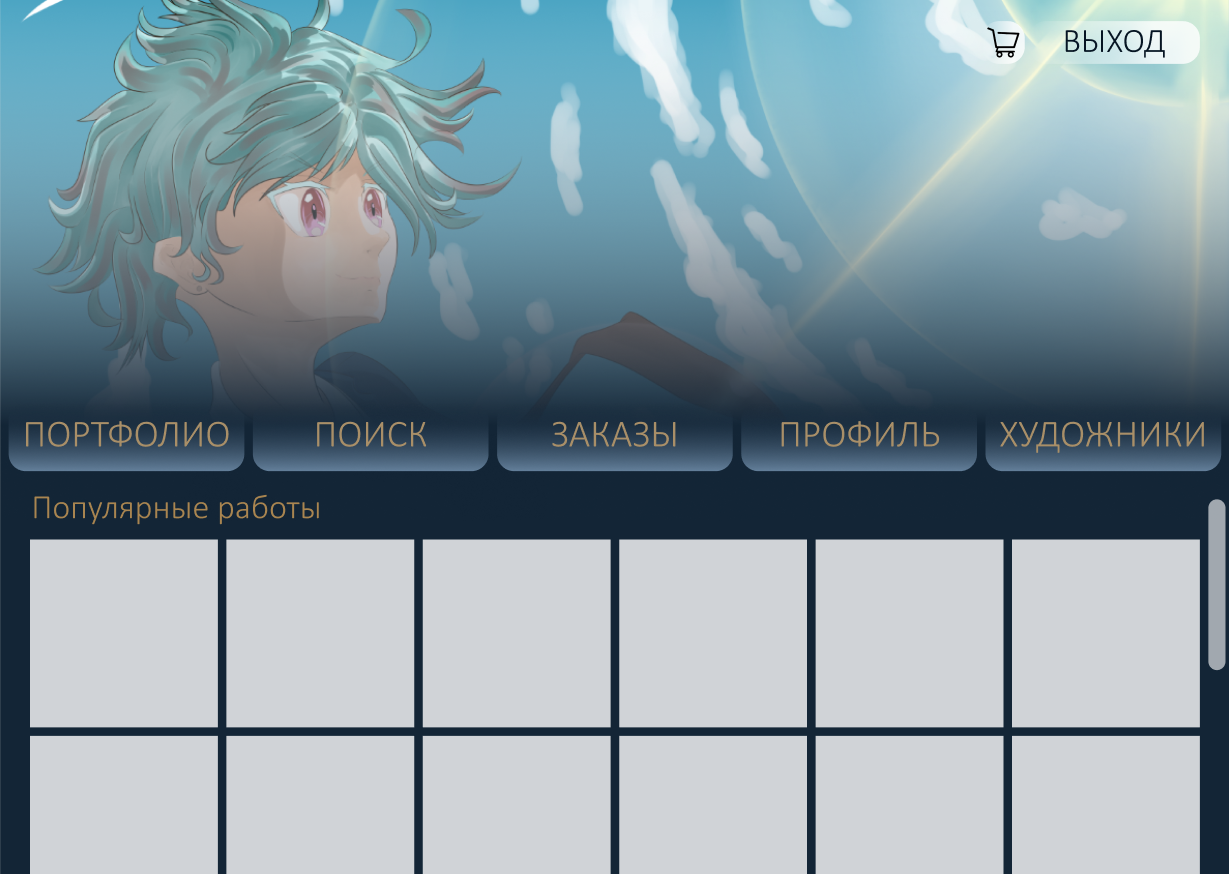
Далее, после нажатия кнопки «Зарегистрироваться», пользователь переходит на главную страницу сайта, изображённую на рисунке 11. На верхней панели под главным фоном расположены 5 кнопок. Каждая обозначает определенный раздел системы. Когда пользователь находится на определенной странице кнопка с ее названием выделятся цветом. Данная страница позволяет видеть последние популярные работы, выставленные на площадке.

Рис. 11. Главная страница площадки

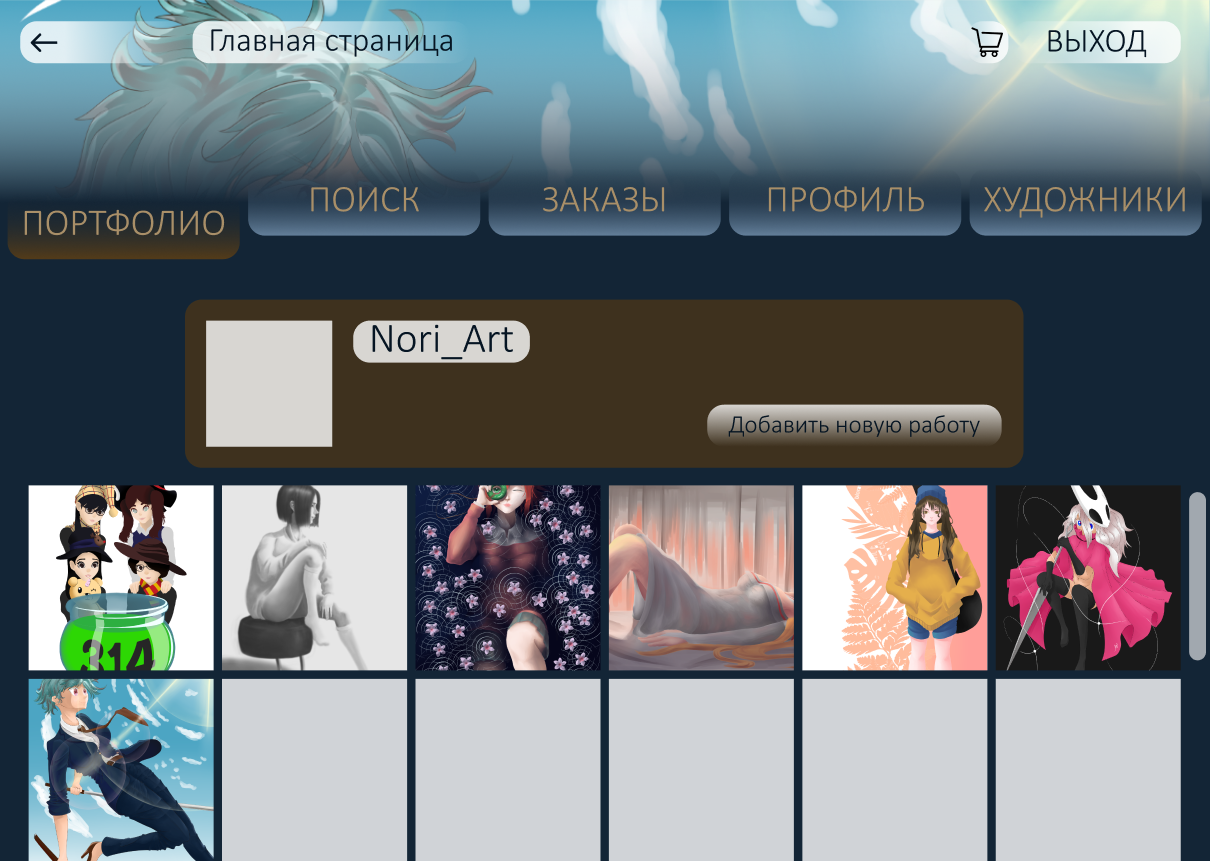
При нажатии кнопки «Портфолио» пользователь перейдёт на страницу, где расположены выставленные им работы (Рисунок 12). На данной странице он может добавить новые работы, а также, нажав на один из рисунков (Рисунок 13), посмотреть оценку и комментарии пользователей.

Рис. 12. Портфолио пользователя

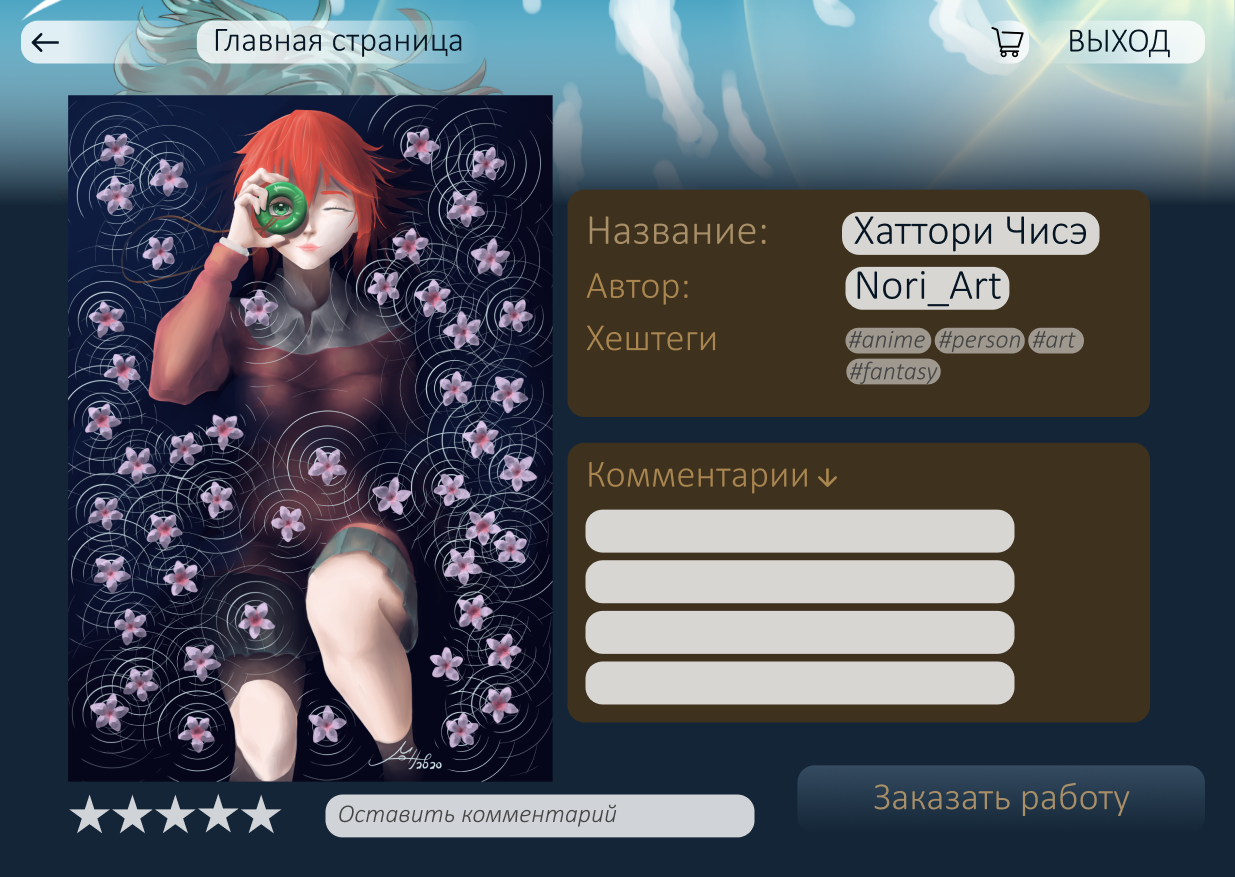


Рис. 13. Окно рисунка

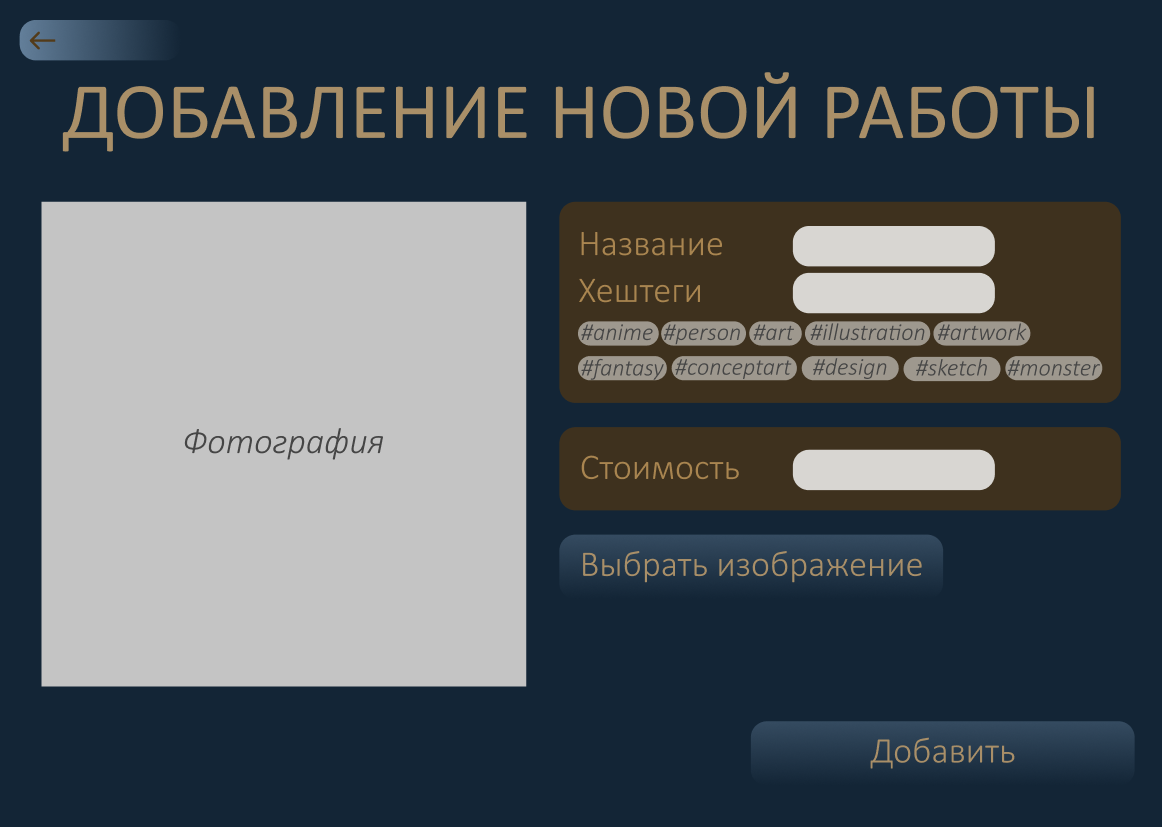
Если пользователь захочет добавить новую работу в портфолио, то после нажатия на соответствующую кнопку он перейдёт в форму добавления новой работы (рисунок 14).

Рис. 14. Окно добавления новой работы

В случае, когда пользователю площадки необходимо найти определённого художника, на главной панели расположена кнопка «Поиск». Данная кнопка перемещает пользователя на окно поиска (Рисунок 15). Поиск можно производить с помощью имени художника, названия или фотографии его работы, или, в случае, когда необходим определённый стиль или жанр работ, на площадке представлен поиск по хештегам. После ввода имеющихся данных, с правой стороны появятся результаты поиска, в которых есть возможность выбора среди работ или среди художников.

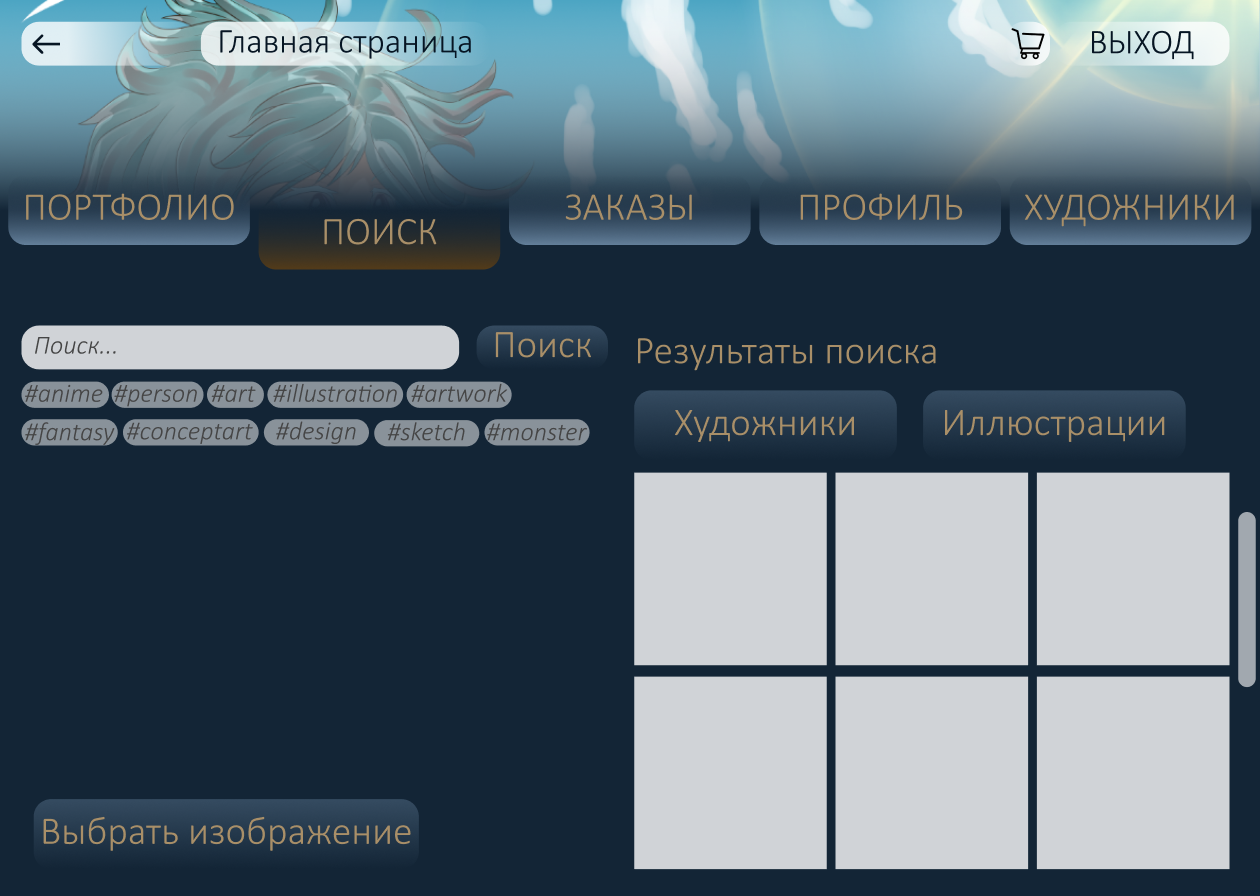


Рис. 15. Окно поиска

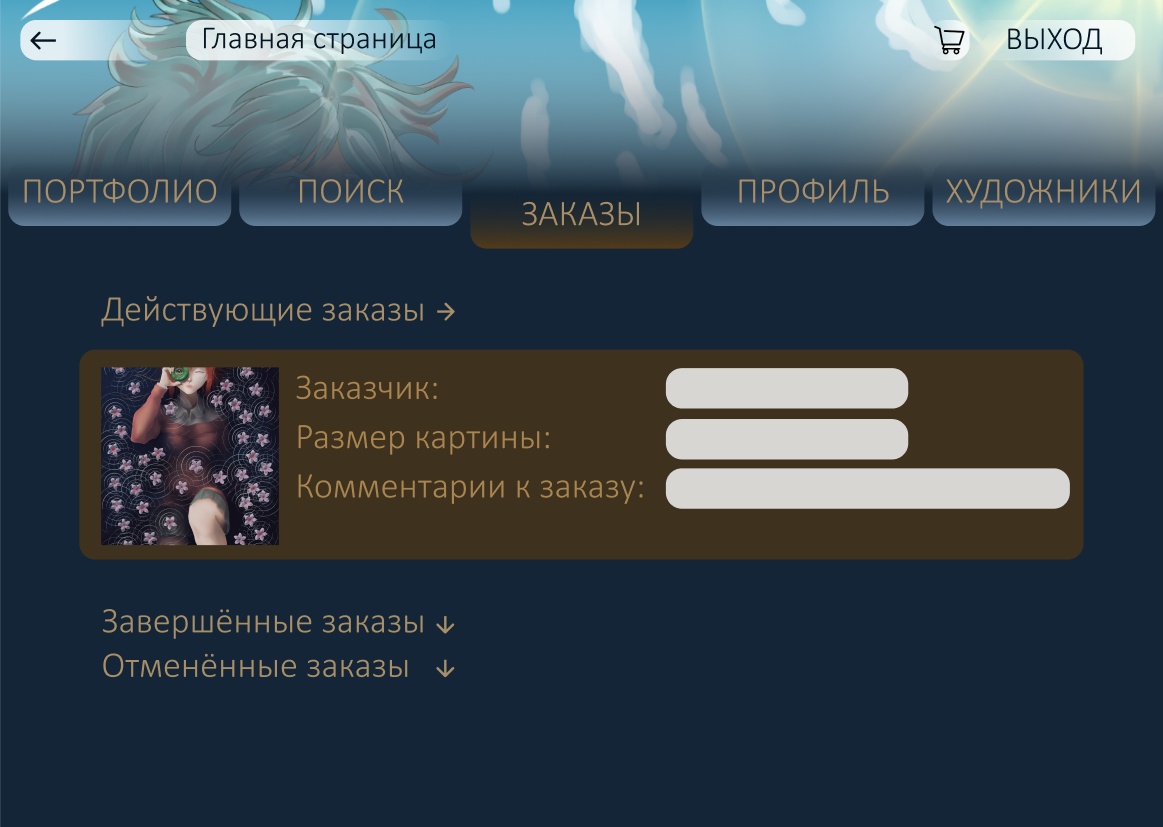
Кнопка «Заказы» переносит пользователя в окно заказов (Рисунок 16). Здесь художник может просмотреть действующие, завершённые и отменённые заказы, сделанные пользователями.

Рис. 16. Окно просмотра заказов

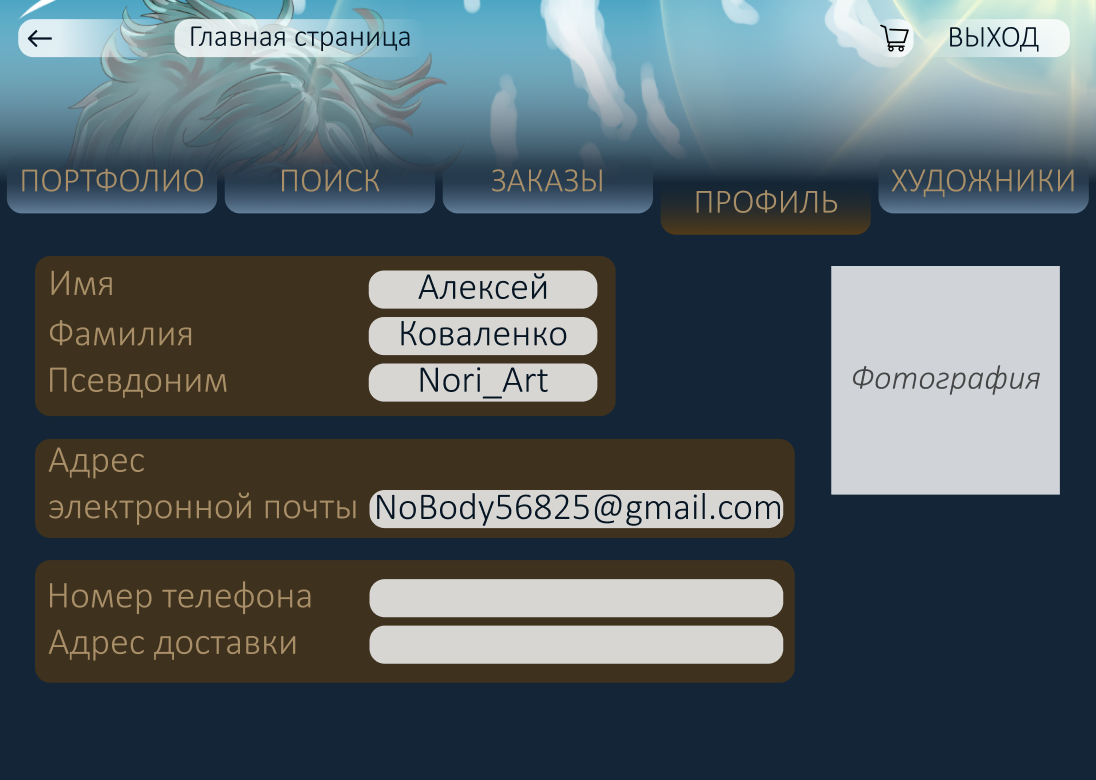
На рисунке 17 изображено окно профиля, на которое пользователь переходит после нажатия на кнопку «Профиль». На этой странице он может изменить некоторые данные о себе. 

Рис. 17. Окно профиля пользователя

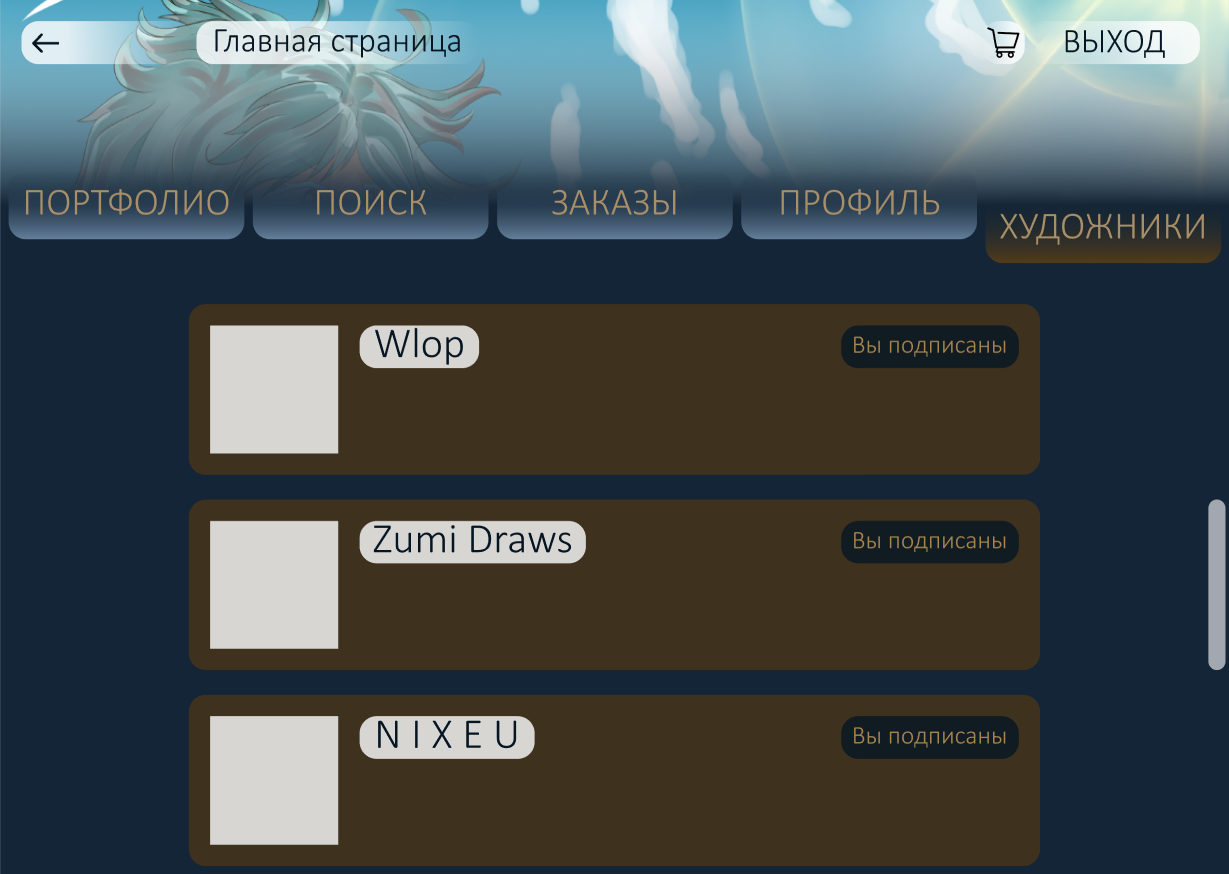
Пользователь также может просмотреть художников, на которых он подписан. При нажатии на кнопку «Художники» его автоматически переводит на соответствующую страницу, изображённую на рисунке 18.

Рис. 18. Окно, где расположены художники, на которых подписан пользователь

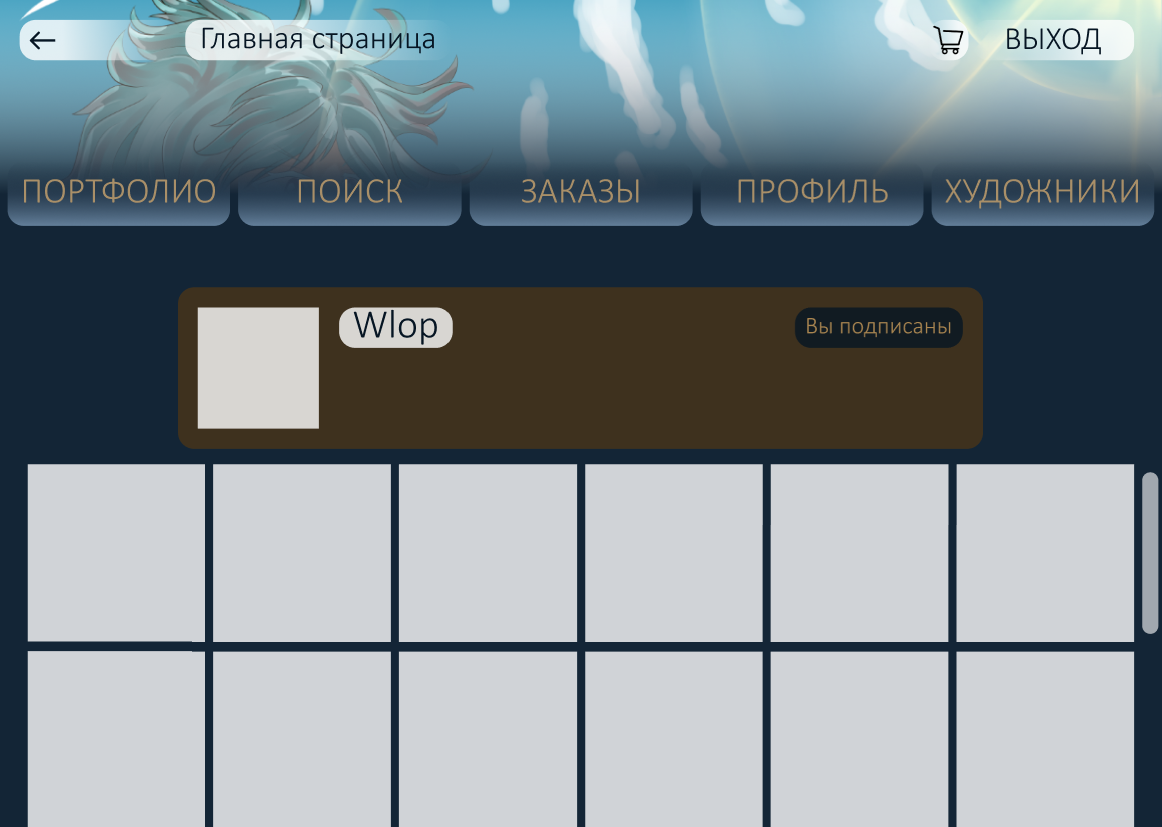
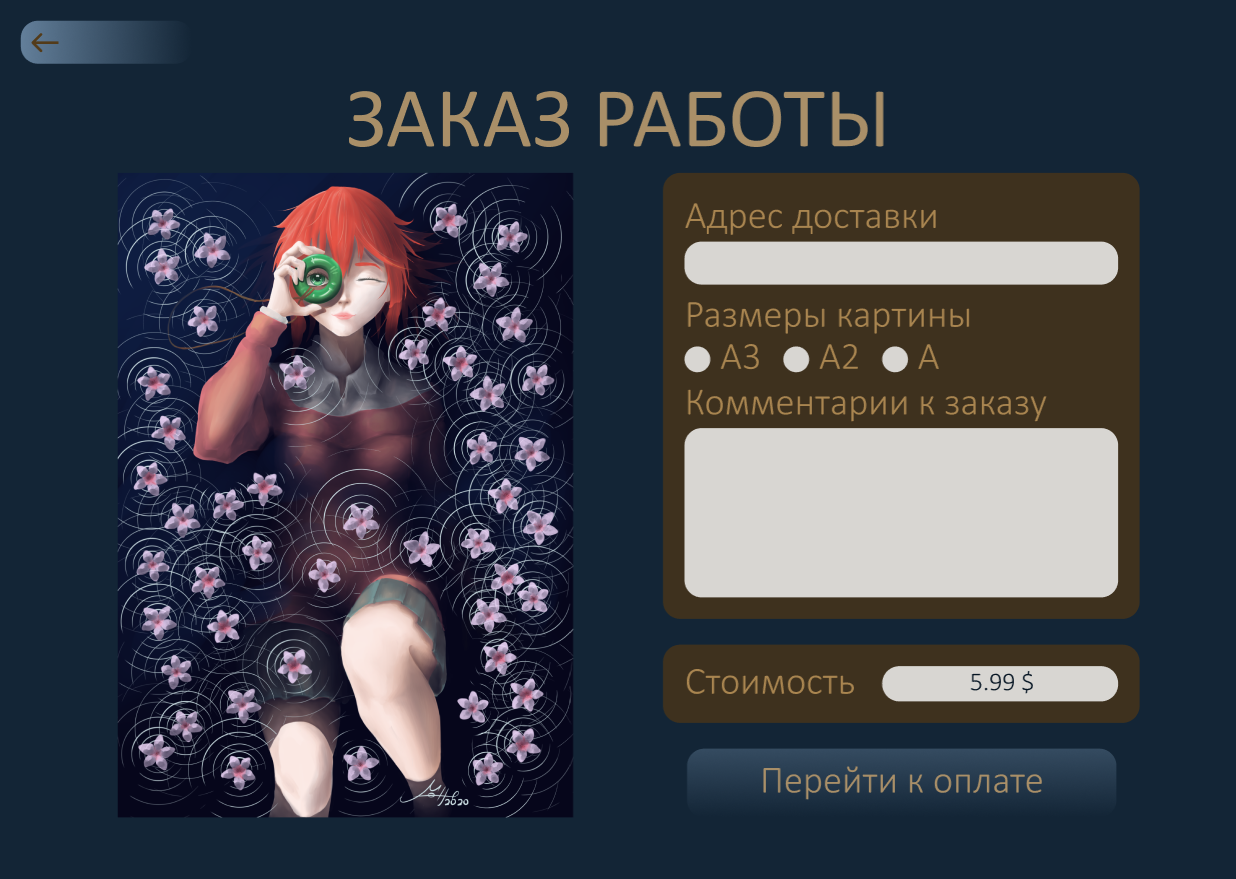
Нажав, на фотографию или имя художника, пользователь переходит в его портфолио (Рисунок 19), где он может просмотреть работы художника, а также, при желании заказать себе физическую копию работы (Рисунок 20).

Рис. 19. Окно портфолио художника

В данной форме можно ввести адрес доставки (если он не указан в профиле), выбрать размеры работы, а после оплатить её.

Рис. 20. Форма заказа работы



# Заключение

В результате курсовой работы была спроектирована информационная система, предназначенная для художников, пользователь и работодателей. Основой для создания информационной системы послужили проблемы предметной области. Были разработаны следующие диаграммы и модели: диаграммы прецедентов, диаграммы деятельности, диаграммы последовательностей, диаграмма классов, логическая и физическая модели данных. В качестве среды проектирования было использовано программное средство *Microsoft Visio*, предназначенное для автоматизации этапов анализа и проектирования предметной области. Средством реализации прототипа интерфейса проектируемой системы стала *Figma*.

Использование данной ИС позволит улучшить процесс оказания услуг художника, скорость и удобство его взаимодействия с клиентами. Благодаря системе перед деятелями искусства откроется более лёгкий путь для поиска своей аудитории и работодателей, что упростит поиск обеим сторонам. Доход увеличится за счет повышения количества клиентов и цен на услуги, также система будет способствовать популяризации услуг клиента.

# Список литературы

1. Савватеева, Т. П. Технологии проектирования информационных систем: учеб. Пособие / Т. П. Савватеева, И. А. Соколов. – Дубна: Гос. ун-т «Дубна», 2020. — 59 с.
2. Учебное пособие по диаграмме прецедентов: учебное пособие [Электронный ресурс] — Режим доступа: <https://creately.com/blog/ru/> (дата обращения: 1.03.2022).
3. Давиденко С. Диаграмма деятельности UML: статья [Электронный ресурс] — Режим доступа: https://planerka.info/item/diagrammy-deyatelnosti-uml/ (дата обращения: 3.03.2022).
4. Основы UML. Диаграммы последовательности: статья [Электронный ресурс] — Режим доступа: https://pro-prof.com/archives/2769 (дата обращения: 8.03.2022).
5. Построение диаграммы классов: статья [Электронный ресурс] — Режим доступа: https://flexberry.github.io/ru/gpg\_class-diagram.html (дата обращения: 8.03.2022).
6. Уровни представления информации, понятие модели данных: статья [Электронный ресурс] — Режим доступа: https://sites.google.com/site/gosyvmkss12/bazy-dannyh/02-urovni-predstavlenia-informacii-ponatie-modeli-dannyh (дата обращения: 5.03.2022).