

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Новгородский государственный университет имени Ярослава Мудрого»  
ИНСТИТУТ ЭЛЕКТРОННЫХ И ИНФОРМАЦИОННЫХ СИСТЕМ  
ПОЛИТЕХНИЧЕСКИЙ КОЛЛЕДЖ

УТВЕРЖДАЮ

Зам. директора по УМ и ВР

\_\_\_\_\_ Л.Н. Иванова

«\_\_\_» \_\_\_\_\_ 2023 г.

РАЗРАБОТКА ИНФОРМАЦИОННОЙ СИСТЕМЫ ДЛЯ ПРОЕКТНОЙ  
ДЕЯТЕЛЬНОСТИ НОВГУ

Пояснительная записка к дипломному проекту по специальности  
09.02.03 Программирование в компьютерных системах  
ПТК. ДП 9903 06. 000ПЗ

Согласовано:

Консультант по спец. части

\_\_\_\_\_ Л.Н. Цымбалюк

«\_\_\_» \_\_\_\_\_ 2023 года

Консультант по экон. части

\_\_\_\_\_ Г.В. Лебедева

«\_\_\_» \_\_\_\_\_ 2023 года

Нормоконтроль

\_\_\_\_\_ А.М. Чернега

«\_\_\_» \_\_\_\_\_ 2023 года

Заместитель директора по УПР

\_\_\_\_\_ А.М. Чернега

«\_\_\_» \_\_\_\_\_ 2023 года

Руководитель

\_\_\_\_\_ Н.В. Сазонова

«\_\_\_» \_\_\_\_\_ 2023 года

Выполнил:

обучающийся группы 9903

\_\_\_\_\_ О.Г. Молоканов

«\_\_\_» \_\_\_\_\_ 2023 года

Введение .....	3
1    Общая часть .....	5
1.1   Постановка задачи .....	5
1.2   Обоснование проектных решения .....	8
1.3   Обзор и анализ существующих программных систем .....	10
2    Специальная, практическая, опытно-экспериментальная часть .....	12
2.1   Анализ задачи .....	12
2.2   Описание логической структуры .....	15
2.3   Описание работы программы .....	21
2.4   Руководство оператора .....	24
3    Экономическая часть .....	28
3.1   Расчёт основной и дополнительной заработной платы .....	28
3.2   Расчёт стоимости материалов и программного обеспечения .....	30
3.3   Расчёт накладных расходов .....	31
3.4   Составление и расчёт цены реализации программного продукта .....	31
Заключение .....	33
Список литературы .....	35
Список обозначений .....	36
Приложение А .....	37
Приложение Б .....	48
Приложение В .....	52
Приложение Г .....	60

Проектная деятельность позволяет обучающимся применить на практике теоретические знания, что имеет очень большое значение в процессе обучения, а также повышает конкурентоспособность выпускников на рынке труда, так как современными работодателями практические навыки потенциальных работников ценятся выше одних лишь теоретических навыков.

Обучающиеся приобретают такие навыки как работа в команде, так называемые «мягкие навыки» (soft skills), работа в условиях ограниченного количества времени и ресурсов, генерирование новых идей и подходов к работе, умение работать с заказчиком проекта, способность брать ответственность на себя за принятые решения и итоговый продукт.

Разработка информационной системы для проектной деятельности могла бы помочь в более слаженной командной работе обучающихся над проектом, путём создания проектов в самой информационной системе и формирование команды с распределением роли каждого участника, в том числе создание вакансий для обучающихся на недостающие роли в конкретном проекте, сделать контроль и оценивание выполнения проектной работы для руководителя проекта более комфортным, а также создать более благоприятные условия для слаженного взаимодействия руководителя проекта и обучающихся на протяжении всех этапов работы над проектом, кроме этого информационная система способствовала бы работе администратору проектной деятельности, в этом и заключается практическая значимость выбранной темы дипломного проектирования.

Разрабатываемая информационная система предназначена для средних специальных учебных заведений и высших учебных заведений, в которых предусмотрена проектная деятельность в рамках учебного плана обучающихся в различных направлениях подготовки.

Объектом исследования дипломного проектирования является проектная деятельность. Предметом исследования является использование информационной системы в проектной деятельности.

Целью исследования дипломного проектирования является создание информационной системы для проектной деятельности.

В соответствие с поставленной целью необходимо решить следующие задачи дипломного проектирования:

- произвести анализ предметной области информационной системы;
- осуществить обзор аналогов информационной системы;
- выявить экономическую целесообразность создание информационной системы;
- спроектировать архитектуру разрабатываемого приложения;
- разработать спроектированное приложение;
- составить справочное руководство и руководство пользователя;
- протестировать и отладить работу созданного приложения.

## 1 Общая часть

### 1.1 Постановка задачи

#### 1.1.1 Обоснование необходимости разработки

В соответствии с заданием на выпускную квалификационную работу, требуется разработать информационную систему для работы проектной деятельности. Требуется разработать программный продукт для создания проектов для команд по проектной деятельности, а также должен присутствовать следующий функционал:

- возможность добавлять и изменять информацию о проекте участником проекта;
- поиск, фильтрация и сортировка пользователем проектов;
- возможность пользователю подать заявку на вступление в проект;
- возможность участнику проекта принять заявку от пользователя на вступление в проект;
- администрирование базы данных администратором системы;
- возможность выставление оценок проекту наставниками проектов;
- самостоятельная регистрация и авторизация пользователей в системе;
- просмотр своих проектов и информации о себе пользователем в личном кабинете;
- редактирование информации о себе пользователем.

Исходя из вышесказанного, построим функциональную модель предметной области информационной системы и представим её на рисунке А.1.

Создание информационной системы для работы проектной деятельности, сводится к следующим этапам:

- произвести анализ предметной области разрабатываемой системы;
- осуществить обзор аналогов разрабатываемой системы;
- сформировать требования к системе;

- спроектировать архитектуру разрабатываемого приложения;
- разработать базу данных средствами СУБД\*;
- создать разрабатываемое приложение;
- описать методы защиты и их реализации;
- составить справочного руководства и руководства пользователя;
- протестировать работу созданного приложения.

Разрабатываемый программный продукт предназначен для упрощения работы над проектом как для обучающихся, так и для руководителей проектной деятельности.

#### 1.1.2 Техничко-математическое описание задачи

Программный продукт будет разработан с помощью применения следующих технологий:

- технология ADO.NET;
- технология WPF;
- технология CLR.

##### 1.1.2.1 Технология ADO.NET

ADO.NET предоставляет собой технологию работы с данными, которая основана на платформе .NET Framework. Эта технология представляет нам набор классов, через которые мы можем отправлять запросы к базам данных, устанавливать подключения, получать ответ от базы данных и производить ряд других операций.

ADO.NET поставляется с тремя пространствами имен клиента базы данных: одно для SQL Server, другое для источников данных Open Database Connectivity и третье для любой базы данных, доступной через OLE DB.

##### 1.1.2.2 Технология WPF

WPF является часть экосистемы платформы .NET и представляет собой подсистему для построения графических интерфейсов.

---

\* Все обозначения, встречающиеся в тексте пояснительной записки, представлены в «Список обозначений»

Если при создании традиционных приложений на основе Windows Forms за отрисовку элементов управления и графики отвечают такие части операционной системы Windows, как User32 и GDI\*, то приложения WPF основаны на DirectX. В этом состоит ключевая особенность рендеринга графики в WPF: используя WPF, значительная часть работы по отрисовке графики, как простейших кнопочек, так и сложных 3D-моделей, ложиться на графический процессор видеокарты, что также позволяет воспользоваться аппаратным ускорением графики.

Одной из важных особенностей является использование языка декларативной разметки интерфейса XAML, основанного на XML. С его помощью можно создавать насыщенный графический интерфейс, используя или декларативное объявление интерфейса, или код на управляемых языках C#, Visual Basic .NET и F#, либо совмещать и то, и другое.

#### 1.1.2.3 Технология CLR

CLR – это компонент платформы .NET Framework, виртуальная машина, исполняющая программа, написанная на .NET-совместимых языках программирования.

CLR интерпретирует и исполняет код на языке IL, а также предоставляет MSIL-программам доступ к библиотекам классов .NET Framework. Среда CLR является реализацией спецификации CLI, спецификации общезыковой инфраструктуры компании Microsoft.

#### 1.1.3 Характеристика бизнес-процессов

Так как одной из задач дипломного проекта является создания приложения, то для его работы необходимо создание базы данных, в которой, помимо всего прочего, будут храниться пользователи системы. Пользователей необходимо разбить на уровни доступа к системе, следовательно, необходимо определить основные функции, которые будут выполняться в приложении разными категориями пользователей системы.

---

\* Все обозначения, встречающиеся в тексте пояснительной записки, представлены в «Список обозначений»

Разобьем пользователей разрабатываемой системы на две группы: пользователи, которые будут иметь неограниченный и ограниченные типы доступа и представим диаграмму на рисунке А.2.

Для графического изображения того, как организованы бизнес-процессы проектной деятельности построена диаграмма бизнес-прецедентов и представим её на рисунке А.3, на данной диаграмме можно наглядно увидеть, какие функции выполняют работники организации (бизнес-процессы).

#### 1.1.4 Требования к программе

Требования к оборудованию:

- процессор x86 (Intel, AMD\*) с частотой как минимум 1,8 ГГц, рекомендуется использовать как минимум двухъядерный процессор;
- 2 ГБ ОЗУ, рекомендуется 4 ГБ ОЗУ (минимум 2,5 ГБ при выполнении на виртуальной машине);
- доступное место на жестком диске объемом от 2 до 10 ГБ;
- жесткий диск 5400 об/мин;
- видеоадаптер с поддержкой DirectX 9 и разрешением экрана 600x800 или выше.

Требования к системе:

- операционная система: Windows 2016 Server, Windows Server 2019, Windows 2022 Server, Windows 7, 8, 8.1, 10, 11.

### 1.2 Обоснование проектных решения

#### 1.2.1 Обоснование выбора языков программирования

Для написания программы был выбран язык программирования С#. С# – объектно-ориентированный, ориентированный на компоненты язык программирования. С# предоставляет языковые конструкции для

---

\* Все обозначения, встречающиеся в тексте пояснительной записки, представлены в «Список обозначений»



непосредственной поддержки такой концепции работы. Благодаря этому C# подходит для создания и применения программных компонентов.

### 1.2.2 Инструментальные средства

Поскольку проектируемая система является многопользовательской, в качестве сервера баз данных был выбран MS\* SQL Server.

MS SQL Server – это платформа для решения критически важных задач в масштабе предприятия, обладающая высокой доступностью, повышенной производительностью и безопасностью. Решение представляет собой хорошо масштабируемый, полностью реляционный, быстродействующий сервер, способный обрабатывать большие объемы данных для клиент-серверных приложений. Рекордная производительность MS SQL Server обеспечивается новыми технологиями работы с памятью. Кроме того, SQL Server имеет расширенные функции безопасности, в сочетании со встроенными, удобными для использования инструментами и управляемым доступом к данным, автоматизацию рутинных административных задач.

### 1.2.3 Обоснование выбора среды программирования

Для создания приложения была выбрана интегрированная среда разработки (IDE) MS Visual Studio 2022. Visual Studio предоставляет набор инструментов, облачных сервисов и расширения, которые необходимы для разработки приложения. Visual Studio помогает редактировать, отлаживать и создавать код, а затем публиковать приложение. Visual Studio включает в себя компиляторы, средства автодополнения кода, визуальные редакторы макетов и многие другие функции, облегчающие процесс разработки программного обеспечения.

### 1.2.4 Информационное обеспечение

Для построения диаграмм и блок-схем был выбран редактор - MS Visio 2016. Для оформления документации к приложения был выбран текстовый редактор - MS Word. Для презентации работы была выбрана программа для

---

\* Все обозначения, встречающиеся в тексте пояснительной записки, представлены в «Список обозначений»

подготовки презентаций и их просмотра - MS PowerPoint. Все эти программы являются частью офисного пакета приложений - MS Office. И их интегрированность друг с другом позволяет существенно упростить работу.

### 1.3 Обзор и анализ существующих программных систем

На рынке существуют аналоги разрабатываемой программной системы. Рассмотрим некоторые системы управления проектами:

- Trello;
- Битрикс24.

#### 1.3.1 Trello

Trello — платформа, помогающую работать с применением облачных технологий. Работа сервиса основана на принципах канбан, благодаря которым организовать управление командами, проектами стало легко, вне зависимости от масштаба.

Преимущества Trello:

- понятный интерфейс;
- набор необходимых функций работы даже в бесплатной версии;
- возможность интегрировать данные из других приложений;
- подойдет для планирования личных задач, управления продуктом и командой;
- легко внедрить в коллектив;
- многоплатформенность.

Недостатки Trello:

- в бесплатной версии ограничение на количество расширений;
- функционала недостаточно для действительно крупных компаний.

#### 1.3.2 Битрикс24

Битрикс24 предоставляет комплексный инструмент организации совместной работы с проектами. Сервис помогает эффективно управлять

несколькими проектами одновременно, контролировать работу на каждом этапе и оценивать успешность выполнения.

#### Преимущества Битрикс24:

- бесплатный тариф с неограниченным сроком действия;
- постоянное улучшение платформы;
- многоуровневая система безопасности;
- огромный маркетплейс приложений с продуктами;
- высокое быстродействие;
- стабильная работа.

#### Недостатки Битрикс24:

- возможности бесплатного тарифа сильно ограничены;
- нельзя связаться с представителями сервиса без оплаты подписки;
- постоянная агитация перехода на платный тариф;
- длительное обучение;
- масштабные обновления интерфейса.

## 2 Специальная, практическая, опытно-экспериментальная часть

### 2.1 Анализ задачи

#### 2.1.1 Информационное моделирование предметной области

На основе анализа предметной области при моделировании базы данных были выделены следующие сущности:

- projects (проекты), атрибуты: id\_project (уникальный идентификатор), project\_type (тип проекта), project\_name (название проекта), date\_creation (дата создания проекта), customer (заказчик проекта), project\_objective (цель проекта), project\_tasks (задачи проекта), project\_problem (проблематика проекта), project\_timeline (сроки выполнения проекта), product\_type (тип разрабатываемого продукта), project\_relevance (актуальность проекта), required\_resources (требуемые ресурсы), target\_audience (целевая аудитория), analogues\_competitors (аналоги и конкуренты), project\_novelty (новизна проекта), business\_plan (бизнес-план проекта), achieved\_level (достигнутый уровень), final\_result (финальный результат), project\_logo (логотип проекта), project\_grades (оценки проекта), project\_vacancies (вакансии проекта);
- users (пользователи), атрибуты: id\_user (уникальный идентификатор), surname (фамилия), name (имя), patronymic (отчество), email (электронная почта), phone (телефон), role (роль), login (логин), password (пароль), avatar (аватарка);
- roles (роли пользователей), атрибуты: id\_role (уникальный идентификатор), role\_name (название роли), access\_level (уровень доступа);
- project types (типы проектов), атрибуты: id\_project\_type (уникальный идентификатор), type\_name (название типа);
- product types (типы продуктов), атрибуты: id\_product\_type (уникальный идентификатор), type\_name (название типа);
- teams (состав команд), атрибуты: id\_user (идентификатор пользователя), id\_project (идентификатор проекта), team\_role (роль в команде), application\_status (статус заявки);

- team roles (роли в команде), атрибуты: id\_team\_role (уникальный идентификатор), name\_team\_role (название роли);
- achieved levels (достигнутые уровни), атрибуты: id\_achieved\_level (уникальный идентификатор), level\_name (название уровня).
- application status (статусы заявок), атрибуты: id\_application\_status (уникальный идентификатор), name\_application\_status (название статуса заявки);
- grades (оценки), атрибуты: id\_grade (уникальный идентификатор), final\_grade (итоговая оценка), idea\_grade (оценка за идею), team\_grade (оценка за команду), innovation\_grade (оценка за инновацию), marketing\_grade (оценка за маркетинг), economy\_grade (оценка за экономику), management\_grade (оценка за менеджмент), business\_plan\_grade (оценка за бизнес-план), presentation\_grade (оценка за презентацию).

ER-модель базы данных представлена на рисунке А.4.

Реляционная модель базы данных представлена на рисунке А.5.

Произведем анализ сущностей:

- сущности находятся в первой нормальной форме, так как удовлетворяют следующему требованию: нет повторяющихся групп;
- сущности находятся во второй нормальной форме, так как удовлетворяют следующему требованию: устранены избыточные данные;
- сущности находятся в третьей нормальной форме, так как удовлетворяют следующему требованию: устранены данные, не зависящих от ключа.

В СУБД для защиты базы данных были реализованы следующие методы защиты:

- защита паролем, шифрование данных (алгоритм хэширования MD5\*);
- разграничение прав доступа к объектам базы данных;
- поддержание целостности данных в СУБД.

---

\* Все обозначения, встречающиеся в тексте пояснительной записки, представлены в «Список обозначений»

### 2.1.2 Проектирование пользовательского интерфейса

Для разработки интерфейса для основных экранов приложения были разработаны эскизы:

- эскиз экрана регистрации пользователя представлен на рисунке Г.1;
- эскиз экрана авторизации пользователя представлен на рисунке Г.2;
- эскиз экрана списка проектов представлен на рисунке Г.3;
- эскиз экрана создания нового проекта представлен на рисунке Г.4;
- эскиз экрана редактирования проекта представлен на рисунке Г.5;
- эскиз экрана личного кабинета представлен на рисунке Г.6;
- эскиз экрана редактирования профиля представлен на рисунке Г.7.
- эскиз экрана панели администратора представлен на рисунке Г.8.

Описание разработки интерфейса:

- экран авторизации пользователя с полями для ввода логина и пароля, кнопкой «Войти» и гиперссылкой «Регистрация» представлен на рисунке Г.9;
- экран регистрации пользователя с полями для ввода личных данных, кнопкой «Зарегистрироваться» и гиперссылкой «Авторизация» представлен на рисунке Г.10;
- экран списка всех проектов, с возможностью поиска, сортировки и фильтрации проектов, а также кнопка «Добавить проект» представлен на рисунке Г.11;
- экран создания нового проекта с полем для ввода названия проекта и кнопкой «Добавить новый проект» представлен на рисунке Г.12;
- экран редактирования проекта с полями для ввода информации о проекте, оценках проекта и кнопках «Сохранить изменения», «Оценить», «Подать заявку на вступление в проект», а также участниках проекта представлен на рисунке Г.13;
- экран личного кабинета пользователя, с информацией о пользователе и проектах, в которых он участвует, а также кнопка «Редактировать профиль» представлен на рисунке Г.14;

- экран редактирования профиля с полями для ввода личных данных и кнопкой «Сохранить изменения» представлен на рисунке Г.15;
- экран панели администратора с кнопками для перехода представлен на рисунке Г.16.

## 2.2 Описание логической структуры

### 2.2.1 Алгоритм программы

При запуске программы открывается форма авторизации в системе, на которой пользователь должен ввести свой логин и пароль (пользователь может посмотреть введённый пароль по переключению чек-бокса в состояние «включен»), которые пользователь указал при регистрации, в поля «Логин» и «Пароль» соответственно. После ввода логина и пароля пользователь должен нажать на кнопку «Войти», после этого программа проверит наличие введённых данных в базе данных с помощью LINQ\* запроса к базе данных, заранее программа зашифрует пароль, введённый пользователем, с помощью технологии шифрования MD5, так как именно в таком виде он хранится в базе данных. Если данные введены некорректно, то программа выдаст сообщение об ошибке с текстом «Пользователь не найден!», если данные введены корректно, то программа выдаст сообщение об успешной авторизации с текстом «Успешный вход!», пользователь перейдёт на форму со списком всех проектов, а в шапке приложения появится кнопка с личной информацией авторизовавшегося пользователя, при нажатии на которую пользователь перейдёт на форму личного кабинета. Если пользователь не зарегистрирован в системе, то при нажатии на гиперссылку «Регистрация» пользователь перейдёт на форму регистрации в системе. Интерфейс формы авторизации представлен на рисунке Г.10. Алгоритм регистрации представлен на рисунках А.6 и А.7.

---

\* Все обозначения, встречающиеся в тексте пояснительной записки, представлены в «Список обозначений»

На форме регистрации в системе пользователь должен заполнить поля «Фамилия», «Имя», «Отчество», «Email», «Телефон» своими личными данными, выбрать роль из выпадающего списка «Ваша роль», а также придумать логин и пароль для входа в систему и внести их в поля «Логин» и «Пароль» соответственно, вдобавок повторить свой пароль в поле «Повторите пароль» (пользователь может посмотреть введенные пароли по переключению чек-бокса в состояние «включен»). В поле «Ваша роль» программа позволяет пользователю выбрать только имеющиеся роли в системе с помощью выпадающего списка. Поля «Телефон» и «Email» пользователь должен заполнить по установленным маскам, которые появляются в подсказке над соответствующими полями при вводе данных в эти поля. После ввода всех данных пользователь нажимает кнопку «Зарегистрироваться», программа выполняет проверку на валидацию полей, если хоть одно поле заполнено некорректно или не заполнено обязательное к заполнению поле, то программа подсветит такие поля красным цветом и выдаст подсказку над таким полем, после этого пользователь может продолжить внесение данных. Если все поля были заполнены корректно, то программа проверит, что пароль в поле «Повторите пароль» соответствует паролю в поле «Пароль», если присутствует несоответствие, то программа выдаст сообщение об ошибке с текстом «Пароли не совпадают!», если пароли соответствуют, то программа зашифрует пароль пользователя с помощью технологии шифрования MD5, далее она выполнит LINQ запрос на добавления всех данных в таблицу «Users» (Пользователи) базы данных, выведет информационное сообщение на экран, что регистрация прошла успешно с текстом «Вы успешно зарегистрировались в системе!», закроет форму регистрации сотрудника и вернется на форму авторизации. Интерфейс формы регистрации сотрудника представлен на рисунке Г.9. Алгоритм авторизации представлен на рисунках А.8 и А.9.

На форме личного кабинета выводится основная информация о пользователе: фамилия, имя, отчество, роль в системе. А также выводится список всех проектов, в которых участвует данный пользователь с основной



информацией о проекте. По нажатию на проект пользователь перейдёт на форму редактирования проекта. Под личной информацией пользователя присутствует кнопка «Редактировать профиль» при нажатии на которую пользователь перейдёт на форму редактирования своего профиля. Интерфейс формы личного кабинета пользователя представлен на рисунке Г.14.

На форме редактирования профиля пользователь имеет право изменить информацию о себе, а именно: фамилию, имя, отчество, электронную почту, номер телефона, свою роль в системе, логин, а также изменить пароль. Для изменения данных пользователь должен ввести действующий пароль, а для изменения пароля ввести ещё и новый пароль, а также подтвердить новый пароль (пользователь может посмотреть введённые пароли по переключению чек-бокса в состояние «включен»). После нажатия на кнопку «Сохранить изменения» программа проверит валидацию полей, поля заполненные некорректно будут подсвечены красным цветом, а также появится подсказка над соответствующим полем. Если пользователь ввёл неверный действующий пароль, то программа выдаст сообщение об ошибке с текстом «Вы ввели неверный пароль!», а если пользователь оставил это поле пустым, то сообщение об ошибке с текстом «Вы не ввели текущий пароль!». Если пользователь не ввёл новый пароль при смене пароля, то программа выдаст сообщение об ошибке с текстом «Вы не ввели новый пароль!», а если новый пароль и подтверждение нового пароля не совпали, то выдаст сообщение об ошибке с текстом «Подтверждение пароля и новый пароль не совпадают!». Интерфейс формы редактирования профиля представлен на рисунке Г.15.

На форме список проектов демонстрируются все проекты. Информация о проекте представлена названием проекта, типом проекта, целью проекта, датой создания проекта, а также вакансиями проекта. По нажатию на проект пользователь перейдёт на редактирование соответствующего проекта. Пользователь имеет возможность сортировать проекты по дате созданию проекта, фильтровать проекты по типу проекта, производить поиск проекта по названию. А также пользователь может добавить новый проект нажав на кнопку

«Добавить новый проект», после чего пользователь перейдёт на форму создания нового проекта. Интерфейс формы списка проектов представлен на рисунке Г.11.

На форме создания нового проекта пользователю необходимо придумать название проекта заполнив поле «Название проекта» и нажать кнопку «Добавить новый проект», после чего программа создаст новый проект и откроет форму списка проектов. Если пользователь оставит поле «Название проекта» пустым и нажмёт кнопку «Добавить новый проект», то программа выдаст сообщение об ошибке с текстом «Укажите название проекта!», а если поле заполнено, то программа выдаст сообщение об успешном создании нового проекта с текстом «Вы успешно создали новый проект!». Интерфейс формы создания нового проекта представлен на рисунке Г.12.

На форме редактирования проекта пользователь имеет право просматривать информацию о проекте, его участниках и оценках, а участник проекта имеет право редактировать информацию о проекте. После редактирования проекта пользователь должен нажать на кнопку «Сохранить изменения», чтобы отредактированные данные были успешно изменены, после чего программа выдаст сообщение об успешном редактировании проекта с текстом «Проект успешно изменен!». Также наставники проектов могут выставять оценки проекту, для этого они должны заполнить соответствующие поля и нажать кнопку «Оценить», после чего программа вычислит итоговую оценку и выведет её на экран, все оценки будут добавлены в базу данных в таблицу «Grades» (Оценки). Пользователи проекта могут подать заявку на вступление в проект, для это им необходимо нажать на кнопку «Подать заявку на вступление в проект», если пользователь уже состоит в данном проекте, то программа выдаст сообщение с текстом «Вы уже состоите в этом проекте!», если пользователь повторно подаёт заявку на вступление в проект, то программа выдаст сообщение с текстом «Вы уже подали заявку в этот проект!», в ином случае программа выдаст сообщение с текстом «Вы успешно подали заявку на вступление в проект!». Участники проекта могут добавлять пользователей, которые подали заявку, нажав на конкретного пользователя, а после нажав «ОК»

в сообщении с текстом «Хотите добавить этого пользователя в свою команду?»  
Интерфейс формы редактирования проекта представлен на рисунке Г.13.  
Алгоритм выставление оценки проекту представлен на рисунках А.10 и А.11.

### 2.2.2 Используемые методы

Используемые методы представлены в таблице 1.

Таблица 1 – Используемые методы

Назначение	Имя	Параметры		Тестирует ся ли локально
		входные	выходные	
1	2	3	4	5
Открывает главное окно приложения	MainWindow	-	-	да
Записывает данные в класс	UpdateData	object sender, object e	-	да
Переход на форму авторизации	btnExit_Click	object sender, RoutedEventArgs e	-	да
Переход на форму личного кабинета	btnPrsnlAcc_C lick	object sender, RoutedEventArgs e	-	да
Переход на форму проектов	btnProject_Cli ck	object sender, RoutedEventArgs e	-	да
Открытие формы авторизации	PageAuthoriza tion	-	-	да
Авторизация в системе	btnAuthorizati on_Click	object sender, RoutedEventArgs e	-	да
Просмотр пароля	ShowPass_Che cked	object sender, RoutedEventArgs e	-	да
Скрытие пароля	ShowPass_Un checked	object sender, RoutedEventArgs e	-	да
Открытие формы регистрации	PageRegistrati on	-	-	да
Регистрация в системе	btnRegistration _Click	object sender, RoutedEventArgs e	-	да
Обновление проектов пользователя	UpdateDataMy Project	object sender, object e	-	да
Переход на форму редактирования профиля	btnPageEditPr ofile_Click	object sender, RoutedEventArgs e	-	да
Переход на форму редактирования проекта	btnEditProject _Click	object sender, RoutedEventArgs e	DataContext as Projects	да
Обновление проектов	UpdateDataPro ject	object sender, object e	-	да
Переход на форму добавления проекта	btnAddProject _Click	object sender, RoutedEventArgs e	-	да
Открытие формы добавления проектов	PageAddProje ct	-	-	да
Добавление нового проекта	btnAddProject _Click	object sender, RoutedEventArgs e	-	да

Окончание таблицы 1

1	2	3	4	5
Открытие формы редактирования проекта	PageEditProject	Projects project	-	да
Обновления данных о команде проекта	UpdateDataTeams	-	-	да
Сохранение результатов изменения проекта	btnSaveChanges_Click	object sender, RoutedEventArgs e	-	да
Обновление данных полей	UpdateTextBox	-	-	да
Выставление оценки проекту	btnResult_Click	object sender, RoutedEventArgs e	-	да
Подача заявки на вступление в проект	btnSubmitApply_Click	object sender, RoutedEventArgs e	-	да
Принятия заявки на вступление в проект	btnAddInTeam_Click	object sender, RoutedEventArgs e	-	да
Удаление проекта из системы	btnDeleteProject_Click	object sender, RoutedEventArgs e	-	да
Открытие формы изменения профиля	PageEditProfile	Users user, int PageBack	-	да
Сохранение результатов изменения профиля	btnEditProfile_Click	object sender, RoutedEventArgs e	-	да
Удаления пользователя из системы	btnDeleteUser_Click	object sender, RoutedEventArgs e	-	да
Вычисление итоговой оценки проекта	final_grade	double one, double two, double three, double four, double five, double six, double seven, double eight	double result	да
Хеширование пароля	hashing	string getString	string result	да
Открытие формы панели администратора	PageAdminPanel	-	-	да
Переход на форму списка проектов	btnProjects_Click	object sender, RoutedEventArgs e	-	да
Переход на форму списка пользователей	btnUsers_Click	object sender, RoutedEventArgs e	-	да

### 2.2.3 Составные части программы и связи между ними

Программа состоит из трёх модулей: база данных, приложения и модуль среды программирования. Связь между базой данных и приложением осуществляется с помощью LINQ запросов, которые приложение направляет к базе данных.

## 2.3 Описание работы программы

### 2.3.1 Общие сведения

Программа называется – «Project».

Первым экраном при запуске приложения будет экран авторизации в системе, где пользователь должен авторизоваться или перейти на экран регистрации для регистрации в системе, если он не зарегистрирован в системе. После прохождения авторизации (регистрации и авторизации, если пользователь еще не зарегистрирован) пользователь перейдет на экран списка проектов. На экране списка проектов пользователь может сортировать проекты по дате созданию, фильтровать проекты по типу проекта, кроме этого, возможен поиск проектов по названию и переход к любому проекту по нажатию на него, вдобавок пользователь имеет право добавить новый проект по нажатию на соответствующую кнопку. На экране проектов пользователь может редактировать информацию о своём проекте, выставлять оценки проекту, если имеют на это соответствующие права. В личном кабинете пользователь может просматривать информацию о себе и своих проектах, а также перейти на экран редактирования своего профиля.

### 2.3.2 Функциональное назначение

Программа позволяет авторизовываться и регистрироваться пользователям в системе. Студенты могут редактировать свои проекты, оставлять заявки на вступление в проекты, а участники проекта могут откликаться на эти заявки и добавлять новых участников в свой проект, также студенты имеют возможность добавлять новые проекты, просматривать другие проекты, наставники также могут оценивать проекты по нескольким категориям, а программа подсчитывает итоговую оценку проекта, системные администраторы имеют возможность производить администрирование базы данных, а именно изменять, добавлять и удалять записи в базе данных через приложение, все пользователи имеют право отредактировать свой профиль в системе через личный кабинет.

## 2.3.3 Входные и выходные данные

Входные и выходные данные представлены в таблице 2.

Таблица 2 – Входные и выходные данные

Назначение	Имя	Характеристика			Характеристика использования: входная/выходная /рабочая
		тип	диапазон	простая/структурная	
1	2	3	4	5	6
Обновление по времени	timer	DispatcherTimer	-	структурная	входная/выходная
Подключения к базе данных	conObj	db_projectEntities	-	структурная	рабочая
Навигация по формам	frameObj	Frame	-	структурная	рабочая
Хранение информации	id_user	int	от 2147483648 до 2147483647	простая	входная/выходная
Хранение информации	surname	string	-	структурная	входная/выходная
Хранение информации	name	string	-	структурная	входная/выходная
Хранение информации	patronymic	string	-	структурная	входная/выходная
Хранение информации	email	string	-	структурная	входная/выходная
Хранение информации	phone	string	-	структурная	входная/выходная
Хранение информации	role	int	от 2147483648 до 2147483647	простая	входная/выходная
Хранение информации	login	string	-	структурная	входная/выходная
Хранение информации	password	string	-	структурная	входная/выходная
Хранение информации	id_project	int	от 2147483648 до 2147483647	простая	входная/выходная
Хранение информации	project_type	int	от 2147483648 до 2147483647	простая	входная/выходная
Хранение информации	project_name	string	-	структурная	входная/выходная
Хранение информации	date_creation	DateTime	-	структурная	входная/выходная
Хранение информации	customer	string	-	структурная	входная/выходная

Окончание таблицы 2

1	2	3	4	5	6
Хранение информации	project_objective	string	-	структурная	входная/выходная
Хранение информации	project_tasks	string	-	структурная	входная/выходная
Хранение информации	project_problem	string	-	структурная	входная/выходная
Хранение информации	project_timeline	string	-	структурная	входная/выходная
Хранение информации	product_type	int	от 2147483648 до 2147483647	простая	входная/выходная
Хранение информации	project_relevance	string	-	структурная	входная/выходная
Хранение информации	required_resources	string	-	структурная	входная/выходная
Хранение информации	tagged_audience	string	-	структурная	входная/выходная
Хранение информации	analgues_competitors	string	-	структурная	входная/выходная
Хранение информации	project_novelty	string	-	структурная	входная/выходная
Хранение информации	business_plan	string	-	структурная	входная/выходная
Хранение информации	achieved_level	string	-	структурная	входная/выходная
Хранение информации	final_result	string	-	структурная	входная/выходная
Хранение информации	project	Projects	-	структурная	входная/выходная
Хранение информации	myProject	var	-	структурная	входная/выходная
Хранение информации	user	Users	-	структурная	входная/выходная
Хранение информации	result	double	от 5E-324 до 1,7E+308	простая	входная/выходная

### 2.3.4 Тестирование

Был разработан отдельный модуль программы под названием «Project.Tests» для тестирования приложения, в котором реализован автоматический тест «UnitTestCalcFinalGrade» для тестирования вычисления итоговой оценки за проект.

Тест-кейс 1 – пользователь запускает приложение, переходит на экран регистрации по нажатию на соответствующую гиперссылку, заполняет все

необходимые поля на форме, после этого нажимает кнопку «Зарегистрироваться», если все поля были заполнены и пароль в поле «Повторите пароль» совпадает с паролем из поля «Пароль», то система должна выдать сообщение об успешной регистрации и добавить пользователя в базу данных.

Тест-кейс 2 – пользователь запускает приложение, открывается экран авторизации, пользователь заполняет свои логин и пароль в соответствующие поля, нажимает на кнопку «Войти», после этого система обращается к базе данных, если такой пользователь существует, то система выдаёт сообщение об успешной авторизации и открывает экран списка проектов.

Тест-кейс 3 – пользователь запускает приложение, проходит авторизацию и переходит на форму списка проектов, затем переходит к форме добавления нового проекта по нажатию кнопки «Добавить проект», заполняет поле «Название проекта» и нажимает на кнопку «Добавить новый проект», если пользователь заполнил поле с названием проекта, то система выдаст сообщение об добавление нового проекта, добавит его в базу данных и перейдёт на форму списка проектов, в котором появится созданный проект.

#### 2.3.5 Вызов и загрузка

Программа открывается с помощью файла «Project.exe.»

### 2.4 Руководство оператора

#### 2.4.1 Назначение программы

Программа предназначена для средних специальных учебных заведений и высших учебных заведений, в которых предусмотрена проектная деятельность в рамках учебного плана обучающихся в различных направлениях подготовки, именно она должна служить более комфортной и слаженной работе участников и руководителей в рамках проектной деятельности.



Разделение участников проектной деятельности на разные категории пользователей с разным уровнем доступа к системе позволяет каждому пользователю использовать особый функционал приложения.

#### 2.4.2 Условия выполнения

Программа предназначена для компьютеров на операционных системах семейства Windows, с установленной программой Microsoft SQL Server Management Studio 18 для работы с базой данных.

#### 2.4.3 Выполнение программы

##### 2.4.3.1 Работа пользователя

Пользователь может зарегистрироваться и авторизоваться в системе, изменить свой пароль, от уровня доступа пользователь может: создавать команды, добавлять вакансии в свою команду, окликаться на вакансии команд и другое.

##### 2.4.3.2 Работа администратора

Администратор системы имеет все возможности пользователя с самым высоким уровнем доступа, а также может производить администрирование базы данных прямо из приложения.

#### 2.4.4 Техника безопасности при работе с компьютером

Требования безопасности перед началом работы с компьютером:

- осмотреть и убедиться в исправности оборудования, электропроводки, при обнаружении неисправностей к работе не приступать;
- проверить освещение рабочего места, при необходимости принять меры к его нормализации;
- проверить наличие и надёжность защитного заземления оборудования;
- тщательно проветрить помещение с персональным компьютером и оргтехникой, убедиться, что микроклимат в помещении находится в допустимых пределах: температура воздуха в холодный период года от плюс 22 до плюс 24°C, в теплый период года от плюс 23 до плюс 25°C, относительная влажность воздуха от 40 до 60%;

- включить монитор и проверить стабильность и четкость изображения на экране, убедиться в отсутствии запаха дыма от компьютера и оргтехники.

Требования безопасности во время работы с компьютером:

- включайте и выключайте компьютер, оргтехнику только выключателями, запрещается проводить отключение вытаскиванием вилки из розетки;

- не допускать к компьютеру и оргтехнике посторонних лиц, которые не участвуют в работе;

- запрещается перемещать и переносить системный блок, монитор, принтер, любое оборудование, которое находится под напряжением;

- запрещается любое физическое вмешательство в устройство компьютера, оргтехники во время их работы;

- запрещается класть предметы на компьютер и оргтехнику;

- строго выполнять общие требования по электробезопасности и пожарной безопасности;

- самостоятельно разбирать и проводить ремонт электронной и электронно-механической части компьютера, периферийных устройств, оргтехники категорически запрещается;

- суммарное время непосредственной работы с персональным компьютером и другой оргтехникой в течение рабочего дня должно быть не более 6 часов, для педагогов - не более 4 часов в день;

- продолжительность непрерывной работы с персональным компьютером и другой оргтехникой без регламентированного перерыва не должна превышать 2-х часов;

- через каждый час работы следует делать перерыв продолжительностью 15 минут;

- во время регламентированных перерывов с целью снижения нервно-эмоционального напряжения, утомления зрительного анализатора, устранения влияния гиподинамии и гипокинезии следует выполнять комплексы упражнений

для глаз или организовывать физкультурные паузы;

- компьютер, любые его периферийные устройства, оргтехнику необходимо использовать в строгом соответствии с эксплуатационной документацией к ним;

- во время выполнения работы необходимо быть внимательным, не обращать внимание на посторонние вещи.

Требования безопасности после окончания работы с компьютером, принтером, ксероксом, сканером и другой оргтехникой:

- отключить компьютер от электросети;
- протереть внешнюю поверхность компьютера чистой влажной тканью;
- убрать рабочее место;
- тщательно проветрить помещение с персональным компьютером и другой оргтехникой.

### 3 Экономическая часть

#### 3.1 Расчёт основной и дополнительной заработной платы

Этапы и трудоёмкость выполнения работ представлены в таблице 3.

Этапы разработки	В часах	
	Руководитель	Техник
Постановка задачи	6	10
Выбор инструментальных средств	2	4
Разработка математической модели	4	10
Построение алгоритма	2	10
Программирование	8	60
Тестирование	4	20
Отладка	4	16
Написание пояснительной записки	10	30
Итого	40	160

Месячный оклад руководителя составляет 70000,00 руб., месячный оклад техника составляет 35000,00 руб. Разработка производится в апреле 2023 года при 40-часовой рабочей недели, отсюда количество рабочих часов составляет 160 часов.

Основная заработная плата работников рассчитывается по формуле (1), она зависит от оклада работника, трудоёмкости выполнения работ и количества рабочих часов.

Основная заработная плата:

$$ЗП_{\text{осн}} = О \times Т / Ч, \quad (1)$$

где  $О$  – месячный оклад сотрудника, руб.;

$Т$  – трудоёмкость выполнения работ сотрудника, ч.;

$Ч$  – количество рабочих часов, ч.

Основная заработная плата работников представлена в таблице 4.

Таблица 4 – Основная заработная плата работников В рублях

Должность	Заработная плата
Руководитель проект	17500,00
Техник	35000,00
Итого	52500,00

Дополнительная заработная плата составляет 10% от суммы основной заработной платы. Дополнительная заработная плата рассчитывается по формуле (2).

Дополнительная заработная плата:

$$ЗП_{\text{доп}} = ЗП_{\text{осн}}/100\% \times 10\%, \quad (2)$$

где  $ЗП_{\text{осн}}$  – основная заработная плата, руб.

Таким образом, дополнительная заработная плата составляет 5250,00 руб.

Отчисления в социальный фонд России и фонд обязательного медицинского страхования составляют 30% от суммы основной и дополнительной заработной платы. Отчисления в социальный фонд России и фонд обязательного медицинского страхования рассчитывается по формуле (3).

Отчисления в социальный фонд России и фонд обязательного медицинского страхования:

$$З_{\text{стр}} = (ЗП_{\text{осн}} + ЗП_{\text{доп}}) \times 30\% / 100\%, \quad (3)$$

где  $ЗП_{\text{осн}}$  – основная заработная плата, руб.;

$ЗП_{\text{доп}}$  – дополнительная заработная плата, руб.

Таким образом, отчисления в социальный фонд России и фонд обязательного медицинского страхования составляют 17325,00 руб.

В сумме основная, дополнительная заработные платы и отчисления в социальный фонд России составляют 75075,00 руб.

### 3.2 Расчёт стоимости материалов и программного обеспечения

Расчёт стоимости материалов представлен в таблице 5.

Таблица 5 – Расчёт стоимости материалов			В рублях
Наименование материала	Норма расхода, шт.	Цена за единицу	Сумма
Картридж	1	2775,00	2775,00
Флешка	1	350,00	350,00
Упаковка бумаги	1	325,00	325,00
Папка	1	150,00	150,00
Итого			3600,00

Общая стоимость материалов необходимых для реализации проекта составляет 3600,00 руб.

Расчёт стоимости программного обеспечения представлен в таблице 6.

Таблица 6 – Расчёт стоимости программного обеспечения			В рублях
Наименование программного обеспечения	Норма установки, шт.	Цена за единицу	Сумма
Microsoft Windows 10 Home	1	4000,00	4000,00
Microsoft Office 365	1	1000,00	1000,00
Microsoft Visual Studio 2022	1	в свободном доступе	в свободном доступе
Microsoft SQL Server Tools 18	1	в свободном доступе	в свободном доступе
Итого			6000,00

Общая стоимость программного обеспечения составляет 6000,00 руб.

Общие расходы на эксплуатацию составляют сумму основной и дополнительной заработной платы, отчислений в социальный фонд России, стоимость материалов и программного обеспечения. И составляют 84675,00 руб.

### 3.3 Расчёт накладных расходов

Накладные расходы составляют 10% от эксплуатационных расходов. Накладные расходы рассчитываются по формуле (4).

Накладные расходы:

$$H_p = P_{\text{экс}} \times 10\% / 100\%, \quad (4)$$

где  $P_{\text{экс}}$  – величина расходов на эксплуатацию, руб.

Таким образом, накладные расходы составляют 8467,50 руб.

### 3.4 Составление и расчёт цены реализации программного продукта

Смета затрат на программный продукт представлена в таблице 7.

Таблица 7 – Смета затрат на программный продукт

Статьи расходов	Сумма, руб.	Структура затрат, %
Основная и дополнительная заработная плата	57750,00	62,00
Отчисления в социальный фонд России и ФОМС	17325,00	18,60
Стоимость материалов	3600,00	3,86
Стоимость программного обеспечения	6000,00	6,44
Итого расходы на эксплуатацию	84675,00	90,91
Накладные расходы	8467,50	9,09
Итого	93142,50	100,00

Таким образом, общая сумма затрат на разработку программного продукта составляет 93142,50 руб.

Примем уровень рентабельности за 35% от общих затрат. Оптовая цена рассчитывается по формуле (5).

Оптовая цена:

$$Ц_{\text{опт}} = З \times (1 + P), \quad (5)$$

где  $З$  – общие затраты на разработку программного продукта, руб.;

$P$  – уровень рентабельности проекта, коэффициент.

Таким образом, оптовая цена составит 125742,37 руб.

Прибыли от продажи рассчитывается по формуле (6).

Прибыль от продажи:

$$П = Ц_{\text{опт}} - З, \quad (6)$$

где  $Ц_{\text{опт}}$  – оптовая цены программного продукта, руб.;

$З$  – общие затраты на разработку программного продукта, руб.

Таким образом, можно сделать вывод, что данный программный продукт является прибыльным. Прибыль от продажи программного продукта составит 32599,87 руб.

Структура затрат на программный продукт представлена на рисунке А.12.



В результате дипломного проектирования была разработана информационная система для работы проектной деятельности.

На этапе обоснования необходимости разработки информационной системы были сформулированы цели и задачи дипломного проектирования, осуществлено знакомство с предметной областью, произведён обзор и анализ существующих программных систем.

На этапе разработки и утверждения технического задания были сформированы требования к системе, разработано технико-экономическое обоснование разработки системы, определены стадии, этапы и сроки разработки информационной системы и документации к ней, выбраны инструментальные средства для разработки информационной системы, составлено технико-математическое описание задачи.

На этапе проектирования информационной системы были разработаны wireframe эскизы пользовательского интерфейса, спроектированы модели базы данных средствами программы MS Visio 2016, изложены методы защиты данных, осуществлено описание алгоритмов работы программы с помощью блок-схем, составлены тест-кейсы для тестирования работы программы, определены структуры входных и выходных данных.

На этапе разработки информационной системы была создана база данных средствами языка программирования реляционных баз данных SQL в СУБД MS SQL Server, разработано desktop-приложение для работы с базой данных в интегрированной среде разработки MS Visual Studio 2022 средствами языка программирования C#, а также произведена отладка и тестирование системы.

На этапе разработки программной документации информационной системы были подготовлены следующие эксплуатационные документы: руководство пользователя, руководство администратора, инструкция по техники безопасности при работе на компьютере, инструкция по процессу установки программы, описание требований к системе.

Также были разработаны и помещены в приложения к пояснительной записке графический и иной материал дипломного проектирования, а именно фрагменты кода программы, скрипт базы данных, разработанные эскизы окон приложения, окна приложения, схемы алгоритмов, схемы данных, функциональные диаграммы.

В данной дипломной работе были решены все поставленные во введении задачи, начиная со знакомства с предметной областью и заканчивая созданием информационной системы для работы проектной деятельности.

Таким образом, можно сделать вывод, что задание на дипломное проектирование выполнено в полном объёме.

1. ГОСТ 2.105–2019. Общие требования к текстовым документам. – Москва: Стандартинформ, 2021. – 33 с.
2. ГОСТ 19.701–90. Схемы алгоритмов, программ, данных и систем. Обозначения условные и правила выполнения. – Москва: Стандартинформ, 2010. – 22 с.
3. Васильев, А.Н. Программирование на C# для начинающих. Основные сведения. / А.Н. Васильев. – Москва: Эксмо, 2018. – 592 с.
4. Васильев, А.Н. Программирование на C# для начинающих. Особенности языка. / А.Н. Васильев. – Москва: Эксмо, 2019. – 528 с.
5. Дадян, Э.Г. Данные: хранение и обработка. / Э.Г. Дадян. – Москва: Инфра-М, 2020. – 205 с.
6. Короткевич, Д.А. SQL Server. Настройка и оптимизация для профессионалов. / Д.А. Короткевич. – Москва: Прогресс книга, 2023. – 512 с.
7. Логачев, М.С. Информационные системы и программирование. Специалист по информационным системам. Выпускная квалификационная работа. / М.С. Логачев. – Москва: Инфра-М, 2021. – 576 с.
8. Маркин, А. В. Программирование на SQL: учебное пособие для среднего профессионального образования. / А. В. Маркин. – Москва: Юрайт, 2019. – 435 с.
9. Молдованова, О.В. Информационные системы и базы данных. Учебное пособие. / О.В. Молдованова. – Саратов: Профобразование, 2022. – 184 с.
10. Павлов, Л. А. Структуры и алгоритмы обработки данных: учебник. / Л. А. Павлов, Н. В. Первова. – Санкт-Петербург: Лань, 2020. – 256 с.
11. Полное руководство по языку программирования C# 11 и платформе .NET 7. – URL: <https://metanit.com/sharp/tutorial/>
12. Уроки по C# и платформе .NET Framework. – URL: <https://professorweb.ru/>

СУБД - Система управления базами данных.

ADO.NET - ActiveX Data Object для .NET.

WPF - Windows Presentation Foundation.

CLR - Common Language Runtime.

SQL - Structured Query Language.

OLE DB - Object Linking and Embedding, Database.

GDI - Graphics Device Interface.

XAML - Extensible Application Markup Language.

XML -Extensible Markup Language.

CLI - Common Language Infrastructure.

AMD - Advanced Micro Devices.

ОЗУ - Оперативное запоминающее устройство.

MS – Microsoft.

IDE - Integrated Development Environment.

MD5 - Message Digest 5.

LINQ - Language Integrated Query.

# Приложение А (обязательное)



Рисунок А.1 – Диаграмма декомпозиция IDEF0

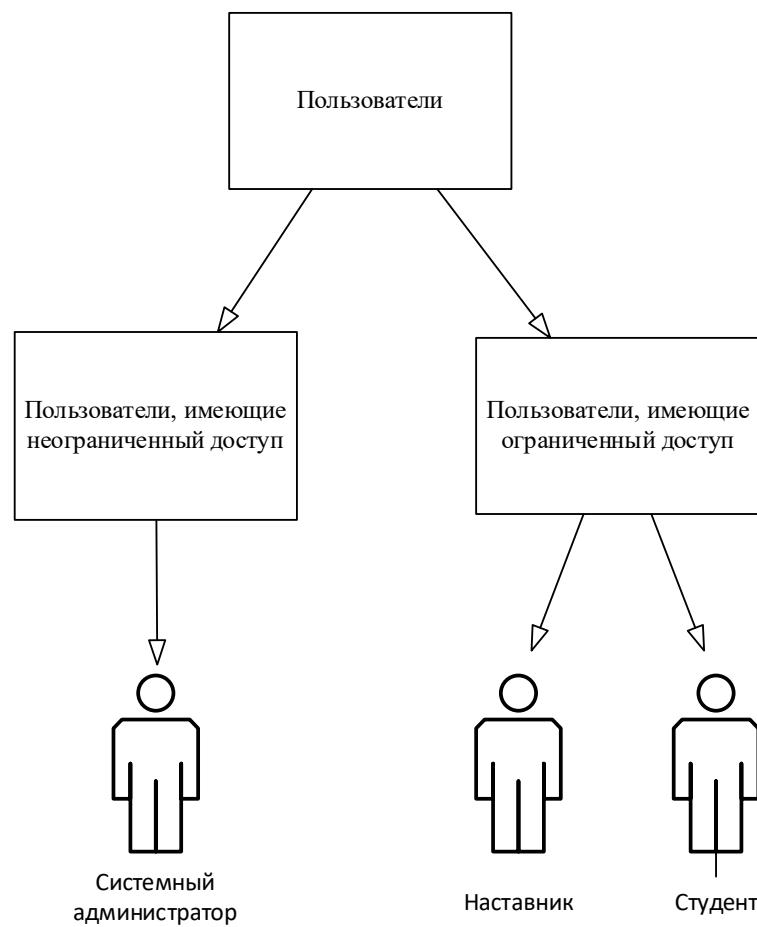


Рисунок А.2 – Группы пользователей системы

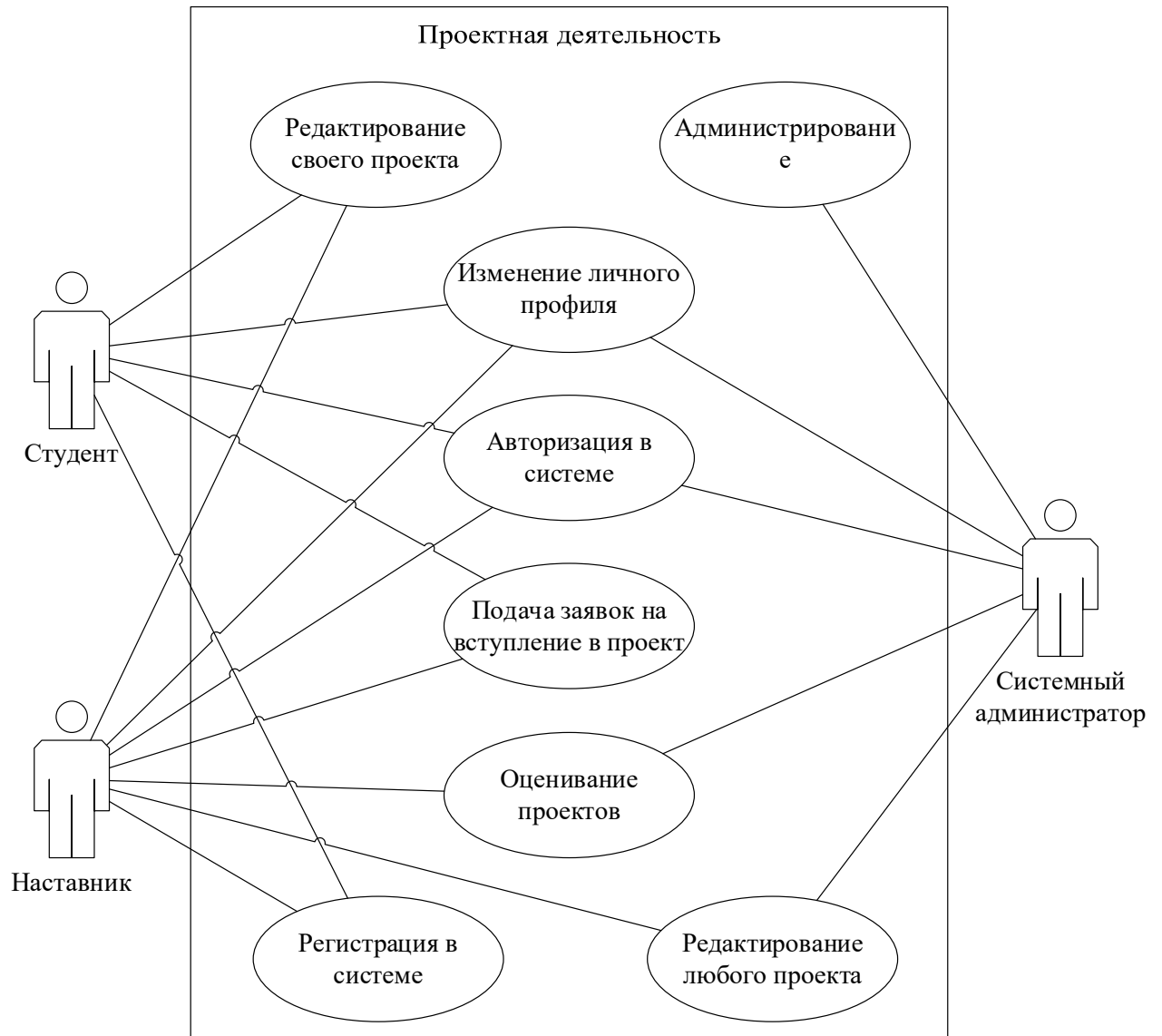


Рисунок А.3 – Диаграмма вариантов использования

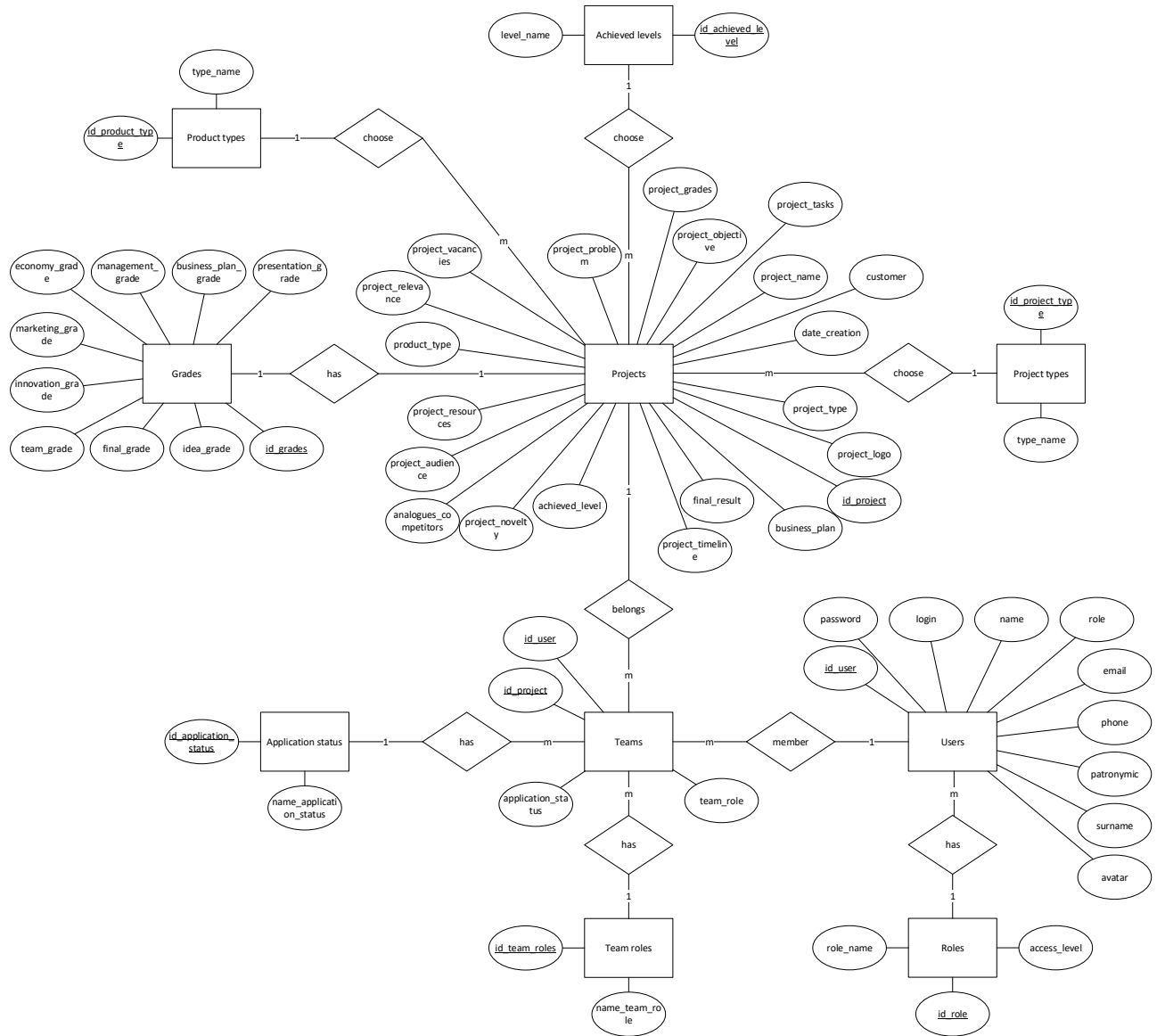


Рисунок А.4 – ER модель базы данных

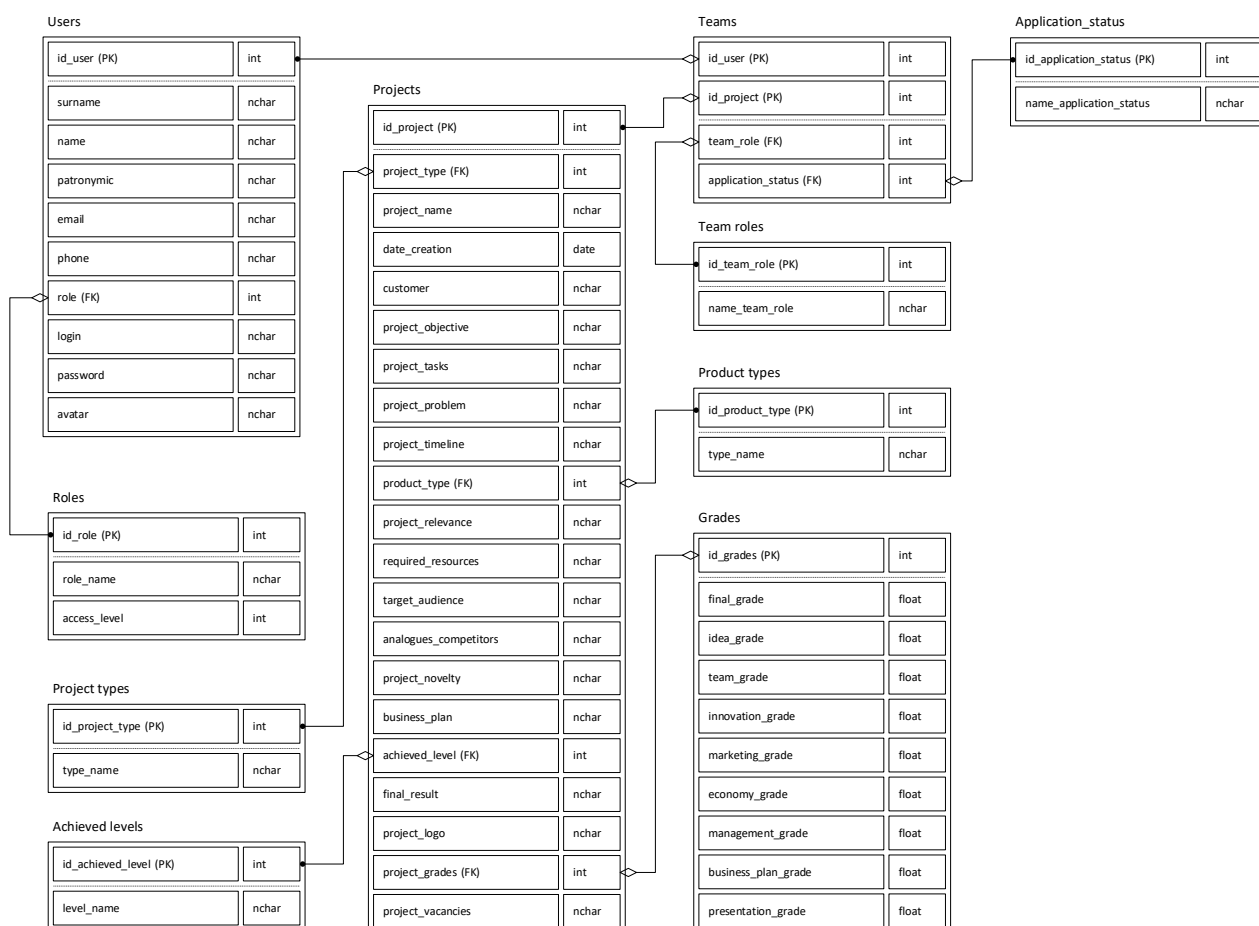


Рисунок А.5 – Реляционная модель базы данных



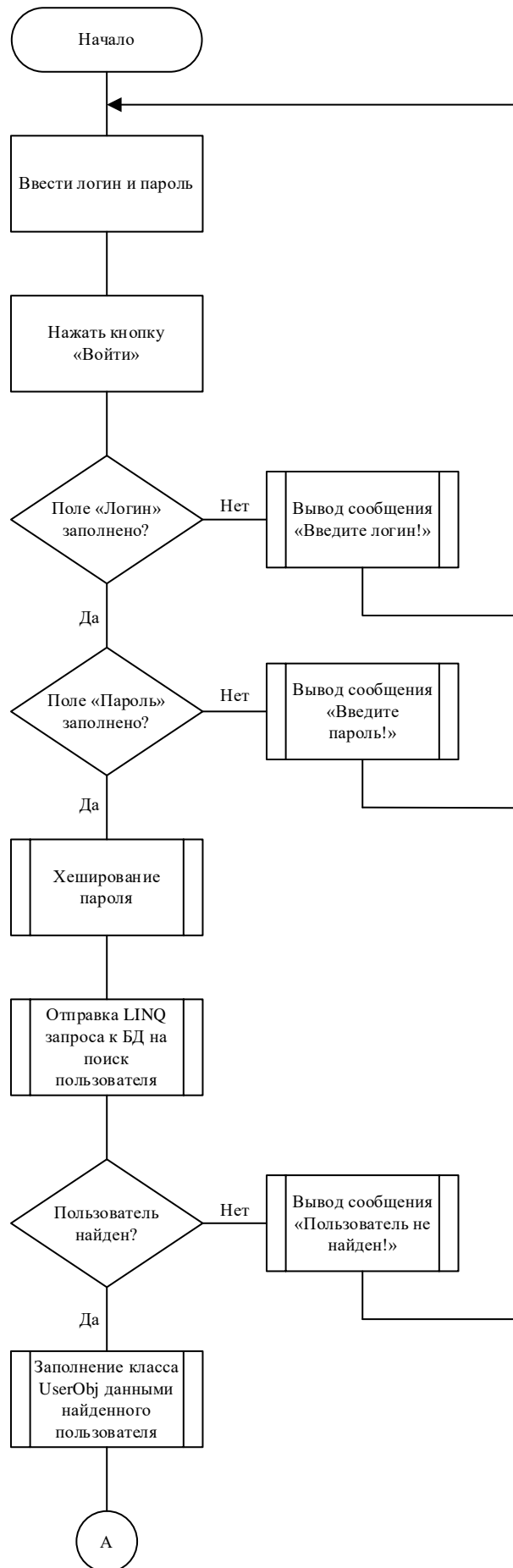


Рисунок А.6 – Блок-схема авторизации в системе



Рисунок А.7 – Блок-схема авторизации в системе

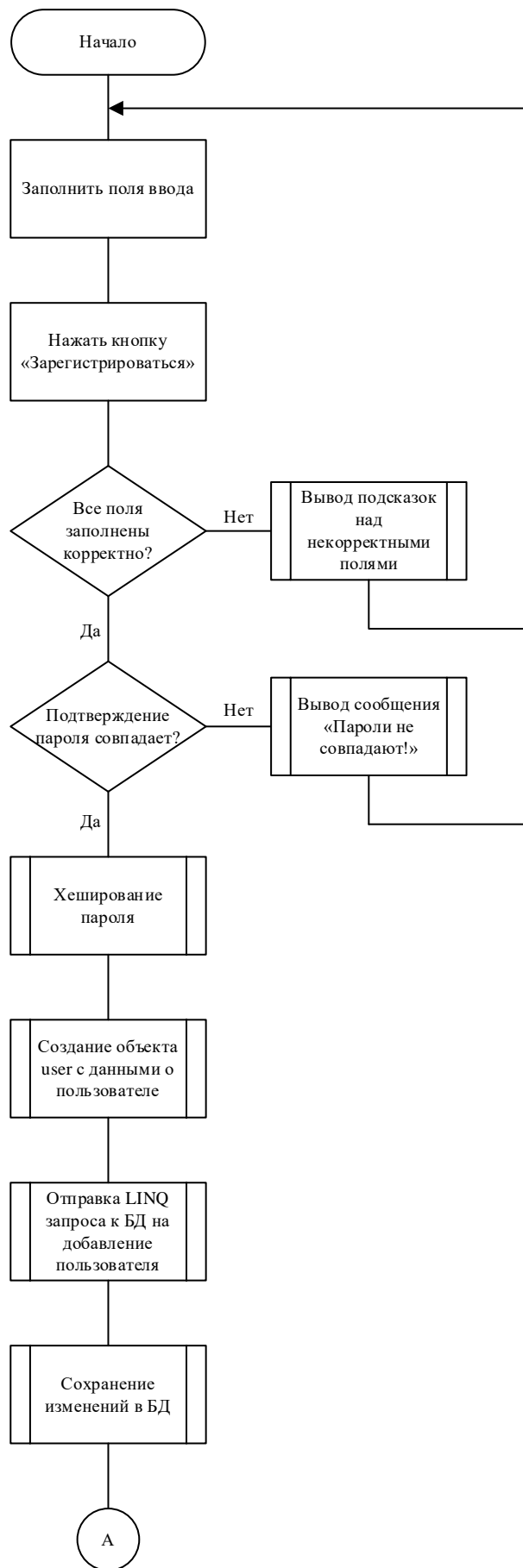


Рисунок А.8 – Блок-схема регистрации в системе

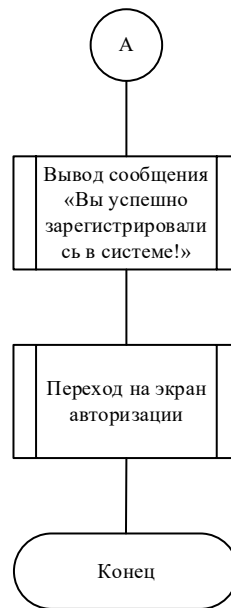


Рисунок А.9 – Блок-схема регистрации в системе

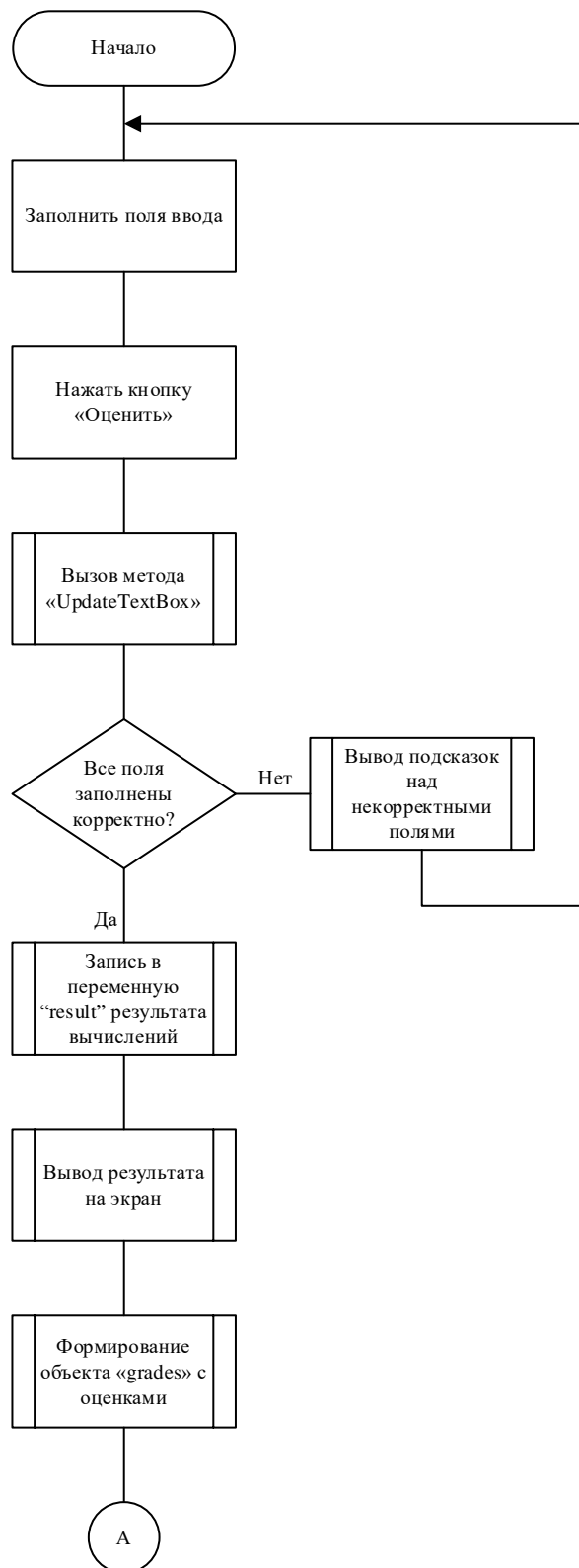


Рисунок А.10 – Блок-схема выставление оценки проекту

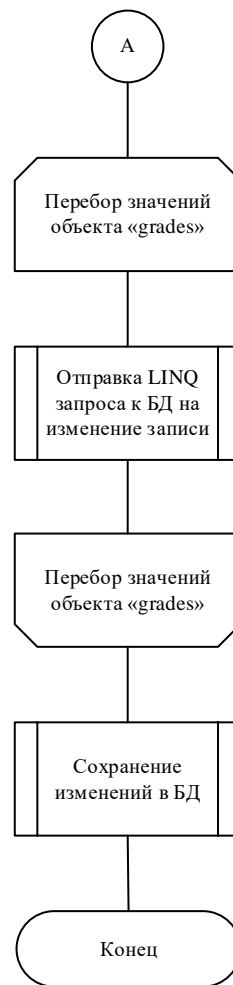


Рисунок А.11 – Блок-схема выставление оценки проекту

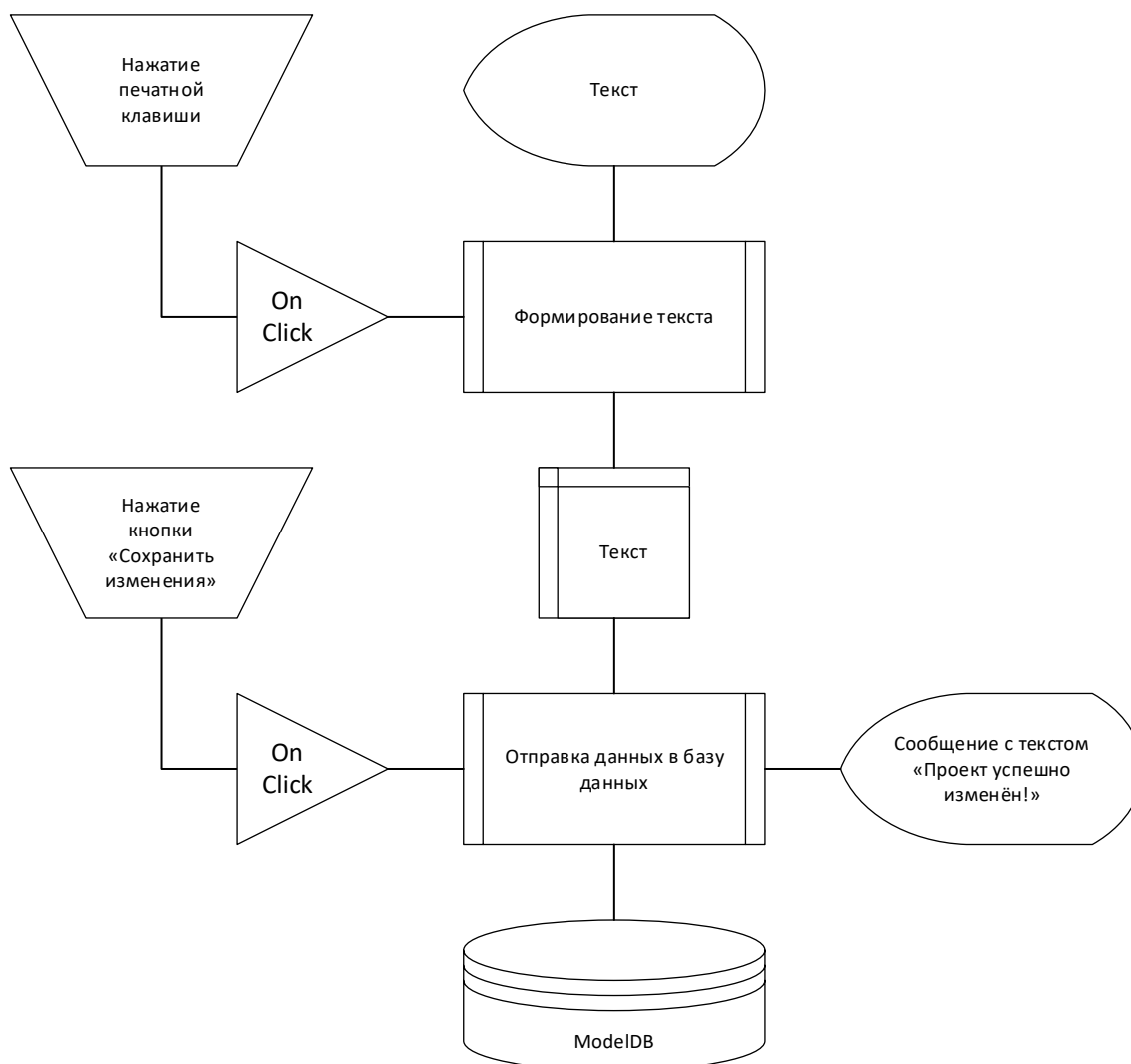


Рисунок А.12 – Схема взаимодействий метода сохранения изменений о проекте

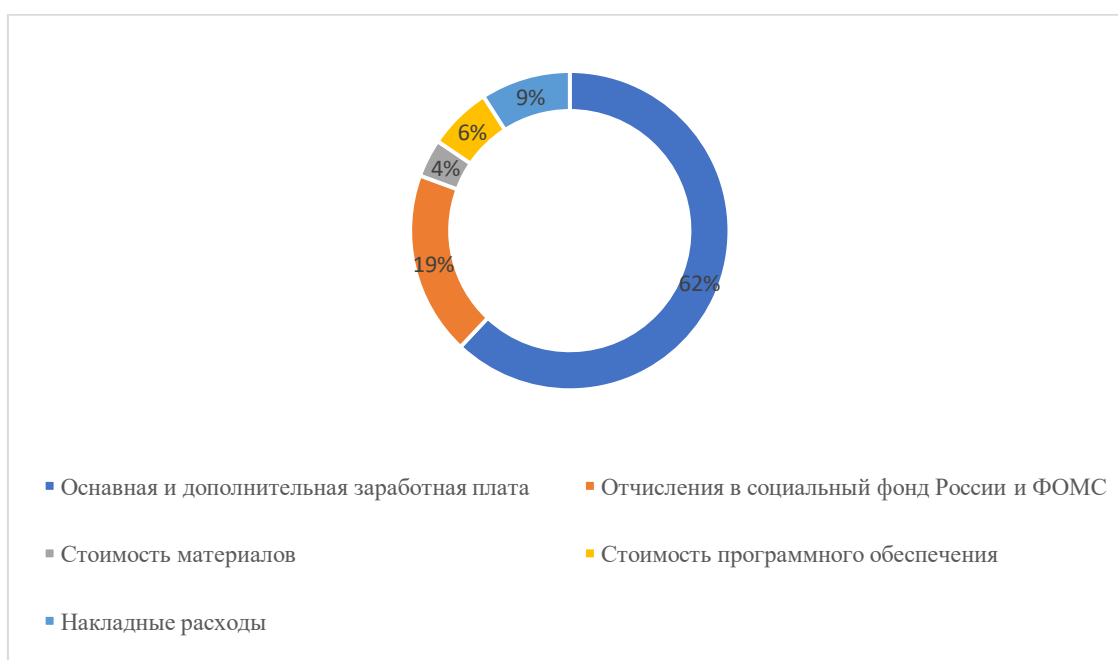


Рисунок А.13 – Структура затрат на программный продукт

## Приложение Б (обязательное)

### Б.1 – Скрипт SQL для создания таблицы «Роли в системе»

```
CREATE TABLE [dbo].[Roles](
    [id_role] [int] IDENTITY(1,1) NOT NULL,
    [role_name] [nvarchar](max) NOT NULL,
    [access_level] [int] NOT NULL,
    CONSTRAINT [PK_Roles] PRIMARY KEY CLUSTERED
(
    [id_role] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
```

### Б.2 – Скрипт SQL для создания таблицы «Роли в команде»

```
CREATE TABLE [dbo].[Team_roles](
    [id_team_role] [int] IDENTITY(1,1) NOT NULL,
    [name_team_role] [nvarchar](max) NOT NULL,
    CONSTRAINT [PK_Team_roles] PRIMARY KEY CLUSTERED
(
    [id_team_role] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
```

### Б.3 – Скрипт SQL для создания таблицы «Состав команды»

```
CREATE TABLE [dbo].[Teams](
    [id_user] [int] NOT NULL,
    [id_project] [int] NOT NULL,
    [team_role] [int] NOT NULL,
    [application_status] [int] NOT NULL,
    CONSTRAINT [PK_Teams] PRIMARY KEY CLUSTERED
(
    [id_user] ASC,
    [id_project] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
```



#### Б.4 – Скрипт SQL для создания таблицы «Пользователи»

```
CREATE TABLE [dbo].[Users] (
    [id_user] [int] IDENTITY(1,1) NOT NULL,
    [surname] [nvarchar](50) NOT NULL,
    [name] [nvarchar](50) NOT NULL,
    [patronymic] [nvarchar](50) NULL,
    [email] [nvarchar](50) NOT NULL,
    [phone] [nvarchar](16) NOT NULL,
    [role] [int] NOT NULL,
    [login] [nvarchar](50) NOT NULL,
    [password] [nvarchar](50) NOT NULL,
    [avatar] [nvarchar](max) NULL,
    CONSTRAINT [PK_Users] PRIMARY KEY CLUSTERED
(
    [id_user] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
```

#### Б.5 – Скрипт SQL для создания таблицы «Достигнутые уровни»

```
CREATE TABLE [dbo].[Achieved_levels] (
    [id_achieved_level] [int] IDENTITY(1,1) NOT NULL,
    [level_name] [nvarchar](max) NOT NULL,
    CONSTRAINT [PK_Achieved_levels] PRIMARY KEY CLUSTERED
(
    [id_achieved_level] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
```

#### Б.6 – Скрипт SQL для создания таблицы «Типы продуктов»

```
CREATE TABLE [dbo].[Product_types] (
    [id_product_type] [int] IDENTITY(1,1) NOT NULL,
    [type_name] [nvarchar](max) NOT NULL,
    CONSTRAINT [PK_Product_types] PRIMARY KEY CLUSTERED
```

```
(
    [id_product_type] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
```

## Б.7 – Скрипт SQL для создания таблицы «Типы проектов»

```
CREATE TABLE [dbo].[Project_types](
    [id_project_type] [int] IDENTITY(1,1) NOT NULL,
    [type_name] [nvarchar](max) NOT NULL,
    CONSTRAINT [PK_Project_types] PRIMARY KEY CLUSTERED
(
    [id_project_type] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
```

## Б.8 – Скрипт SQL для создания таблицы «Проекты»

```
CREATE TABLE [dbo].[Projects](
    [id_project] [int] IDENTITY(1,1) NOT NULL,
    [project_type] [int] NULL,
    [project_name] [nvarchar](max) NOT NULL,
    [date_creation] [date] NOT NULL,
    [customer] [nvarchar](max) NULL,
    [project_objective] [nvarchar](max) NULL,
    [project_tasks] [nvarchar](max) NULL,
    [project_problem] [nvarchar](max) NULL,
    [project_timeline] [nvarchar](max) NULL,
    [product_type] [int] NULL,
    [project_relevance] [nvarchar](max) NULL,
    [required_resources] [nvarchar](max) NULL,
    [target_audience] [nvarchar](max) NULL,
    [analogues_competitors] [nvarchar](max) NULL,
    [project_novelty] [nvarchar](max) NULL,
    [business_plan] [nvarchar](max) NULL,
    [achieved_level] [int] NULL,
    [final_result] [nvarchar](max) NULL,
    [project_logo] [nvarchar](max) NULL,
```

```

        [project_grades] [int] NULL,
        [project_vacancies] [nvarchar](max) NULL,
    CONSTRAINT [PK_Project] PRIMARY KEY CLUSTERED
    (
        [id_project] ASC
    )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
    ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]

```

## Б.9 – Скрипт SQL для создания таблицы «Оценки»

```

CREATE TABLE [dbo].[Grades](
    [id_grade] [int] IDENTITY(1,1) NOT NULL,
    [final_grade] [float] NOT NULL,
    [idea_grade] [float] NOT NULL,
    [team_grade] [float] NOT NULL,
    [innovation_grade] [float] NOT NULL,
    [marketing_grade] [float] NOT NULL,
    [economy_grade] [float] NOT NULL,
    [management_grade] [float] NOT NULL,
    [business_plan_grade] [float] NOT NULL,
    [presentation_grade] [float] NOT NULL,
    CONSTRAINT [PK_Grades] PRIMARY KEY CLUSTERED
    (
        [id_grade] ASC
    )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
    ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

```

## Б.10 – Скрипт SQL для создания таблицы «Статусы заявок»

```

CREATE TABLE [dbo].[Application_status](
    [id_application_status] [int] NOT NULL,
    [name_application_status] [nvarchar](50) NOT NULL,
    CONSTRAINT [PK_Application_status] PRIMARY KEY CLUSTERED
    (
        [id_application_status] ASC
    )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
    ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

```

```
private void btnAuthorization_Click(object sender, RoutedEventArgs)
{
    try
    {
        ShowPassCheckBox.IsChecked = true;
        ShowPassCheckBox.IsChecked = false;

        //Заполнен ли логин
        if (txtbxLogin.Text.Length > 0)
        {
            //Заполнен ли пароль
            if (txtbxPassword.Password.Length > 0)
            {
                //Шифрования пароля с помощью метода
                string result =
                MessageDigest5.hashing(txtbxPassword.Password);

                //Поиск пользователя в БД
                Users user = ConnectObj.conObj.Users.Where(x => x.login
                == txtbxLogin.Text && x.password == result).FirstOrDefault();
                if (user != null)
                {
                    UserObj.id_user = user.id_user;
                    UserObj.surname = user.surname;
                    UserObj.name = user.name;
                    UserObj.patronymic = user.patronymic;
                    UserObj.role = user.role;
                    UserObj.email = user.email;
                    UserObj.phone = user.phone;
                    UserObj.login = user.login;
                    UserObj.password = user.password;
                    UserObj.avatar = user.avatar;

                    //Появление элементов в шапке
                    CheckAuthorization.CheckAuto = true;
                }
            }
        }
    }
}
```

```

//Переход на экран проектов или панели администратора
if (UserObj.role == 1)
{
    FrameObj.frameObj.Navigate(new
PageAdminPanel());
}
else
{
    FrameObj.frameObj.Navigate(new PageProject());
}
MessageBox.Show("Успешный вход!", "Уведомление",
MessageBoxButton.OK, MessageBoxImage.Information);
}
else
{
    MessageBox.Show("Пользователь не найден!",
"Уведомление", MessageBoxButton.OK, MessageBoxImage.Information);
}
}
else
{
    MessageBox.Show("Введите пароль!", "Уведомление",
MessageBoxButton.OK, MessageBoxImage.Information);
}
}
else
{
    MessageBox.Show("Введите логин!", "Уведомление",
MessageBoxButton.OK, MessageBoxImage.Information);
}
}
catch (Exception ex)
{
    MessageBox.Show($"{ex.Message}", "Ошибка", MessageBoxButton.OK,
MessageBoxImage.Error);
}
}

```

## В.2 – Регистрация пользователя

```
private void btnRegistration_Click(object sender, RoutedEventArgs e)
```

```

{
    try
    {
        ShowPassCheckBox.IsChecked = true;
        ShowPassCheckBox.IsChecked = false;

        //Проверка валидации
        if (!Validation.GetHasError(txtbxSurname) &&
            !Validation.GetHasError(txtbxName) && !Validation.GetHasError(txtbxPatronymic)
            && !Validation.GetHasError(txtbxEmail)
            && !Validation.GetHasError(txtbxPhone) &&
            !Validation.GetHasError(txtbxLogin) && !Validation.GetHasError(txtbxPassword) &&
            !Validation.GetHasError(txtbxCheckPass))
        {
            //Совпадают ли пароли
            if (txtbxPassText.Text == txtbxCheckPassText.Text)
            {
                //Шифрования пароля с помощью метода
                string result =
                MessageDigest5.hashing(txtbxPassText.Text);

                //Добавление нового пользователя в БД
                Users user = new Users()
                {
                    surname = txtbxSurname.Text,
                    name = txtbxName.Text,
                    patronymic = txtbxPatronymic.Text,
                    email = txtbxEmail.Text,
                    phone = txtbxPhone.Text,
                    role = cmbbxRole.SelectedIndex + 2,
                    login = txtbxLogin.Text,
                    password = result,
                    avatar = "\\Resources\\AvatatDefault.png"
                };
                ConnectObj.conObj.Users.Add(user);
                ConnectObj.conObj.SaveChanges();
                MessageBox.Show("Вы успешно зарегистрировались в
системе!", "Уведомление", MessageBoxButton.OK, MessageBoxImage.Information);

                //Переход на прошлую страницу
                FrameObj.frameObj.GoBack();
            }
        }
    }
    else

```

```

        {
            MessageBox.Show("Пароли не совпадают!", "Уведомление",
MessageBoxButton.OK, MessageBoxImage.Information);
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show($"{ex.Message}", "Ошибка", MessageBoxButton.OK,
MessageBoxImage.Error);
    }
}

```

### В.3 – Вывод списка проектов

```

public void UpdateDataProject(object sender, object e)
{
    try
    {
        if (Sort.SelectedIndex == 0)
        {
            if (Filtr.SelectedIndex == 0)
            {
                ListProject.ItemsSource =
ConnectObj.conObj.Projects.Where(x =>
x.project_name.StartsWith(Search.Text)).OrderByDescending(x
x.date_creation).ToList();
            }
            else if (Filtr.SelectedIndex > 0)
            {
                ListProject.ItemsSource =
ConnectObj.conObj.Projects.Where(x => x.project_type == Filtr.SelectedIndex &&
x.project_name.StartsWith(Search.Text)).
OrderByDescending(x => x.date_creation).ToList();
            }
        }
        else if (Sort.SelectedIndex == 1)
        {
            if (Filtr.SelectedIndex == 0)
            {

```

```

        ListProject.ItemsSource =
ConnectObj.conObj.Projects.Where(x =>
x.project_name.StartsWith(Search.Text)).OrderBy(x => x.date_creation).ToList();
    }
    else if (Filtr.SelectedIndex > 0)
    {
        ListProject.ItemsSource =
ConnectObj.conObj.Projects.Where(x => x.project_type == Filtr.SelectedIndex &&
x.project_name.StartsWith(Search.Text)).
        OrderBy(x => x.date_creation).ToList();
    }
}
}
catch (Exception ex)
{
    MessageBox.Show($"{ex.Message}", "Ошибка", MessageBoxButton.OK,
    MessageBoxImage.Error);
}
}

```

## В.4 – Создание нового проекта

```

private void btnAddProject_Click(object sender, RoutedEventArgs e)
{
    try
    {
        if (txtbxProjectName.Text.Length > 0)
        {
            //Добавление записи с оценками в БД
            Grades grade = new Grades()
            {
                final_grade = 0,
                idea_grade = 0,
                team_grade = 0,
                innovation_grade = 0,
                marketing_grade = 0,
                economy_grade = 0,
                management_grade = 0,
                business_plan_grade = 0,
                presentation_grade = 0
            };
        }
    }
}

```



```

ConnectObj.conObj.Grades.Add(grade);

//Добавление проекта в БД
Projects project = new Projects()
{
    project_name = txtbxProjectName.Text,
    date_creation = DateTime.Today,
    project_logo = "\\Resources\\picture.png",
    project_grades = grade.id_grade
};

ConnectObj.conObj.Projects.Add(project);
ConnectObj.conObj.SaveChanges();

MessageBox.Show("Вы успешно создали новый проект!",
"Уведомление", MessageBoxButton.OK, MessageBoxImage.Information);

FrameObj.frameObj.GoBack();
}
else
{
    MessageBox.Show("Укажите название проекта!",
"Уведомление", MessageBoxButton.OK, MessageBoxImage.Information);
}

}
catch (Exception ex)
{
    MessageBox.Show($"{ex.Message}", "Ошибка", MessageBoxButton.OK,
MessageBoxImage.Error);
}
}

```

## В.5 – Редактирование проекта

```

private void btnSaveChanges_Click(object sender, RoutedEventArgs e)
{
    try
    {
        IEnumerable<Projects> projects =
ConnectObj.conObj.Projects.Where(x => x.id_project ==
ProjectObj.id_project).AsEnumerable().
Select(x =>
{
    x.project_name = txtbxProjectName.Text;

```

```

        x.customer = txtbxCustomer.Text;
        x.project_objective = txtbxProjectObjective.Text;
        x.project_tasks = txtbxProjectTasks.Text;
        x.project_problem = txtbxProjectProblem.Text;
        x.project_timeline = txtbxProjectTimeline.Text;
        x.project_relevance = txtbxProjectRelevance.Text;
        x.required_resources = txtbxRequiredResources.Text;
        x.target_audience = txtbxTargetAudience.Text;
        x.analogues_competitors =
txtbxAnaloguesCompetitors.Text;
        x.project_novelty = txtbxProjectNovelty.Text;
        x.business_plan = txtbxBusinessPlan.Text;
        x.final_result = txtbxFinalResult.Text;
        x.project_vacancies = txtbxProjectVacancies.Text;
        if (cmbbxProjectType.SelectedIndex != -1)
        {
            x.project_type = cmbbxProjectType.SelectedIndex + 1;
        }
        if (cmbbxProductType.SelectedIndex != -1)
        {
            x.prodoct_type = cmbbxProductType.SelectedIndex + 1;
        }
        if (cmbbxAchievedLevel.SelectedIndex != -1)
        {
            x.achieved_level = cmbbxAchievedLevel.SelectedIndex +
1;
        }
        return x;
    });
    foreach (Projects project in projects)
    {
        ConnectObj.conObj.Entry(project).State =
System.Data.Entity.EntityState.Modified;
    }
    ConnectObj.conObj.SaveChanges();
    MessageBox.Show("Проект успешно изменен!", "Уведомление",
MessageBoxButton.OK, MessageBoxImage.Information);
    FrameObj.frameObj.GoBack();
}
catch (Exception ex)
{
    MessageBox.Show($"{ex.Message}", "Ошибка", MessageBoxButton.OK,
MessageBoxImage.Error);

```

```

    }
}

```

## В.6 – Удаление проекта

```

private void btnDeleteProject_Click(object sender, RoutedEventArgs e)
{
    try
    {
        if (MessageBox.Show("Вы уверены, что хотите удалить этот проект?",
            "Сообщение",
            MessageBoxButton.YesNo,
            MessageBoxImage.Question) == MessageBoxResult.Yes)
        {
            Projects project = ConnectObj.conObj.Projects.Where(x =>
x.id_project == ProjectObj.id_project).FirstOrDefault();
            ConnectObj.conObj.Projects.Remove(project);

            ConnectObj.conObj.SaveChanges();
            MessageBox.Show("Вы успешно удалили проект!", "Уведомление",
MessageBoxButton.OK, MessageBoxImage.Information);
            FrameObj.frameObj.Navigate(new PageProject());
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show($"{ex.Message}", "Ошибка", MessageBoxButton.OK,
MessageBoxImage.Error);
    }
}

```

Приложение Г  
(обязательное)

The sketch shows a web browser window with a standard title bar (close, maximize, and zoom buttons). The main content area is divided into two sections. The top section is a header bar with a gray background on the left and a white background on the right. The bottom section has a gray background. In the center of the gray background is a white rectangular box containing the registration form. The form consists of four horizontal input fields, each preceded by a short horizontal line representing a label. Below these fields is a single, wider button with a horizontal line in the center, representing a submit button.

Рисунок Г.1 – Эскиз экрана регистрации пользователя

The sketch shows a web browser window with a standard title bar (close, maximize, and zoom buttons). The main content area is divided into two sections. The top section is a header bar with a gray background on the left and a white background on the right. The bottom section has a gray background. In the center of the gray background is a white rectangular box containing the authorization form. The form consists of two horizontal input fields, each preceded by a short horizontal line representing a label. Below these fields is a single, wider button with a horizontal line in the center, representing a submit button. Below the button is a horizontal line, likely representing a link for password recovery or registration.

Рисунок Г.2 – Эскиз экрана авторизации пользователя

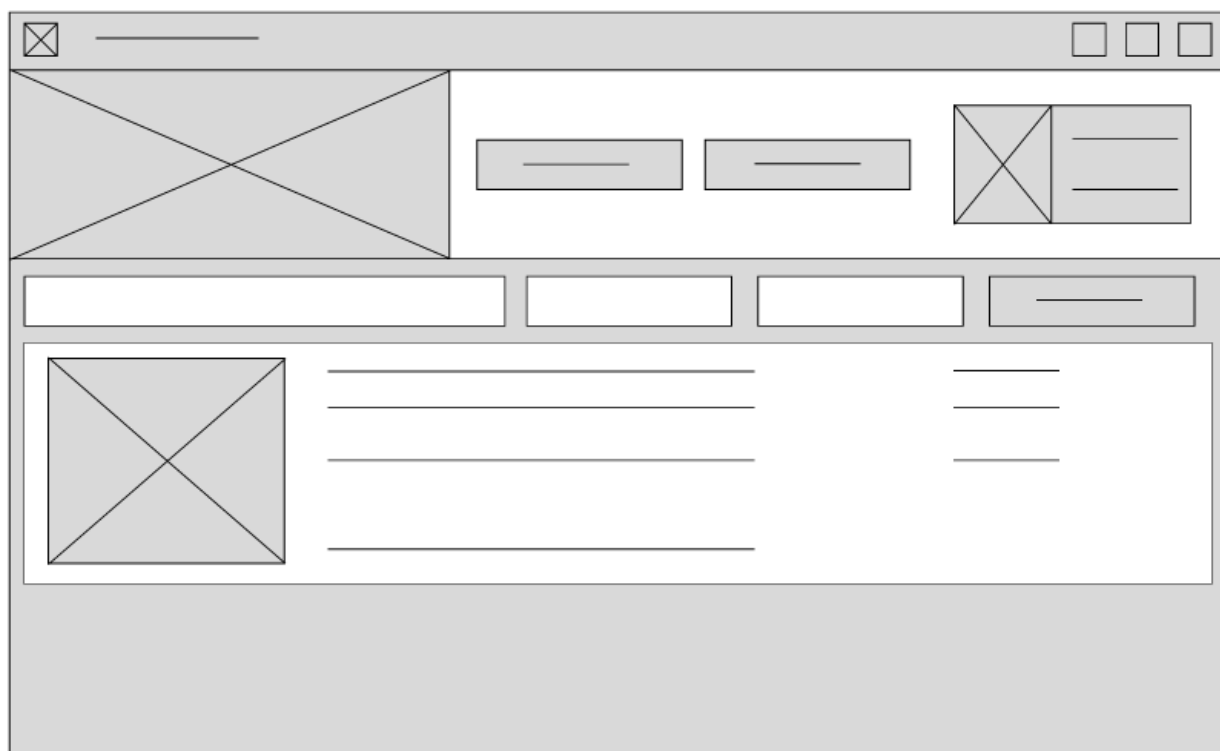


Рисунок Г.3 – Эскиз экрана списка проектов

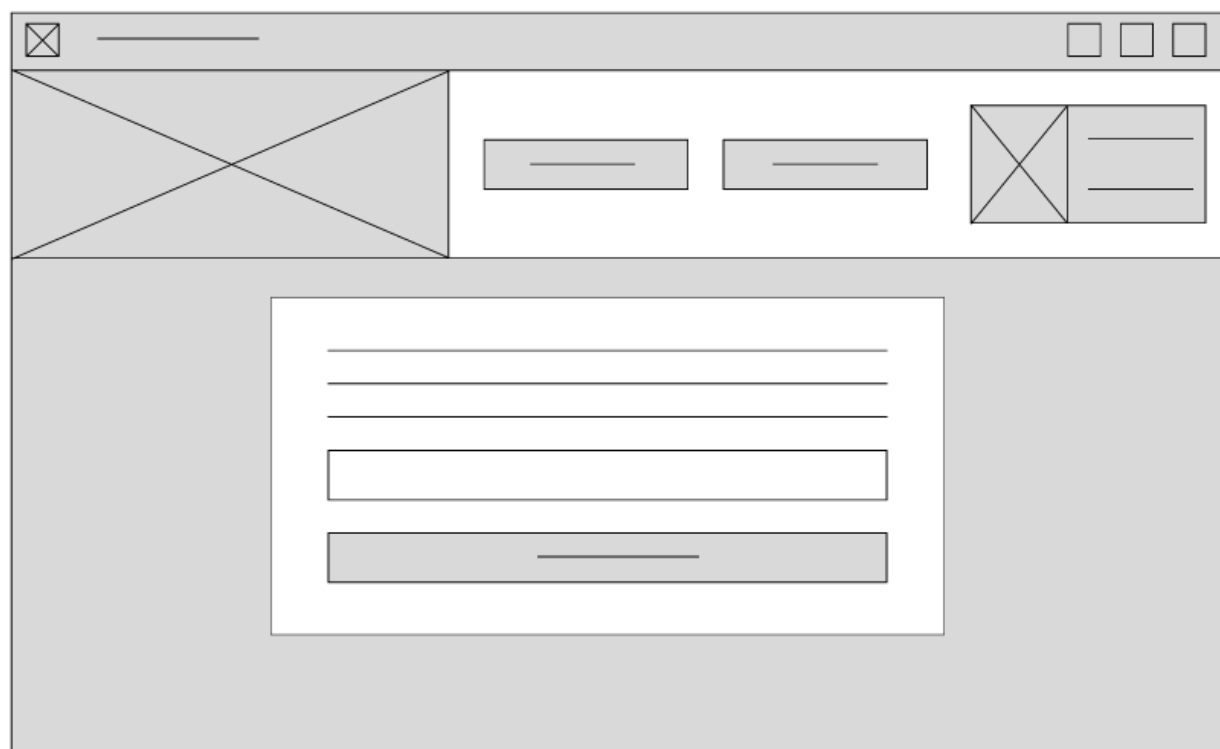


Рисунок Г.4 – Эскиз экрана добавления нового проекта

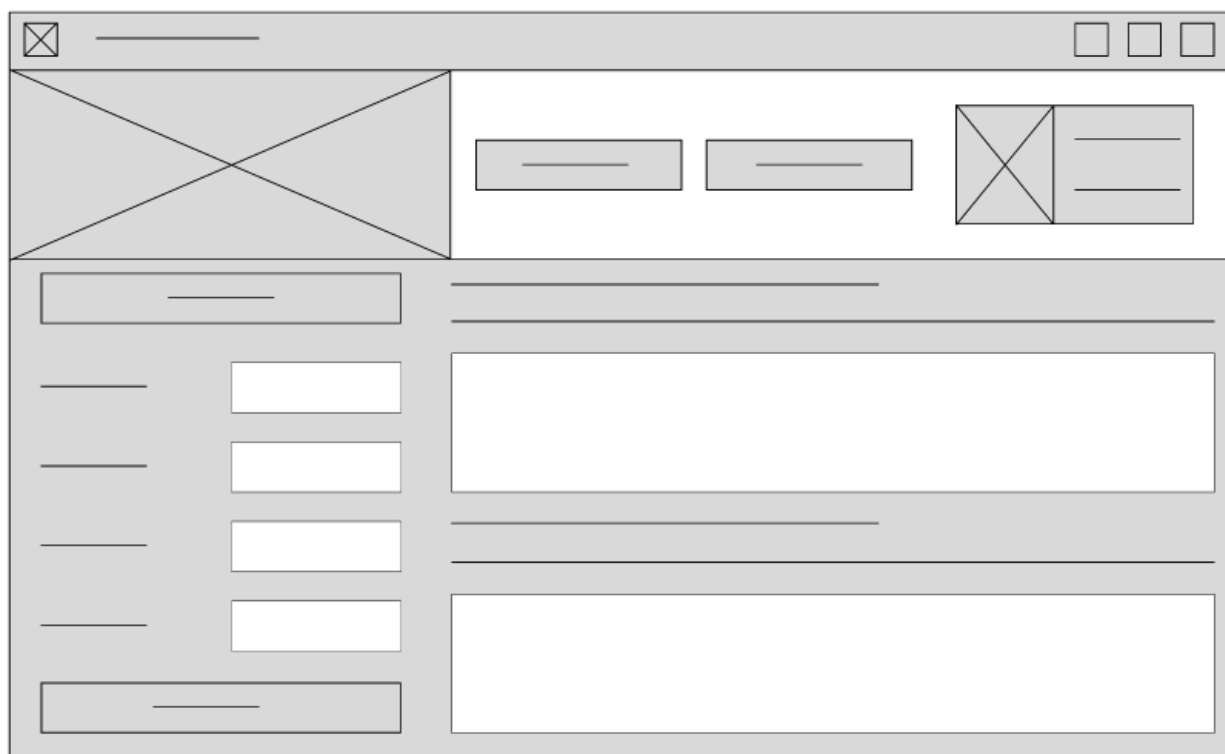


Рисунок Г.5 – Эскиз экрана редактирования проекта

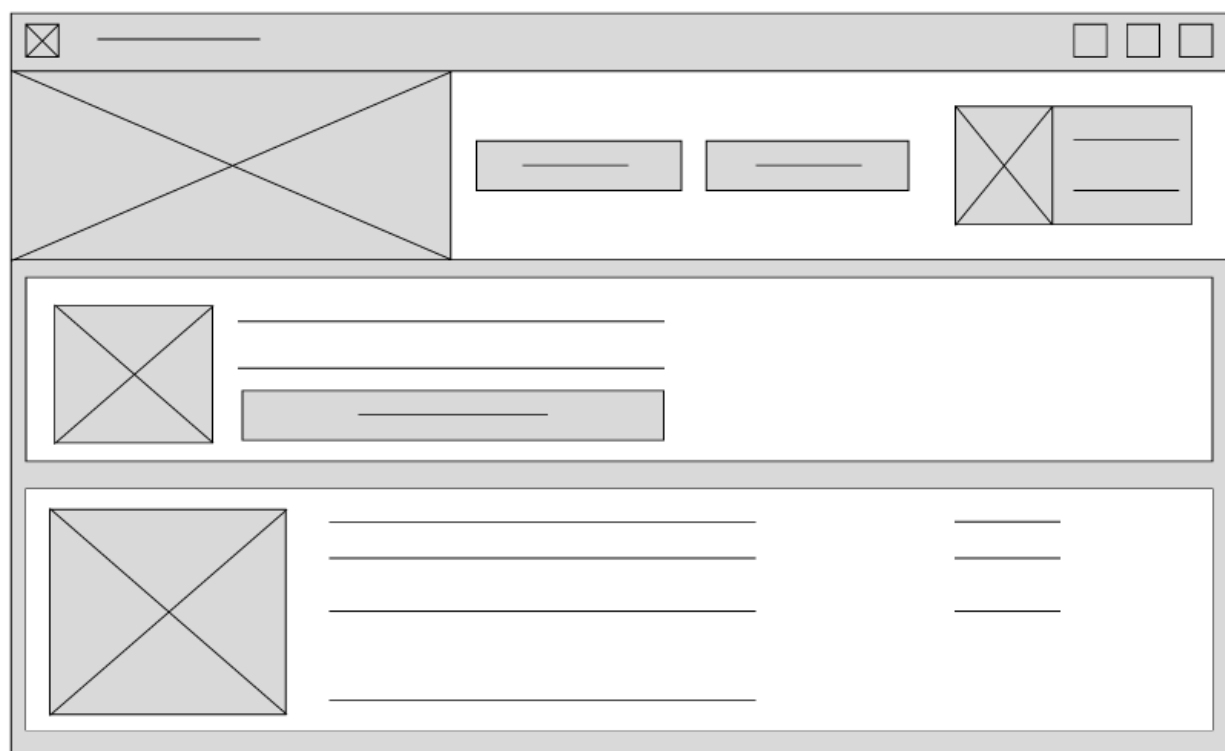


Рисунок Г.6 – Эскиз экрана личного кабинета пользователя

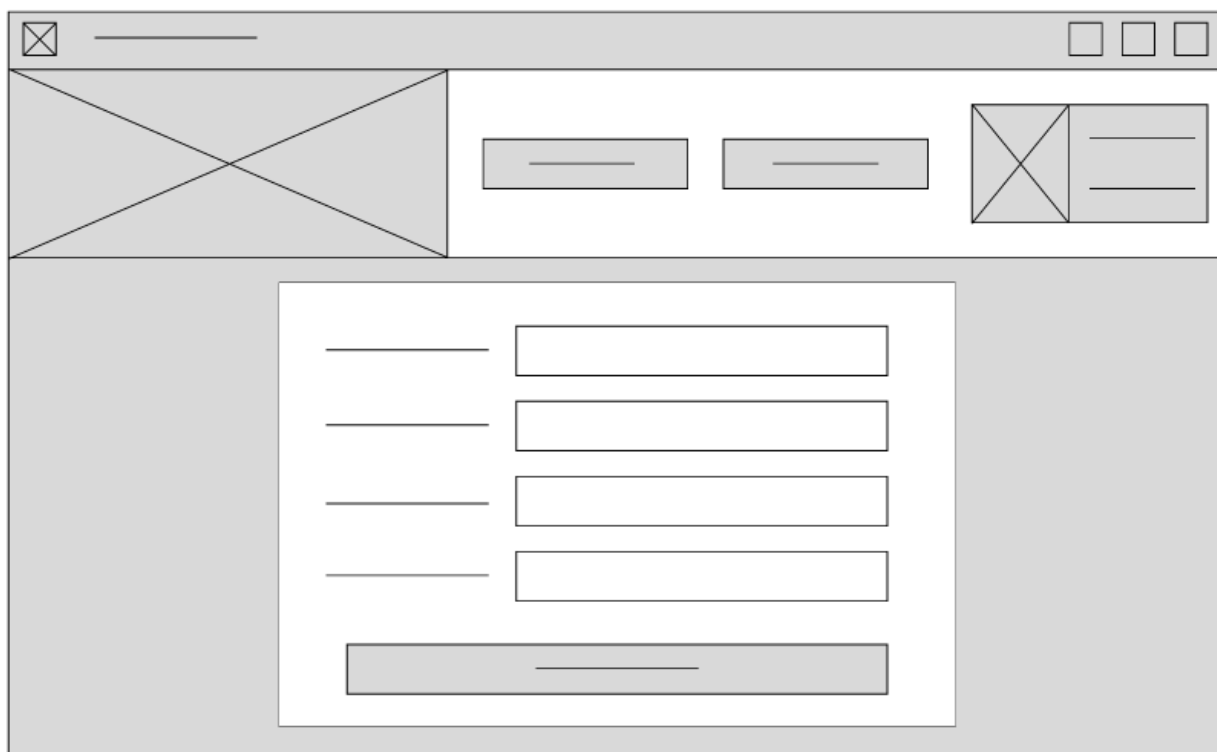


Рисунок Г.7 – Эскиз экрана редактирования профиля

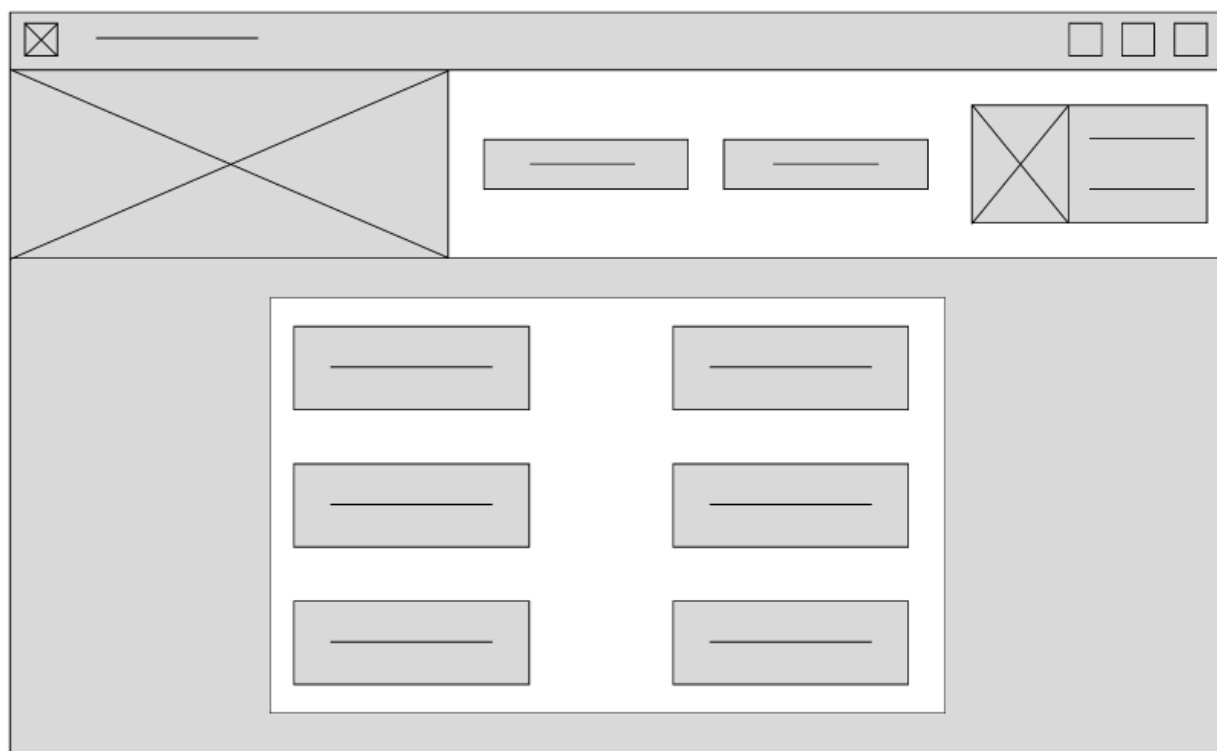


Рисунок Г.8 – Эскиз экрана панели администратора



НОВГОРОДСКИЙ  
ГОСУДАРСТВЕННЫЙ  
УНИВЕРСИТЕТ  
ИМЕНИ ЯРОСЛАВА МУДРОГО

Логин:

Пароль:

☐ Показать текст пароля

[Войти](#)

[Регистрация](#)

Рисунок Г.9 – Экран регистрации пользователя



НОВГОРОДСКИЙ  
ГОСУДАРСТВЕННЫЙ  
УНИВЕРСИТЕТ  
ИМЕНИ ЯРОСЛАВА МУДРОГО

Фамилия:

Имя:

Отчество:

E-mail:

Телефон:

Ваша роль:

Логин:

Пароль:

Повторите пароль:

☐ Показать текст пароля

[Зарегистрироваться](#)

[Авторизация](#)

Рисунок Г.10 – Экран авторизации пользователя







			Все проекты	Сначала новые	+ Добавить проект
	Black Sparrow	5	<p>Идея проекта состоит в том, чтобы перед входом в каждую аудиторию установить Считыватель кампусных карт студента, который будет автоматически фиксировать посещаемость занятия и вносить эти данные в электронный журнал.</p>		
	12.06.2023				
	Open City	4	<p>Идея проекта состоит в том, чтобы перед входом в каждую аудиторию установить Считыватель кампусных карт студента, который будет автоматически фиксировать посещаемость занятия и вносить эти данные в электронный журнал.</p>		
	11.06.2023				

Рисунок Г.11 – Экран списка проектов



Укажите название проекта, остальные данные можно будет добавить в карточке проекта:

Добавить новый проект

Рисунок Г.12 – Экран создания нового проекта



Удалить проект

Оценка: Идея

Оценка: Команда

Оценка: Инновационность

Оценка: Маркетинг

Оценка: Экономика

Оценка: Управление

Оценка: Бизнес-план

Оценка: Презентация

Оценить

Итоговая оценка - 0

Гашек  
Ярослав  
Наставник

Мицкевич  
Адам  
Менеджер

Название проекта

Должно отражать основную идею проекта.

Black Sparrow

Тип проекта

Выбрать:

Креативный

Вакансии

Напишите, кто нужен для вашего проекта (должность/роль).

1. Генератор идей

Заказчик проекта

Кто является заказчиком вашего проекта (если есть).

НовГУ

Цель проекта

Опишите цель вашего проекта. Помните, что цель должна быть достижимой и измеримой.

Идея проекта состоит в том, чтобы перед входом в каждую аудиторию установить Считыватель кампусных карт студента, который будет автоматически фиксировать посещаемость занятия и вносить эти данные в электронный журнал.

Рисунок Г.13 – Экран редактирования проекта



Мицкевич Адам Николаевич

Студент

Редактировать профиль

Hero in desert

Инновационный

Идея проекта состоит в том, чтобы перед входом в каждую аудиторию установить Считыватель кампусных карт студента, который будет автоматически фиксировать посещаемость занятия и вносить эти данные в электронный журнал.

06.06.2023

Вакансии:

1. Руководитель проекта 2. Финансовый менеджер 3. Менеджер по развитию

Open City

Предпринимательский

Идея проекта состоит в том, чтобы перед входом в каждую аудиторию установить Считыватель кампусных карт студента, который будет автоматически фиксировать посещаемость занятия и вносить эти данные в электронный журнал.

11.06.2023

Вакансии:

1. Участник

Black Sparrow

Креативный

Идея проекта состоит в том, чтобы перед входом в каждую аудиторию установить Считыватель кампусных карт студента, который будет автоматически фиксировать посещаемость занятия и вносить эти данные в электронный журнал.

Вакансии:

1. Генератор идей

Рисунок Г.14 – Экран личного кабинета пользователя

Проектная деятельность

НОВГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ ЯРОСЛАВА МУДРОГО

ПРОЕКТЫ

ВЫХОД

Адам Мицкевич

Фамилия: Мицкевич

Имя: Адам

Отчество: Николаевич

E-mail: Tadeusz@mail.com

Телефон: +7-951-251-51-81

Ваша роль: Студент

Логин: Adam

Текущий пароль:

Новый пароль:

Повторите пароль:

☐ Показать текст пароля

Сохранить изменения

Рисунок Г.15 – Экран редактирования профиля

Проектная деятельность

НОВГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ ЯРОСЛАВА МУДРОГО

АДМИН-ПАНЕЛЬ

ВЫХОД

Франц Кафка

Пользователи	Проекты
Статусы заявок	Роли в командах
Типы проектов	Типы продуктов
Роли в системе	Этапы проектов

Рисунок Г.16 – Экран панели администратора