

Министерство науки и высшего образования Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«Национальный исследовательский университет ИТМО»
(Университет ИТМО)

Факультет Прикладной информатики

Образовательная программа

О Т Ч Е Т

об учебно-ознакомительной практике

Тема задания: Разработка алгоритмов работы веб-платформы агрегатора портфолио и их визуализация с помощью блок-схем

Обучающийся: Зенин Данил Дмитриевич, К3220

Согласовано:

Руководитель практики от университета: Казанов Андрей Робертович,
практикант факультета прикладной информатики

Практика пройдена с оценкой _____
Дата 15.02.2025

Санкт-Петербург
2025

СОДЕРЖАНИЕ

Стр.

ВВЕДЕНИЕ	4
1 Анализ существующих аналогов.....	5
1.1 LinkedIn	5
1.2 Сетка	6
1.3 Taplink.....	6
1.4 GitHub.....	7
1.5 Вывод по аналогам.....	7
2 Формирование требований к системе	8
2.1 Описание системы.....	8
2.2 Необходимые к реализации функции	8
2.3 Нефункциональные требования	9
2.4 Выбор технологий разработки.....	9
3 Создание диаграммы вариантов использования.....	10
3.1 Определение типов пользователей	10
3.2 Создание диаграммы вариантов использования.....	10
4 Создание диаграммы классов	12
4.1 Класс User	12
4.2 Класс Moderator.....	13
4.3 Класс Admin	14
4.4 Класс Portfolio.....	14
4.5 Класс Project	15
4.6 Класс Education	16
4.7 Класс Tag	16
4.8 Класс Like.....	17
4.9 Класс Comment.....	17
4.10 Класс PortfolioSearch.....	18
4.11 Класс AuthManager	18
4.12 Класс RoleManager.....	19
4.13 Диаграмма классов	19
5 Создание блок-схем	21
5.1 Определение основополагающих алгоритмов системы.....	21

5.2	Авторизация пользователя	21
5.3	Добавление проекта в портфолио	22
5.4	Поиск портфолио	23
5.5	Блокировка портфолио модератором	24
5.6	Назначение пользователя модератором.....	25
5.7	Лайк портфолио.....	25
5.8	Комментарий портфолио	26
ЗАКЛЮЧЕНИЕ		27
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....		28

ВВЕДЕНИЕ

В современном профессиональном пространстве очень важно показать себя достойным специалистом, и портфолио является важным инструментом для демонстрации как навыков, так и достижений специалиста в различных сферах. Веб-платформы позволяют пользователям удобно публиковать свои работы, находить клиентов и взаимодействовать с потенциальными работодателями. Однако современные площадки для публикации проектов, как правило, направлены на какую-то одну область, будь то код или дизайн, и иногда их функционала может не хватать.

Далее описаны цели и задачи, которые были выдвинуты для выполнения задания.

Цель: Создание базисной основы для разработки клиент-серверного веб-приложения Portfolio Hub для создания, редактирования и поиска портфолио.

Для реализации цели нужно выполнить следующие задачи:

- Сформировать требования к системе.
- Выбрать технологии разработки.
- Определение основных функций приложения.
- Создание диаграмм вариантов использования.
- Выделение основных классов приложения.
- Создание диаграмм классов.
- Создание блок-схем.

1 Анализ существующих аналогов

При анализе и проектировании системы важным шагом является анализ существующих аналогов. Он может помочь найти преимущества перед другими веб-сервисами и провести редактирование требований к системе с учетом функций аналогов. В качестве наиболее подходящих существующих аналогов системы были выбраны: LinkedIn, Сетка, Taplink и GitHub, потому что это самые известные аналоги, которые имеют схожий с описываемой системой функционал. Ниже описывается каждый из аналогов следующим образом: вначале дается краткое описание системы, далее описывается ее функционал, после чего его функционал сравнивается с описываемой системой и делается вывод о схожести и различиях между двумя системами.

1.1 LinkedIn

LinkedIn [9] — это социальная сеть, ориентированная на профессионалов, которая позволяет пользователям устанавливать деловые контакты, находить работу, делиться опытом и искать сотрудников. Этот веб-сервис особенно распространен за рубежом. У него более миллиарда активных пользователей в месяц.

На сервисе можно размещать резюме, настраивать личный профиль и искать работу. Также можно отслеживать информацию, которой делится пользователь и просматривать резюме других пользователей.

Кроме этого, можно общаться с другими людьми в личных сообщениях и групповых чатах.

Этой системой успешно пользуются и компании для подбора персонала по резюме и написанной в ней информации. Для этого существуют множественные системы фильтрации и рекомендаций.

Вывод: LinkedIn – это самая крупная социальная сеть для работы в мире. У нее огромный функционал, с которым наш веб-сервис никогда не сможет сравняться, но в разрабатываемой системе делается акцент именно на портфолио, а не на резюме. Наша система может быть отличным дополнением к резюме в LinkedIn.

1.2 Сетка

Сетка [3] - это российская социальная сеть от одного из главных сайтов по подбору персонала hh.ru. Во многом она схожа с LinkedIn. На ней также можно вести канал с публикациями для авторов. Кроме того, можно редактировать профиль пользователя.

Плюсом Сетки по сравнению с LinkedIn является то, что Сетка предлагает пользователям по составленному в профиле резюме входить в чаты с пользователями с похожими показателями. Это позволит сильно упростить нетворкинг, что является важной частью работы любого специалиста.

Вывод: Сетка – это очень похожая на LinkedIn система, о которой можно сказать то же самое, что и о LinkedIn. Наш веб-сервис отличается от этой системы тем, что наша система специализируется на составлении портфолио, а не резюме.

1.3 Taplink

Taplink [4] – это веб-приложение, которое может помочь создать сайт лендинг с личной информацией. Туда можно добавить информацию о себе и ссылки на все социальные сети.

Это очень удобное средство для создания электронной визитки. Профиль созданный с Taplink может быть очень легко интегрирован с социальными сетями. Профиль можно сделать полностью кастомизированным и со своим дизайном. Также предлагается множество шаблонов.

Вывод: Taplink – это удобная платформа для создания кастомизированного профиля. Taplink является частичным аналогом описываемой системы, так как в ней можно составить профиль о своей информации и даже описать проекты, которыми ты занимался. Но наш проект не является аналогом Taplink, он будет хорошим дополнением этой

системы, так как ссылка на все портфолио отлично подойдет по структуре профиля Taplink.

1.4 GitHub

GitHub [5] – это сервис для хранения репозиторий и взаимодействия с ними. Он работает с библиотекой git, которая нужна для управления версиями кода. Над каждым проектом может работать несколько человек. На гитхаб работает объемная система ролей. В каждом проекте может лежать подробное описание с различными вставками, такие как картинки, код и многое другое.

Но GitHub специализируется именно на разработчиках и имеет не очень удобную систему просмотра общего списка проектов, а нужен скорее для подробного рассмотрения одного проекта.

Вывод: GitHub – это великолепная система для разработчиков, но она в меньшей степени подходит для создания портфолио, на чем наша система будет специализироваться.

1.5 Вывод по аналогам

Все приведенные выше аналоги не имеют полный функционал описываемой системы или специализируются на другой сфере. В первых трех случаях наша система станет только отличным дополнением к системам-аналогам, а последний вариант выполняет немного другие функции. Таким образом, идея, стоящая за этой системой, может найти свою нишу и быть используемой.

2 Формирование требований к системе

2.1 Описание системы

Веб-платформа агрегатора должна предоставлять возможность пользователям создавать профили, в которых выставлять свои портфолио с описанием себя как работника, своих проектов, образования и иметь возможности для расширения функционала. Кроме того, в системе должна быть реализована система поиска других профилей пользователей, чтобы была возможность находить друг друга.

2.2 Необходимые к реализации функции

Итак, на платформе должно быть можно:

- Создавать профиль.
- Добавлять проект.
- Удалять проект.
- Искать портфолио других пользователей.
- Просматривать портфолио других пользователей.
- Оставлять лайки на портфолио других пользователей.
- Оставлять комментарии под портфолио других пользователей.
- Удалять портфолио.
- Редактировать описание о себе.
- Управлять тегами.
- Блокировать портфолио других пользователей.
- Просматривать статистику платформы.
- Назначить администратора.
- Управлять ролями.
- Изменять системные настройки.

Все эти функции будут разделены разным типами пользователей и представлены в виде диаграммы UML далее в отчёте.

2.3 Нефункциональные требования

Для системы предоставлены несколько нефункциональных требований, необходимых для масштабирования системы и ее продвижения на рынке:

- Высокая производительность (до 100 000 пользователей).
- Защита данных (шифрование паролей, защита API).
- Гибкость и масштабируемость.
- SEO-оптимизация.
- Интуитивно понятный UI/UX.

2.4 Выбор технологий разработки

Разработка серверной части (backend) приложения будет осуществляться на Java + Spring Boot.

Разработка внешнего интерфейса (frontend) будет осуществляться на React.js.

3 Создание диаграммы вариантов использования

В данной главе будет разработана диаграмма вариантов использования для клиент-серверного приложения Portfolio Hub. Будут определены типы пользователей, их функции.

3.1 Определение типов пользователей

После анализа требований к проекту и необходимых к реализации функций, было выявлено три типа пользователей, определенных в системе:

- Обычный пользователь.
- Модератор.
- Администратор.

Каждый из этих пользователей будет выполнять свои функции. Кроме того, типы пользователей иерархически связаны, то есть пользователь и максимальным уровнем доступа(администратор) будет иметь все права модератора и обычного пользователя, модератор будет иметь все права обычного пользователя, но не администратора, а обычный пользователь находится на нижнем уровне иерархии.

3.2 Создание диаграммы вариантов использования

На основе вышеописанных функций и типах пользователей, была составлена диаграмма вариантов использования (UML). Функции, допустимые для каждого типа пользователя, были определены на ней (рисунок 3.1).



Рисунок 3.1 — Диаграмма UML – функции для пользователей. Выполнена с помощью [1].

Создание диаграммы вариантов использования поможет правильно определить роли пользователям и функции, необходимые для реализации.

4 Создание диаграммы классов

Система будет реализована, используя концепции объектно-ориентированного программирования. Для системы были выделены классы, представленные ниже.

4.1 Класс User

Класс User – базовый класс для всех пользователей, который хранит личные данные.

Атрибуты класса представлены в таблице 4.1.

Таблица 4.1 — Атрибуты класса User

Имя	Описание	Тип
id	Уникальный идентификатор	UUID
username	Логин пользователя	String
passwordHash	Хэш пароля	String
email	Электронная почта	String
role	Роль (user/moderator/admin)	Enum

Операции класса представлены в таблице 4.2.

Таблица 4.2 — Операции класса User

Имя	Описание	Сигнатура
createPortfolio	Создать портфолио	createPortfolio(title: String)
editProfile	Редактировать профиль	editProfile(data: JSON)
searchPortfolios	Поиск портфолио	searchPortfolios(query: String)
likePortfolio	Поставить лайк	likePortfolio(portfolioId: UUID)
addComment	Добавить комментарий	addComment(portfolioId: UUID, text: String)

4.2 Класс Moderator

Класс Moderator – класс, наследующий User. Добавляет функции модерации. Своих атрибутов не имеет, так как наследуется от базового класса.

Операции класса представлены в таблице 4.3.

Таблица 4.3 — Операции класса Moderator

Имя	Описание	Сигнатура
blockPortfolio	Заблокировать портфолио	blockPortfolio(portfolioId: UUID)
unblockPortfolio	Разблокировать портфолио	unblockPortfolio(portfolioId: UUID)
editTag	Изменить тэг пользователя	editTag(userId: UUID, tag: String)

4.3 Класс Admin

Класс Admin – класс, наследующий User. Добавляет функции управления системой. Своих атрибутов не имеет, так как наследуется от базового класса.

Операции класса представлены в таблице 4.4.

Таблица 4.4 — Операции класса Admin

Имя	Описание	Сигнатура
assignModerator	Назначить модератора	assignModerator(userId: UUID)
updateSystemSettings	Изменить настройки системы	updateSystemSettings(settings: JSON)

4.4 Класс Portfolio

Класс Portfolio – класс, представляющий портфолио пользователя. Содержит информацию о проектах, образовании и другую информацию.

Атрибуты класса представлены в таблице 4.5.

Таблица 4.5 — Атрибуты класса Portfolio

Имя	Описание	Тип
portfolioId	Уникальный идентификатор	UUID
userId	ID владельца портфолио	UUID
projects	Список проектов	List<Project>
education	Список образовательных записей	List<Education>
bio	Описание пользователя	String
isBlocked	Флаг блокировки	Boolean
tags	Список тегов	List<Tag>

Операции класса представлены в таблице 4.6.

Таблица 4.6 — Операции класса Portfolio

Имя	Описание	Сигнатура
addProject	Добавить проект	addProject(project: Project)
removeProject	Удалить проект	removeProject(projectId: UUID)
updateBio	Обновить описание	updateBio(newBio: String)
addTag	Добавить тег	addTag(tagId: UUID)

4.5 Класс Project

Класс Project – класс, содержащий описание проекта в портфолио (название, описание, даты).

Атрибуты класса представлены в таблице 4.7.

Таблица 4.7 — Атрибуты класса Project

Имя	Описание	Тип
projectId	Уникальный идентификатор	UUID
title	Название проекта	String
description	Описание проекта	String
startDate	Дата начала	Date
endDate	Дата окончания	Date

4.6 Класс Education

Класс Education – класс, содержащий информацию об образовании пользователя.

Атрибуты класса представлены в таблице 4.8.

Таблица 4.8 — Атрибуты класса Education

Имя	Описание	Тип
institution	Учебное заведение	String
degree	Степень	String
startYear	Год начала	Int
endYear	Год окончания	Int

4.7 Класс Tag

Класс Tag – класс, определяющий тэг для категоризации портфолио (например, "UI/UX designer" или "front-end developer").

Атрибуты класса представлены в таблице 4.9.

Таблица 4.9 — Атрибуты класса Tag

Имя	Описание	Тип
tagId	Уникальный идентификатор	UUID
name	Название тега	String
description	Описание тега	String

4.8 Класс Like

Класс Like – класс, представляющий собой лайк, поставленный на портфолио.

Атрибуты класса представлены в таблице 4.10.

Таблица 4.10 — Атрибуты класса Like

Имя	Описание	Тип
likeId	Уникальный идентификатор	UUID
portfolioId	ID портфолио	UUID
userId	ID пользователя	UUID
timestamp	Время создания	DateTime

4.9 Класс Comment

Класс Comment – класс, представляющий собой комментарий, оставленный под портфолио.

Атрибуты класса представлены в таблице 4.11.

Таблица 4.11 — Атрибуты класса Comment

Имя	Описание	Тип
commentId	Уникальный идентификатор	UUID
portfolioId	ID портфолио	UUID
userId	ID пользователя	UUID
text	Текст комментария	String
timestamp	Время создания	DateTime

4.10 Класс PortfolioSearch

Класс PortfolioSearch — класс, являющийся сервисом для поиска портфолио по критериям.

Операции класса представлены в таблице 4.12.

Таблица 4.12 — Операции класса PortfolioSearch

Имя	Описание	Сигнатура
searchByTags	Поиск по тегам	searchByTags(tags: List<Tag>)
searchByName	Поиск по имени	searchByName(name: String)
searchBySkills	Поиск по навыкам	searchBySkills(skills: List<String>)

4.11 Класс AuthManager

Класс AuthManager — класс, предоставляющий управление авторизацией и аутентификацией.

Операции класса представлены в таблице 4.13.

Таблица 4.13 — Операции класса AuthManager

Имя	Описание	Сигнатура
login	Аутентификация	login(username: String, password: String)
logout	Выход из системы	logout()
validateToken	Проверка токена	validateToken(token: String)

4.12 Класс RoleManager

Класс RoleManager – класс, предоставляющий управление ролями пользователей.

Операции класса представлены в таблице 4.14.

Таблица 4.14 — Операции класса RoleManager

Имя	Описание	Сигнатура
promoteToModerator	Назначить модератора	promoteToModerator(userId: UUID)
demoteFromModerator	Снять роль модератора	demoteFromModerator(userId: UUID)
getUserRoles	Получить роли пользователя	getUserRoles(userId: UUID)

4.13 Диаграмма классов

По определенным классам была создана диаграмма классов, показывающая связи между классами, их атрибуты, типы атрибутов и методы.

Диаграмма классов представлена на Рисунке 4.1.

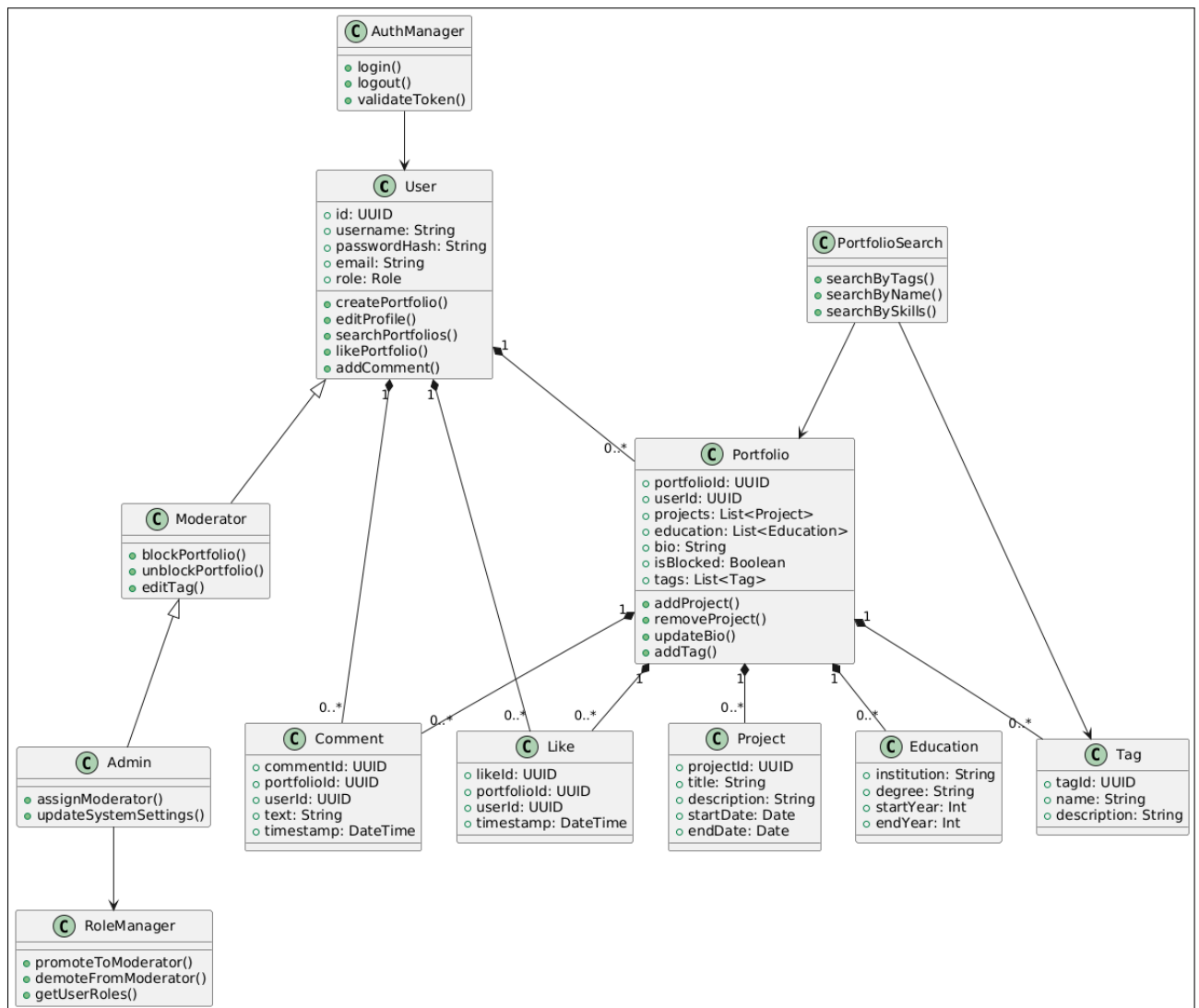


Рисунок 4.1 — Диаграмма классов системы. Выполнена с помощью [1].

5 Создание блок-схем

В данной главе будут разработаны блок-схемы основополагающих алгоритмов клиент-серверного приложения «Агрегатор портфолио».

5.1 Определение основополагающих алгоритмов системы

В системе можно выделить несколько основополагающих алгоритмов:

- Авторизация в аккаунт.
- Добавление проекта в портфолио.
- Поиск портфолио.
- Блокировка портфолио модератором.
- Назначение модератора администратором.
- Лайк портфолио.
- Добавление комментария.

5.2 Авторизация пользователя

Авторизация пользователя является сложной задачей в плане реализации и сохранения безопасности. Для реализации используется класс AuthManager и хэш-функции внутри него. Полный алгоритм представлен на рисунке 5.1.



Рисунок 5.1 — Блок-схема авторизации пользователя. Выполнена с помощью [1]

5.3 Добавление проекта в портфолио

Каждый пользователь может добавить проекты в свое портфолио, только если у него есть на это права (то есть если это его портфолио и если его аккаунт не заблокирован). Алгоритм представлен на рисунке 5.2.



Рисунок 5.2 — Блок-схема добавления проекта в портфолио. Выполнена с помощью [1]

5.4 Поиск портфолио

Портфолио являются публичными на платформе и каждый может найти портфолио других пользователей через поиск. Алгоритм представлен на рисунке 5.3.

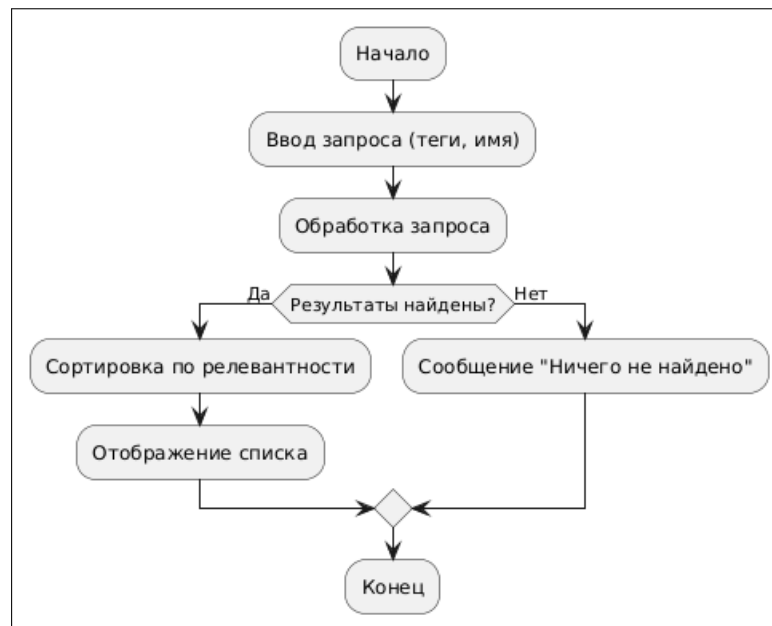


Рисунок 5.3 — Блок-схема поиска портфолио. Выполнена с помощью [1]

5.5 Блокировка портфолио модератором

Каждый модератор или администратор может заблокировать портфолио пользователя, только если в нем есть нарушения. Алгоритм представлен на рисунке 5.4.

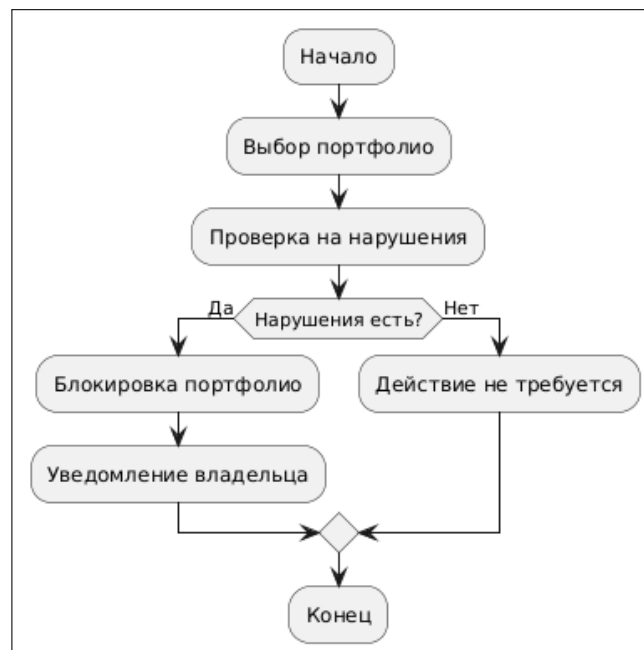


Рисунок 5.4 — Блок-схема блокировки портфолио модератором. Выполнена с помощью [1]

5.6 Назначение пользователя модератором

Администратор может назначить пользователя модератором, если пользователь существует. Алгоритм представлен на рисунке 5.5.

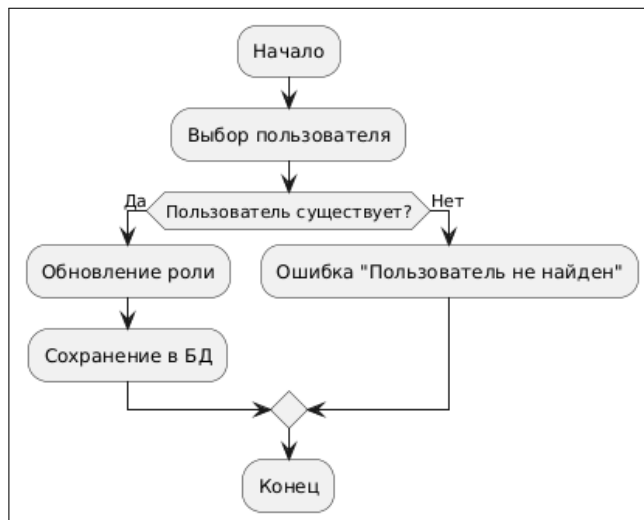


Рисунок 5.5 — Блок-схема назначения пользователя модератором. Выполнена с помощью [1]

5.7 Лайк портфолио

Каждый пользователь может поставить лайк понравившемуся портфолио на платформе. Алгоритм представлен на рисунке 5.6.

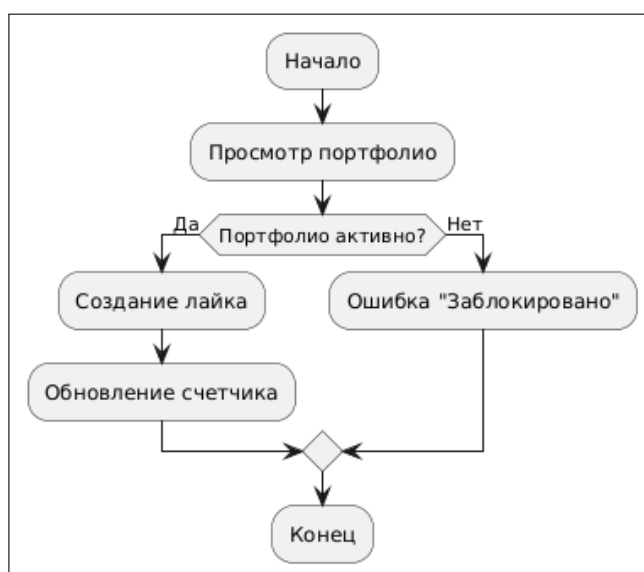


Рисунок 5.6 — Блок-схема лайка портфолио. Выполнена с помощью [1]

5.8 Комментарий портфолио

Каждый пользователь может оставить комментарий понравившемуся портфолио на платформе. Алгоритм представлен на рисунке 5.7.

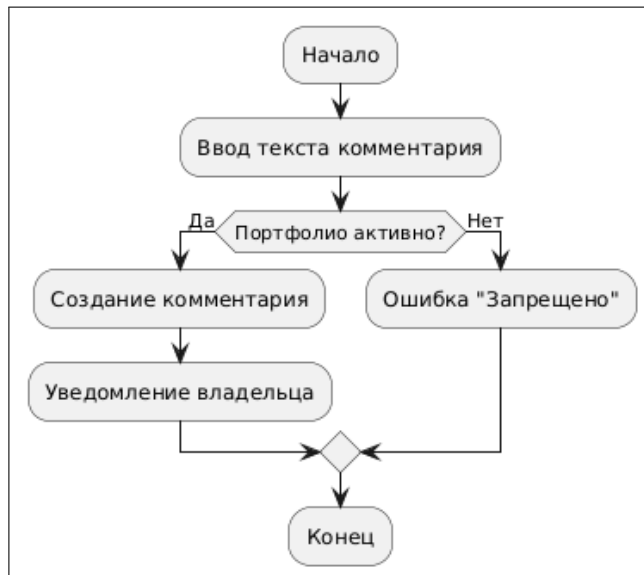


Рисунок 5.7 — Блок-схема комментария портфолио. Выполнена с помощью [1]

ЗАКЛЮЧЕНИЕ

В ходе работы были создана базисная основа приложения, опираясь на которую, можно разработать качественный и рабочий продукт.

Поставленные задачи в виде построения UML-диаграммы, диаграммы классов и создание блок-схем некоторых алгоритмов были выполнены.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Онлайн-редактор plantuml - URL: <https://plantuml.com/> (дата обращения: 10.02.2025).
2. LinkedIn [Электронный ресурс]: [сайт]. – URL: <https://www.linkedin.com/> (дата обращения: 10.02.2025)
3. Что такое Сетка? [Электронный ресурс]: [сайт]. – URL: <https://setka.ru/faq> (дата обращения: 10.02.2025)
4. Taplink [Электронный ресурс]: [сайт]. – URL: <https://taplink.ru/> (дата обращения: 10.02.2025)
5. GitHub [Электронный ресурс]: [сайт]. – URL: <https://github.com/> (дата обращения: 10.02.2025)
6. React – The library for web and native user interfaces [Электронный ресурс]: [сайт]. – URL: <https://react.dev/> (дата обращения: 10.02.2025)
7. Java Documentation [Электронный ресурс]: [сайт]. – URL: <https://docs.oracle.com/en/java/> (дата обращения: 10.02.2025)
8. Spring Documentation [Электронный ресурс]: [сайт]. – URL: <https://docs.spring.io/spring-framework/reference/overview.html> (дата обращения: 10.02.2025)
9. LinkedIn [Электронный ресурс]: [сайт]. – URL: <https://www.linkedin.com/> (дата обращения: 10.02.2025)