МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ

«ПОЛЕССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

|  |  |
| --- | --- |
| Кафедра информационных технологий и интеллектуальных систем | Допущено к защите  Заведующий кафедрой  \_\_\_\_\_\_\_\_\_\_\_\_\_\_ М. А. Романова  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2021 |

**ДИПЛОМНЫЙ ПРОЕКТ**

на тему: **«**Информационная система «Электронная библиотека ПолесГУ**»**

Студент специальности

информационные системы

и технологии (по направлениям) \_\_\_\_\_\_\_\_\_\_ Семашко Маргарита Геннадьевна

4 курса, группы 17ИТ-2 (подпись) «\_\_»\_\_\_\_\_\_\_\_\_\_2021

Научный руководитель \_\_\_\_\_\_\_\_\_ Пигаль Павел Борисович

старший преподаватель (подпись) «\_\_»\_\_\_\_\_\_\_\_\_\_2021

ПИНСК 2021

**РЕФЕРАТ**

Дипломный проект: с., рис., табл., источника, прил.

ЭЛЕКТРОННАЯ БИБЛИОТЕКА, КАТАЛОГ, АВТОРИЗАЦИЯ, УПРАВЛЕНИЕ ПОЛЬЗОВАТЕЛЯМИ, ФУНКЦИОНАЛЬНОЕ ПРОЕТИРОВАНИЕ, ПРОГРАММНЫЙ ПРОДУКТ, АВТОМАТИЗАЦИЯ, ДИАГРАММА, СХЕМА, БАЗА ДАНЫХ, ПРОГРАММНЫЙ КОМПОНЕНТ, ТЕСТИРОВАНИЕ, ЭФФЕКТИВНОСТЬ, ОХРАНА ТРУДА.

Объектом дипломного проекта является библиотека ПолесГУ.

Предметом дипломного проекта является информационная система «Электронная библиотека ПолесГУ».

Целью дипломного проекта является разработка программного продукта **«**информационная система «Электронная библиотека ПолесГУ**»**.

Для достижения поставленной цели были выполнение следующие пункты:

* проанализированы практическиие материалы и полученная информация;
* выполнено концептуальное и функциональное моделирование автоматизированной системы;
* проведено тестирование и опытная эксплуатация разработанного продукта.
* проведено экономическое обоснование дипломного проекта.

В процессе работы над проектом проведены исследования в области электронной библиотеки и выполнены разработки для автоматизации хранения и использования совокупности информативных данных для создания информационной системы.

Экономический эффект разработанного программного приложения обеспечивает прирост прибыли, повышает эффективность работы сотрудника библиотекаря и, как следствие, в целом деятельности структурного подразделения.

Областью возможного практического применения являются ПолесГУ.

Автор дипломного проекта подтверждает, что приведенный в пояснительной записке расчетно-аналитический материал правильно и объективно отражает состояние исследуемого процесса, а все заимствованные из литературных и других источников теоретические, методологические и методические положения и концепции сопровождаются ссылками на их авторов.

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

(подпись студента)

**ОГЛАВЛЕНИЕ**

[ПЕРЕЧЕНЬ УСЛОВНЫХ ОБОЗНАЧЕНИЙ И СОКРАЩЕНИЙ 5](#_Toc71322231)

[ВВЕДЕНИЕ 6](#_Toc71322232)

[ГЛАВА 1 ТЕОРЕТИЧЕСКИЕ ОСНОВЫ ИНФОРМАЦИОННОЙ СИСТЕМЫ «ЭЛЕКТРОННАЯ БИБЛИОТЕКА ПОЛЕСГУ» 8](#_Toc71322233)

[1.1 Общая характеристика информационной системы 8](#_Toc71322234)

[1.2 Анализ и изучение информационной системы «Электронная библиотека» 9](#_Toc71322235)

[1.3 Сравнительный анализ аналогов информационной системы «Электронная библиотека ПолесГУ» 12](#_Toc71322236)

[ГЛАВА 2 ПРОЕКТИРОВАНИЕ ПРОГРАММНОГО ПРИЛОЖЕНИЯ ИНФОРМАЦИОННАЯ СИСТЕМА «ЭЛЕКТРОННАЯ БИБЛИОТЕКА ПОЛЕСГУ» 14](#_Toc71322237)

[2.1 Функциональное моделирование в IBM Rational Rose 14](#_Toc71322238)

[2.2 Диаграмма вариантов использования (use-case) 15](#_Toc71322239)

[2.3 Диаграммы последовательности 16](#_Toc71322240)

[2.4 Диаграммы кооперации 20](#_Toc71322241)

[2.5 Диаграммы классов 22](#_Toc71322242)

[2.6 Диаграммы состояний 24](#_Toc71322243)

[2.7 Диаграмма развертывания 25](#_Toc71322244)

[ГЛАВА 3 ПРОГРАММНАЯ РЕАЛИЗАЦИЯ ПРИЛОЖЕНИЯ ИНФОРМАЦИОННАЯ СИСТЕМА «ЭЛЕКТРОННАЯ БИБЛИОТЕКА ПОЛЕСГУ» 28](#_Toc71322245)

[3.1 Обоснование выбора среды разработки и языка программирования 28](#_Toc71322246)

[3.2 Построение базы данных системы 31](#_Toc71322247)

[3.3 Инфологическая модель базы данных 32](#_Toc71322248)

[3.3.1 Разработка логической модели 33](#_Toc71322249)

[3.3.2 Описание сущностей 33](#_Toc71322250)

[3.3.3 Описание связей 35](#_Toc71322251)

[3.3.4 ER-диаграмма 37](#_Toc71322252)

[3.4 Физическая модель 38](#_Toc71322253)

[3.4.1 Диаграмма связи по полям 40](#_Toc71322254)

[3.5 Описание структурных частей приложения 40](#_Toc71322255)

[3.5.1 Организация и управление данными 43](#_Toc71322256)

[3.5.2 Интерфейс информационной системы 43](#_Toc71322257)

[3.5.3 Тестирование информационной системы 43](#_Toc71322258)

[ГЛАВА 4 ОЦЕНКА ЭКОНОМИЧЕСКОЙ ЭФФЕКТИВНОСТИ РАЗРАБОТКИ ПРОГРАММНОГО ПРИЛОЖЕНИЯ 44](#_Toc71322259)

[ГЛАВА 5 ОХРАНА ТРУДА ПРИ ЭКСПЛУАТАЦИИ ПРОГРАММНОГО ПРОДУКТА 45](#_Toc71322260)

[5.1 Общая информация 45](#_Toc71322261)

[5.2 Охрана труда инженера-программиста 45](#_Toc71322262)

[5.3 Охрана труда при работе на ПК 47](#_Toc71322263)

[ЗАКЛЮЧЕНИЕ 49](#_Toc71322264)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 50](#_Toc71322265)

[ПРИЛОЖЕНИЯ 53](#_Toc71322266)

[ВЕДОМОСТЬ ОБЪЁМА ДИПЛОМНОГО ПРОЕКТА 69](#_Toc71322269)

ПЕРЕЧЕНЬ УСЛОВНЫХ ОБОЗНАЧЕНИЙ И СОКРАЩЕНИЙ

В настоящей пояснительной записке применяются следующие термины, обозначения и сокращения.

БД — База Данных.

СУБД — Система Управления Базами Данных.

ТБД –– Таблица Базы Данных.

ПО — Программное Обеспечение.

MS — MicroSoft.

COM (Component Object Model) — модель компонентного объекта.

GUI (Graphical User Interface) — графический интерфейс пользователя.

SQL (Structured Query Language) — язык структурированных запросов. Универсальный компьютерный язык, применяемый для создания, модификации и управления данными в реляционных базах данных.

CASE (Computer-Aided Software Engineering) –– набор инструментов и методов для проектирования ПО, который помогает обеспечить высокое качество программ, отсутствие ошибок и простоту в обслуживании программных продуктов.

ORM - технология программирования, которая позволяет преобразовывать несовместимые типы моделей в ООП, в частности, между хранилищем данных и объектами программирования.

ООП - объектно-ориентированное программирование, парадигма программирования.

# ВВЕДЕНИЕ

Деятельность каждой библиотеки базируется на определенной технологии, формирование которой осуществляется под влиянием ряда факторов: внешних и внутренних. Решающая роль в данном случае принадлежит технологической среде, окружающей библиотеку.

Технологические нововведения активно влияют в первую очередь на процессы, связанные с формированием информационных ресурсов, и организацию обслуживания пользователей. Вся история развития библиотек — это история использования в их деятельности тех технологических решений, которые получили широкое распространение в общественной жизни.

На развитие библиотечного дела наибольшее влияние оказали технологические решения в области компьютерной техники и средств связи. Использование данных технологий позволило библиотекам перейти на приобретение электронных документов, продолжить работы по миниатюризации документов и созданию их электронных копий, что значительно увеличило сохранность фондов и снизило потребность в площади хранилищ.

Благодаря вычислительной технике значительно расширились возможности поискового аппарата библиотек. Перевод традиционных карточных каталогов и картотек, а также библиографических указателей в электронную форму увеличил объем предоставляемой информации, расширил возможности поиска необходимых сведений.

Компьютеризация позволила также внести существенные изменения в организацию обслуживания пользователей. Наличие электронных документов и современных средств связи создало условия для удаленного доступа пользователей к информационным ресурсам библиотек и получения необходимых материалов на дом или на рабочее место практически из любой библиотеки мира [1].

Целью дипломного проекта является разработка программного продукта **«**информационная система «Электронная библиотека ПолесГУ**»**. Для достижения цели необходимо выполнить такие задачи, как:

* анализ изученных практических материалов и полученной информации;
* оценка эффективности методов разрешения проблем с точки зрения их целесообразности, результативности и ресурсного обеспечения;
* выполнить концептуальное и функциональное моделирование автоматизированной системы;
* провести тестирование и опытную эксплуатацию разработанного продукта;
* провести экономическое обоснование дипломного проекта;
* предоставить охрану труда при эксплуатации программного продукта.

Разработка программного продукта информационной системы «Электронная библиотека ПолесГУ» позволит:

* повысить эффективность деятельности библиотекарей;
* сократить время на поиск необходимой информации;
* увеличить скорость и качество работы.

# ГЛАВА 1 ТЕОРЕТИЧЕСКИЕ ОСНОВЫ ИНФОРМАЦИОННОЙ СИСТЕМЫ «ЭЛЕКТРОННАЯ БИБЛИОТЕКА ПОЛЕСГУ»

* 1. Общая характеристика информационной системы

Под системой понимают любой объект, который одновременно рассматривается и как единое целое, и как объединенная в интересах достижения поставленных целей совокупность разнородных элементов. Системы значительно отличаются между собой как по составу, так и по главным целям.

**Система** – это сложный объект, состоящий из взаимосвязанных частей (элементов) и существующий как единое целое.

**Подсистема** – это часть системы, выделенная по какому-либо признаку [2].

**Информационная система** (ИС) — [система](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0), предназначенная для хранения, поиска и обработки [информации](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D0%B8%D1%8F), и соответствующие организационные ресурсы (человеческие, технические, финансовые и т. д.), которые обеспечивают и распространяют информацию [3].

Классификация информационной системы:

* Классификация ИС по назначению:
  + Информационно – управляющие;
  + Системы поддержки принятия решений;
  + Информационно – поисковые;
  + Информационно – справочные;
  + Системы обработки данных;
* Классификация ИС по структуре аппаратных средств:
  + Однопроцессорные;
  + Многопроцессорные;
  + Многомашинные;
  + Системы с удаленным доступом;
  + Вычислительные сети;
* Классификация ИС по режиму работы:
  + Пакетная обработка;
  + Режим индивидуального пользования;
  + Режим коллективного пользования;
* Классификация ИС по характеру взаимодействия с пользователем:
  + Диалоговый режим;
  + Интерактивный режим;
  + Режим реального времени;
* Классификация ИС по уровню автоматизации:
  + Ручные;
  + Автоматизированные;
  + Автоматические [4].
  1. Анализ и изучение информационной системы «Электронная библиотека»

Электронная библиотека - распределенная информационная система, позволяющая надежно сохранять и эффективно использовать разнородные коллекции электронных документов через глобальные сети передачи данных в удобном для конечного пользователя виде.

**Электронная библиотека** — это упорядоченная коллекция разнородных электронных документов, снабженных средствами навигации и поиска. Может быть веб-сайтом, где постепенно накапливаются различные тексты (чаще литературные, но также и любые другие, вплоть до компьютерных программ) и медиа-файлы, каждый из которых самодостаточен и в любой момент может быть востребован читателем [5].

Электронная библиотека может быть реализована в форме Web-сайта, где постепенно накапливаются различные тексты, мультимедиа-файлы, каждый из которых самодостаточен и в любой момент может быть востребован читателем.

Основным способом структурирования источников в библиотеке является каталогизация. Каталогизация — это совокупность процессов, обеспечивающих создание и функционирование библиотечных каталогов. Библиотечный каталог — совокупность расположенных по определенным правилам библиографических записей, раскрывающих состав и содержание фонда библиотеки. Библиотечные каталоги могут иметь карточную и машиночитаемую форму и быть реализованы на электронных носителях или в форме книжного издания.

С внедрением средств информационных технологий библиотеки приступили к созданию электронных каталогов.

**Электронный каталог** — это библиотечный каталог в машиночитаемой форме, обеспечивающий доступ к ресурсам каталога на расстоянии.

Поиск документов в электронных каталогах осуществляется по различным признакам: ключевым словам, предметным рубрикам и всем другим полям электронного каталога [6].

Существуют разные типы электронных библиотек, рассмотрим некоторые из них:

* Институциональные репозитории.

Многие академические библиотеки активно участвуют в создании институциональных репозиториев книг, статей, диссертаций и других работ учреждения, которые могут быть оцифрованы или были «рождены цифровыми». Многие из этих репозиториев доступны для широкой публики с небольшими ограничениями в соответствии с целями открытого доступа, в отличие от публикации исследований в коммерческих журналах, где издатели часто ограничивают права доступа. Институциональные, действительно бесплатные и корпоративные репозитории иногда называют электронными библиотеками. Программное обеспечение институционального репозитория предназначено для архивирования, систематизации и поиска содержимого библиотеки.

* Коллекции национальной библиотеки.

Обязательный депозит часто регулируется законодательством об авторском праве, а иногда и законами, относящимися к обязательному экземпляру, и требует, чтобы одна или несколько копий всех материалов изданные в стране должны быть отправлены на хранение в учреждение, обычно в национальную библиотеку. С появлением электронных документов в законодательство пришлось внести поправки, чтобы охватить новые форматы.

* Цифровые архивы.

Физические архивы отличаются от физических библиотек несколькими способами. Традиционно архивы определяются как:

* + содержащие первичные источники информации (как правило, письма и бумаги, непосредственно созданные отдельным лицом или организацией), а не вторичные источники, найденные в библиотеке (книги, периодические издания и т. д.)).
* организация их содержимого по группам, а не по отдельным элементам.
* уникальное содержимое.

Технология, используемая для создания электронных библиотек, является еще более революционной для архивов, поскольку она разбивает второе и третье из этих общих правил. Другими словами, «цифровые архивы» или «онлайн-архивы», как правило, по-прежнему будут содержать первоисточники, но они, вероятно, будут описаны индивидуально, а не в группах или коллекциях (или в дополнение к ним). Кроме того, поскольку они являются цифровыми, их содержимое легко воспроизводимо и действительно могло быть воспроизведено из других источников.

Архивы отличаются от библиотек по характеру хранимых материалов. Библиотеки собирают отдельные опубликованные книги и сериалы или ограниченные наборы отдельных предметов. Книги и журналы, хранящиеся в библиотеках, не уникальны, поскольку существует несколько копий, и любой данный экземпляр обычно оказывается таким же удовлетворительным, как и любой другой экземпляр. Материалы в архивах и библиотеках рукописей представляют собой «уникальные записи юридических лиц и документы отдельных лиц и семей».

Фундаментальной характеристикой архивов является то, что они должны сохранять контекст, в котором были созданы их записи и сеть взаимоотношений между ними, чтобы сохранить их информативное содержание и предоставить понятную и полезную информацию с течением времени. Основная характеристика архивов заключается в их иерархической организации, выражающей контекст посредством архивной связи. Архивные описания являются основным средством описания, понимания, поиска и доступа к архивным материалам. На цифровом уровне архивные описания обычно кодируются с помощью формата XML Encoded Archival Description. EAD - это стандартизированное электронное представление архивного описания, которое позволяет предоставить профсоюзу доступ к подробным архивным описаниям и ресурсам в репозиториях, распределенных по всему миру.

Учитывая важность архивов, была определена специальная формальная модель, названная Сетевые сети для иерархий объектов (NESTOR), построенные вокруг их специфических составляющих. NESTOR основан на идее выражения иерархических отношений между объектами через свойство включения между наборами, в отличие от двоичного отношения между узлами, используемыми деревом. NESTOR использовался для формального расширения модели 5S, чтобы определить цифровой архив как особый случай цифровой библиотеки, способной учитывать особенности архивов [7].

* 1. Сравнительный анализ аналогов информационной системы «Электронная библиотека ПолесГУ»

Независимо от того, является ли Электронная библиотека локальной или она выставлена в Интернете (с разными условиями доступа), ее создание должно быть направлено на достижение основной цели, которая видится в удовлетворении информационных потребностей. Используемые технологии и методики должны соответствовать специфике обозначенных информационных потребностей, рациональной организации массива электронных документов, сформированного по выбранным критериям отбора.

Электронная библиотека, наиболее отвечающая требованиям конкретного ВУЗа должна быть универсальной: включать научную, учебную, методическую литературу (в том числе периодические издания) по всем дисциплинам данного вуза.

Однако не все существующие Интернет-библиотеки отвечают этим требованиям, и это естественно, поскольку создатели электронных библиотек в сети, чаще всего, делали упор либо на узкоспециальной тематике, либо на определенном типе литературы: учебники и научная литература или периодические издания. В настоящее время при возникшей ситуации существующие электронные библиотеки начинают процесс «доукомплектования» (универсализации) своего электронного контента недостающими типами литературы либо расширяют тематику и количество произведений для соответствия требованиям, предъявляемым к Электронная библиотека.

При анализе существующих Интернет-библиотек были учтены несколько основных критериев:

* лицензионно чистый контент;
* актуальность контента;
* удобный многоуровневый поиск, интерфейс.

Рассмотрим три Электронные библиотеки Белорусских ВУЗов в таблицу 1.1.

Таблица 1.1 – Сравнительная характеристика Электронных библиотек Белорусских ВУЗов.

|  |  |  |  |
| --- | --- | --- | --- |
| Пункты сравнения | Электронная библиотека БГУ | Электронная библиотека БГЭУ | Электронная библиотека ГРГУ |
| Представлены полнотекстовые документы/библиография/ссылки на другие сайты | Полнотекстовые документ | Полнотекстовые документ | Библиография |
| Лицензионно чистый контент | Да | Да | Да |
| Статистика посещения сайта | Нет | Нет | Нет |
| На какую аудиторию расчитаны ресурсы | Студенты и преподаватели | Студенты и преподаватели | Студенты и преподаватели |
| Актуальностьконтента | Да | Да | Да |
| Удобный многоуровневый поиск, интерфейс | Да | Да | Нет |
| Классификация электронной библиотеки | Репозиторий + Электронные издания | Репозиторий + Электронные издания | Электронный каталог |
| Кто является обладателем сайта | БГУ | БГЭУ | ГРГУ |
| Есть возможность размещение свободных оцифрованых произведений | Нет | Нет | Нет |
| Могут пользователя оставлять замечания | Да | Да | Да |
| Есть возможность смены языка | Да | Да | Нет |
| Есть возможность входа | Да | Да | Нет |

На данный момент Полесский Государственный Университет не может конкурировать с данными ВУЗами по части Электронной библиотеки. В ПолесГУ представлен хороший репозиторий, так же существует каталог книг, но он очень неудобен и не функционален. Многие студенты даже не знают о его существовании.

Разрабатываемая информационная система должна упростить работу библиотекарям, а также помочь студентам проще получать информацию о библиотеке, о количестве книг на руках, о новых поступлениях и т.д.

# ГЛАВА 2 ПРОЕКТИРОВАНИЕ ПРОГРАММНОГО ПРИЛОЖЕНИЯ ИНФОРМАЦИОННАЯ СИСТЕМА «ЭЛЕКТРОННАЯ БИБЛИОТЕКА ПОЛЕСГУ»

* 1. Функциональное моделирование в IBM Rational Rose

К числу хорошо распространённых средств визуального моделирования объектно-ориентированных информационных систем (ИС) относится Rational Rose.

Rational Rose –– мощное CASE-средство для проектирования программных систем любой сложности. Одним из достоинств этого программного продукта будет возможность использования диаграмм на языке UML, с помощью которого Rational Rose может решить задачи в сфере проектирования информационных систем: начиная с исследования бизнес-процессов, заканчивая кодогенерацией на установленном языке программирования.

Rational Rose разрешает разработать не только высокоуровневые, но и низкоуровневые модели, выполняя или абстрактное, или логическое проектирования. Также Rational Rose обладает всем нужным комплектом визуальных средств проектирования. Rational Rose идеально подходит, чтобы разрешить вопросы с кодогенерацией на конкретном языке программирования. Rational Rose реализовывает следующие подходы: Round Trip Engineering, а также прямое и обратное проектирование. Можно сказать, что Rational Rose является графическим редактором UML диаграмм [8].

В распоряжение проектировщика системы Rational Rose предоставляет следующие типы диаграмм, последовательное создание которых позволяет получить полное представление о всей проектируемой системе и об отдельных ее компонентах:

* Use case diagram (диаграммы прецедентов);
* Deployment diagram (диаграммы топологии);
* Statechart diagram (диаграммы состояний);
* Activity diagram (диаграммы активности);
* Interaction diagram (диаграммы взаимодействия);
* Sequence diagram (диаграммы последовательностей действий);
* Collaboration diagram (диаграммы сотрудничества);
* Class diagram (диаграммы классов);
* Component diagram (диаграммы компонент) [9].
  1. Диаграмма вариантов использования (use-case)

Диаграмма вариантов использования в UML — диаграмма, отражающая отношения между актёрами и прецедентами и являющаяся составной частью модели прецедентов, позволяющей описать систему на концептуальном уровне [10].

На рисунке 2.1 изображена диаграмма вариантов использования информационной системы. Данная диаграмма формирует статическое представление вариантов использования системы, но не является прямым отражением ее работы. На диаграмме представлены абстрактные действующие лица.



**Рисунок 2.1 – Диаграмма вариантов использования**

На диаграмме представлены абстрактные действующие лица: «Студент», «Супер админ», «Сотрудник», «Гость».

Система в общих чертах работает следующем образом: гость просматривает каталог; студент входит в систему, просматривает каталог, просматривает количество книг на руках, выбирает нужную ему книгу, оставляет запрос, если книга есть в наличии. Сотрудник просматривает запрос и отвечает, что студент может получить книгу определённого числа в определённое время. Сотрудник может добавлять пользователей, книги и редактировать данные. Супер админ имеет доступ ко всему. Диаграмма представлена в приложении А.

* 1. Диаграммы последовательности

Диаграмма последовательности — UML-диаграмма, на которой для некоторого набора объектов на единой временной оси показан жизненный цикл объекта и взаимодействие актеров информационной системы в рамках прецедента [11].

Для подробного описания информационной системы используются несколько диаграмм последовательностей.

На рисунке 2.2 представлена диаграмма последовательности варианта использования «Добавление книги».



**Рисунок 2.2 –– Диаграмма последовательности варианта использования «Добавление книги»**

Описание последовательности действий:

* Сотрудник входит в систему;
* Открывает форму добавления книги;
* Вводит все необходимые данные и нажимает кнопку добавить;
* Происходит проверка введённых данных, если всё верно книга созданы и отображается на панели админа, если нет, то форма просит ввести данные по-новому;
* Добавление книги в базу данных.

На рисунке 2.3 представлена диаграмма последовательности варианта использования «Создание пользователя».



**Рисунок 2.3 –– Диаграмма последовательности варианта использования «Создание пользователя»**

Описание последовательности действий:

* Супер админ входит в систему;
* Открывает форму создания книги;
* Вводит все необходимые данные и нажимает кнопку добавить;
* Происходит проверка введённых данных, если всё верно пользователь создан и отображается на панели админа, если нет, то форма просит ввести данные по-новому;
* Добавление пользователя в базу данных.

На рисунке 2.4 представлена диаграмма последовательности варианта использования «Запрос на книгу».



**Рисунок 2.4 –– Диаграмма последовательности варианта использования «Запрос на книгу**

Описание последовательности действий:

* Студент входит в систему;
* Выбирает нужную ему книгу;
* Отправляет заявку на данную книгу
* Если книга имеется в наличии студент может её получить, если книга у кого-то на руках, то студент может оставить заявку на будущее;
* Студент получает книгу.

Таким образом данный вид диаграмм подробно описывает действие за действием каждого актера в процессе управления персоналом в рамках одного варианта использования.

* 1. Диаграммы кооперации

Диаграмма кооперации — диаграмма, на которой изображаются взаимодействия между частями композитной структуры или ролями кооперации. В отличие от диаграммы последовательности, на диаграмме коммуникации явно указываются отношения между объектами, а время как отдельное измерение не используется.

Диаграмма кооперации является разновидностью диаграммы последовательности, и, в контексте языка UML, описывает динамический аспект кооперации объектов, при реализации различных вариантов использования. Обычно такие диаграммы описывают реализацию типичного хода событий при работе системы. Она предназначена для описания поведения системы на уровне отдельных объектов, которые обмениваются между собой сообщениями, чтобы достичь нужной цели или реализовать некоторый вариант использования.

Сообщения в языке UML также специфицируют роли, которые играют объекты — отправитель и получатель сообщения. Сообщения на диаграмме кооперации изображаются дополнительными стрелками рядом с соответствующей связью или ролью ассоциации. Направление стрелки указывает на получателя сообщения.

Процесс построения диаграммы кооперации должен быть согласован с процессами построения диаграммы классов и диаграммы последовательности. В первом случае, как уже отмечалось, необходимо следить за использованием только тех объектов, для которых определены порождающие их классы. Во втором случае необходимо согласовывать последовательности передаваемых сообщений [12].

На рисунке 2.5 представлена диаграмма кооперации варианта использования «Добавление книги».



**Рисунок 2.5 –– Диаграмма кооперации варианта использования «Добавление книги»**

На рисунке 2.6 представлена диаграмма кооперации варианта использования «Запрос на книгу».



**Рисунок 2.6 –– Диаграмма кооперации варианта использования «Запрос на книгу»**

На рисунке 2.7 представлена диаграмма кооперации варианта использования «Создание пользователя».



**Рисунок 2.7 –– Диаграмма кооперации варианта использования «Создание пользователя»**

* 1. Диаграммы классов

Диаграмма классов — структурная диаграмма языка моделирования UML, демонстрирующая общую структуру иерархии классов системы, их коопераций, атрибутов, методов, интерфейсов и взаимосвязей между ними. Под классами описаны атрибуты класса и их операции. В данной диаграмме класс связаны таким отношением, как ассоциация [13].

На данной диаграмме все классы разделены на пакеты: «Управление», «Границы», «Сущности». К пакету «Управление» относят такие классы, как:

* База данных;
* Менеджер книг;
* Менеджер пользователей;
* Менеджер запросов.

К пакету «Границы» относят такие классы, как:

* Каталог книг;
* Форма входа;
* Форма добавления книг;
* Форма добавления пользователя.

К пакету «Сущности» относят такие классы, как:

* Книга;
* Пользователь.

Диаграмма класов «Добавление пользователя» представлена на рисунке 2.8.



**Рисунок 2.8 –– Диаграмма классов «Добавление пользователя»**

Диаграмма класов «Добавление книги» представлена на рисунке 2.9.



**Рисунок 2.9 –– Диаграмма классов «Добавление книги»**

Диаграмма класов «Запрос на книгу» представлена на рисунке 2.10.



**Рисунок 2.10 –– Диаграмма классов «Запрос на книгу»**

* 1. Диаграммы состояний

Диаграмма состояний — это, по существу, диаграмма состояний из теории автоматов со стандартизированными условными обозначениями, которая может определять множество систем от компьютерных программ до бизнес-процессов [14].

Диаграмма состояний показывает все возможные состояния, в которых может находиться объект, а также процесс смены состояний в результате внешнего влияния. Основными элементами диаграммы состояний являются «Состояние» и «Переход». Диаграмма состояний имеет схожую семантику с диаграммой деятельности, только деятельность здесь заменена состоянием, переходы символизируют действия [15].

Одним из главных процессов является добавление книги в базу, который затрагивает стадии диаграммы действий оператора. Поэтому в данном пункте будет представлена на рисунке 2.11 диаграмма состояний для объекта создания книги.



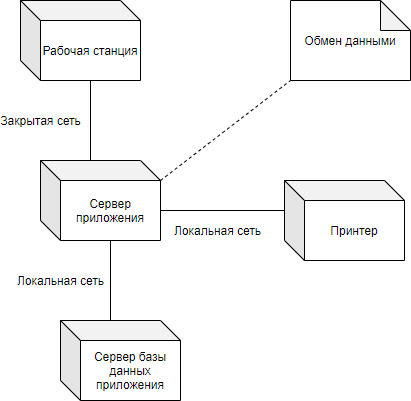
**Рисунок 2.11 –– Диаграмма состояний «Добавление книги»**

* 1. Диаграмма развертывания

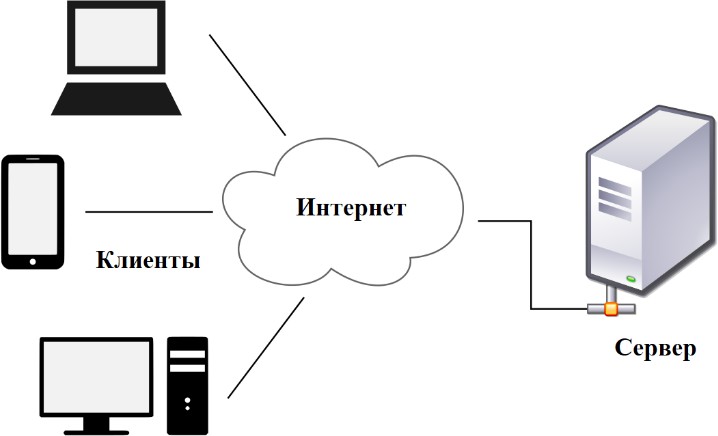
Данная диаграмма показывает топологию системы и распределение компонентов системы по ее узлам, а также соединения –– маршруты передачи информации между аппаратными узлами [16].

Диаграммы размещения – это один из двух видов диаграмм, используемых при моделировании физических аспектов объектно- ориентированной системы. Диаграмма используются для моделирования статического представления системы с точки зрения размещения. В основном под этим понимается моделирование топологии аппаратных средств, на которых работает система. На рисунке 2.12 представлена диаграмма размещения для разрабатываемого программного приложения.

В основе разрабатываемой системы лежит архитектура «клиент- сервер», в которой задания или сетевая нагрузка распределены между поставщиками услуг (сервисов), называемых серверами, и заказчиками услуг, называемых клиентами. В качестве среды взаимодействия клиента с сервером используется интернет (рисунок 2.13).



**Рисунок 2.12 - Диаграмма развёртывания**



**2.13 – Архитектура приложения**

В соответствии с изученной предметной областью было произведено функциональное проектирование в IBM Rational Rose и спроектированы все виды диаграмм, представленные выше.

Для описания основных функций программного продукта была использована диаграмма вариантов использования, которая более детально показывает логику работы приложения.

# ГЛАВА 3 ПРОГРАММНАЯ РЕАЛИЗАЦИЯ ПРИЛОЖЕНИЯ ИНФОРМАЦИОННАЯ СИСТЕМА «ЭЛЕКТРОННАЯ БИБЛИОТЕКА ПОЛЕСГУ»

3.1 Обоснование выбора среды разработки и языка программирования

Для достижения цели дипломного проекта было выбрано объектно- ориентированное CASE-средство проектирования информационных систем - IBM Rational Rose, имеющее возможности: проектирование систем любой сложности, поддержка языка UML, генерирование кода, многоплатформенность, интеграция с Microsoft Visual Studio и др. Данное CASE-средство относится к разряду средств объектно-ориентированного моделирования.

Для разработки программного приложения была использована интегрированная среда разработки программ Visual Studio, платформа ASP.NET Core MVC с использование объектно-ориентированного языка C#.

Платформа .NET Framework служит средой для поддержки разработки и выполнения сильно распределенных компонентных приложений. В данной среде возможно совместное использование разных языков программирования. А поскольку в C# библиотека классов .NET используется автоматически, то программы на C# заведомо оказываются переносимыми во все имеющиеся среды .NET Framework, что является значительным достоинством среды разработки [17].

Основной средой для разработки программ на C# служит Visual Studio корпорации Microsoft. Такая среда дает возможность править, компилировать, выполнять и отлаживать программы на С#, не покидая эту грамотно организованную среду.

Основными преимуществами Visual Studio являются:

* использование вычислительных мощностей локального компьютера и облака;
* простая реализация общих задач и индивидуальный подход;
* быстрое создание высококачественного кода;
* возможность реализации идей и решений для широкого спектра платформ, включая Windows, Windows Server, веб-среду, облачную среду, Office и SharePoint [18];

Visual Studio можно использовать для создания различных типов приложений, от простых приложений для магазина и игр для мобильных клиентов до больших и сложных систем, обслуживающих предприятия и центры обработки данных. Учитывая все эти, характеристики, для своего программного приложения была выбрана среда Visual Studio 2019.

К числу принципиально важных решений, которые реализованы «корпорацией Microsoft» в языке программирования «C#», можно отнести следующие:

* Компонентно‒ориентированный подход к программированию (который характерен и для идеологии «Microsoft .NET» в целом);
* Свойства как средство инкапсуляции данных (характерно также в целом для ООП);
* Обработка событий (имеются расширения, в том числе в части обработки исключений, в частности, оператор «try»);
* Унифицированная система типизации (соответствует идеологии «Microsoft .NET» в целом);
* Перегруженные операторы (развитие ООП);
* Оператор «foreach» (обработка всех элементов классов-коллекций, аналог «Visual Basic»);
* Механизмы «boxing» и «unboxing» для преобразования типов;
* Атрибуты (средство оперирования метаданными в COM-модели) [19].

В процессе написания программного продукта был также использован фреймворк .NET Core.

ASP.NET Core является кроссплатформенным, высокопроизводительным фреймворком с открытым исходным кодом, который позволяет:

* Cоздавать современные приложения;
* Создавать веб-приложения и службы, приложения IoT и серверные части для мобильных приложений;
* Использовать избранные средства разработки в Windows, macOS и Linux;
* Выполнять развертывания в облаке или локальной среде;
* Работать в .NET Core или .NET Framework [20].

ASP.NET Core предоставляет следующие преимущества:

* Единое решение для создания пользовательского веб-интерфейса и веб-API.
* Разработано для тестируемости.
* Razor Pages делает создание кодов сценариев для страниц проще и эффективнее.
* Возможность разработки и запуска в ОС Windows, macOS и Linux.
* Упрощенный высокопроизводительный модульный конвейер HTTP- запросов.
* Возможность размещения в IIS, Nginx, Apache, Docker или в собственном процессе.
* Параллельное управление версиями приложения, ориентированное на .NET Core.
* Инструментарий, упрощающий процесс современной веб-разработки.

Для данного приложения в качестве Веб-сервера используется Microsoft IIS 7.0. А также .NET Core 3.1 IIS и .NET Framework входящие в поставку Windows Server 2019 и не требующие дополнительных действий для установки и большей части конфигурирования.

ASP.NET Core MVC, реализуя шаблон MVC облегчает управление сложными структурами путем разделения приложения на модель, представление и контроллер [21]. Идея, которая лежит в основе конструкционного шаблона MVC очень проста: нужно четко разделять ответственность за различное функционирование в приложении.

ASP.NET Core MVC обеспечивает решения многих задач, с которыми сталкиваются разработчики и организации, которые хотят создать информационную систему, основанную на платформе .NET.

Объектная модель хорошо себя зарекомендовала при разработке больших систем. Однако, при создании сложных систем программисту приходится одновременно работать как в объектной области, так и в области SQL. Это неизбежно приводит к увеличению сложности, разрабатываемой системы. Поэтому для разработки приложения был выбран подход организации данных предлагающий забыть об SQL и мыслить объектно, но в то же время использовать достижения разработчиков реляционных баз данных, путем осуществления отображения объектов и таблиц [22].

Объектно-реляционное отображение (англ. Object-relational mappiпg - ORM) — технология программирования, которая связывает базы данных с концепциями объектно-ориентированных языков программирования, создавая виртуальную объектную базу данных.

В качестве ORM системы была выбрана система ADO.NET Entity Framework, позволяющая создать приложение, работающие с концептуальной моделью данных, а не напрямую с реляционной схемой их хранения [19].

Использование ORM системы позволило облегчить процесс размещения данных в СУБД: MS SQL Server. Одним из преимуществ SQL Server является простота его применения, в частности администрирования. Так же, данная СУБД имеет ряд таких преимуществ, как:

* Высокая степень защиты данных.
* Мощные средства работы с данными.
* Высокая производительность.
* Хранение больших массивов данных.
* Хранение данных, требующих соблюдения режима секретности или при недопустимости их потери.

Для визуальной части использовался веб-интерфейс, с использованием языков CSS, HTML, библиотек Bootstrap, JQuery. Выбор был сделан в сторону использования данных средств визуализации на основании следующих преимуществ, предоставляемых ими:

* адаптивность;
* кросс-браузерность;
* легкость в использовании и быстрота в освоении;
* понятный код;
* единство стилей.

3.2 Построение базы данных системы

В основе разработанного веб-приложения лежит реляционная база данных, которая будет хранить информацию о товарах в магазине и складе.

**Реляционная база данных** — это совокупность взаимосвязанных таблиц, каждая из которых содержит информацию об объектах определенного типа. Строка таблицы содержит данные об одном объекте (например, товаре, клиенте), а столбцы таблицы описывают различные характеристики этих объектов — атрибутов (например, наименование, код товара, сведения о клиенте). Записи, т. е. строки таблицы, имеют одинаковую структуру — они состоят из полей, хранящих атрибуты объекта. Каждое поле, т. е. столбец, описывает только одну характеристику объекта и имеет строго определенный тип данных. Все записи имеют одни и те же поля, только в них отображаются различные информационные свойства объектах [23].

Реляционная модель – модель представления данных, которая описывает структуру данных, допустимые операции над данными и специальные правила, обеспечивающие целостность данных.

**База данных** — это один или несколько файлов данных, предназначенных для хранения, изменения и обработки больших объемов взаимосвязанной информации [24].

В разработанной базе данных используется два известных типа связи: «один ко многим» и «один к одному».

Отношение «один-ко-многим» является наиболее часто используемым типом связи между таблицами. В отношении «один-ко-многим» каждой записи одной таблице могут соответствовать несколько записей в другой таблице [25]. В отношении «один к одному» каждой записи одной таблице могут соответствовать одна запись в другой таблице.

3.3 Инфологическая модель базы данных

Цель инфологического моделирования – обеспечение наиболее естественных для человека способов сбора и представления той информации, которую предполагается хранить в создаваемой базе данных. Поэтому инфологическую модель данных пытаются строить по аналогии с естественным языком, который не может быть использован в чистом виде из-за сложности компьютерной обработки текстов и неоднозначности любого естественного языка. Основными конструктивными элементами инфологических моделей являются сущности, связи между ними и их свойства (атрибуты).

**Сущность**– любой различимый объект (объект, который мы можем отличить от другого), информацию о котором необходимо хранить в базе данных. Сущностями могут быть люди, места, самолеты, рейсы, вкус, цвет и т.д. Необходимо различать такие понятия, как **тип сущности** и **экземпляр сущности.** Понятие тип сущности относится к набору однородных личностей, предметов, событий или идей, выступающих как целое. Экземпляр сущности относится к конкретной вещи в наборе.

**Атрибут** – поименованная характеристика сущности. Его наименование должно быть уникальным для конкретного типа сущности, но может быть одинаковым для различного типа сущностей. Атрибуты используются для определения того, какая информация должна быть собрана о сущности.

**Ключ** – минимальный набор атрибутов, по значениям которых можно однозначно найти требуемый экземпляр сущности. Минимальность означает, что исключение из набора любого атрибута не позволяет идентифицировать сущность по оставшимся [26].

3.3.1 Разработка логической модели

Для создания логической модели БД, нацеленной на использование реляционной СУБД, нужно выполнить следующие действия:

* Создать по одной таблице для каждой сущности. Имя сущности становится именем таблицы. Каждый атрибут становится столбцом таблицы (полем) с тем же именем. Ключевой атрибут сущности становится первичным ключом таблицы.
* Задать внешний ключ для каждой сущности, вступающей во взаимоотношения с другими сущностями как «один-ко- многим» (со стороны многие).

Логическая модель представлена в приложении Б.

3.3.2 Описание сущностей

Сущность – это любой различимый объект (объект, который можем отличить от другого), информацию о котором необходимо хранить в БД. Сущностями могут быть люди, места, самолеты, рейсы, вкус, цвет и т.д. Необходимо различать такие понятия, как тип сущности и экземпляр сущности. Понятие тип сущности относится к набору однородных личностей, предметов, событий или идей, выступающих как целое. Экземпляр сущности относится к конкретной вещи в наборе. Например, типом сущности может быть недвижимость, а экземпляром – дом, квартира и т.д.

В ходе выполнения данной работы были спроектированы данные типы сущностей представлены в таблицах 3.1 – 3.12. Сущность «User» была создана автоматически Identity, поэтому она была расширена следующими атрибутами.

Таблица 3.1 – Сущность «User»

|  |  |
| --- | --- |
| Наименование атрибутов | Описание |
| FirstName | Имя пользователя |
| LastName | Фамилия пользователя |
| UsernameChangeLimit | Лимит изменения Логина |
| NumderReadTicket | Номер читательского билета |
| ProfilePicture | Изображение пользователя |
| GroupNumber | Номер читательского билета |

Таблица 3.2 – Сущность «Book»

|  |  |
| --- | --- |
| Наименование атрибутов | Описание |
| Id | Идентификатор |
| Name | Название книги |
| Author | Автор |
| Description | Описание |
| Year | Год выпуска |
| Articl | Артикул |
| BookPicture | Изображаение книги |
| Count | Количество книг |

Таблица 3.3 – Сущность «Loan»

|  |  |
| --- | --- |
| Наименование атрибутов | Описание |
| Id | Идентификатор |
| BookId | Идентификатор книги |
| UserId | Идентификатор пользователя |
| StatusId | Идентификатор статуса |
| Date | Дата заявки |

Таблица 3.4 – Сущность «Status»

|  |  |
| --- | --- |
| Наименование атрибутов | Описание |
| Id | Идентификатор |
| StatusName | Наименование статуса |

Таблица 3.5 – Сущность «BookDisc»

|  |  |
| --- | --- |
| Наименование атрибутов | Описание |
| Id | Идентификатор |
| BookId | Идентификатор книги |
| DisciplineId | Идентификатор дисциплины |

Таблица 3.6 – Сущность «Discipline»

|  |  |
| --- | --- |
| Наименование атрибутов | Описание |
| Id | Идентификатор |
| NameDiscipline | Наименование дисциплины |

Таблица 3.7 – Сущность «BookGenre»

|  |  |
| --- | --- |
| Наименование атрибутов | Описание |
| Id | Идентификатор |
| BookId | Идентификатор книги |
| GenreId | Идентификатор жанров |

Таблица 3.8 – Сущность «Genre»

|  |  |
| --- | --- |
| Наименование атрибутов | Описание |
| Id | Идентификатор |
| GenreName | Наименование жанра |

Таблица 3.9 – Сущность «BookSpec»

|  |  |
| --- | --- |
| Наименование атрибутов | Описание |
| Id | Идентификатор |
| BookId | Идентификатор книги |
| SpecialtyId | Идентификатор специальности |

Таблица 3.10 – Сущность «Specialty»

|  |  |
| --- | --- |
| Наименование атрибутов | Описание |
| Id | Идентификатор |
| NameSpecialty | Наименование специальности |

Таблица 3.11 – Сущность «Chapter»

|  |  |
| --- | --- |
| Наименование атрибутов | Описание |
| Id | Идентификатор |
| BookId | Идентификатор книги |
| NumberChapter | Номер главы |
| Name | Название главы |

Таблица 3.12 – Сущность «Work»

|  |  |
| --- | --- |
| Наименование атрибутов | Описание |
| Id | Идентификатор |
| Key | Ключ |
| Value | Значение |

3.3.3 Описание связей

**Связь** – взаимосвязь между сущностями в предметной области. Связи представляют собой соединения между записями таблиц.

Связь устанавливается между двумя общими полями (столбцами) двух таблиц. Существуют связи с отношением «*один к одному*», «*один ко многим*» и «*многие ко многим*».

Отношения, которые могут существовать между записями двух таблиц:

- *один к одному* (1:1), каждой записи из одной таблицы соответствует одна запись в другой таблице;

- *один ко многим* (1:М), каждой записи из одной таблицы соответствует несколько записей  другой таблице;

- *многие ко многим* (М:М), множеству записей из одной таблицы соответствует несколько записей в другой таблице.

Тип отношения в создаваемой связи зависит от способа определения связываемых полей:

1. Отношение «*один ко многим*» создается в том случае, когда только одно из полей является полем первичного ключа или уникального индекса.
2. Отношение «*один к одному*» создается в том случае, когда оба связываемых поля являются ключевыми или имеют уникальные индексы.
3. Отношение «*многие ко многим*» фактически является двумя отношениями «один-ко-многим» с третьей таблицей, первичный ключ которой состоит из полей внешнего ключа двух других таблиц [26].

Отношения между сущностями приведены в таблице 3.13.

Таблица 3.13 – Отношения между таблицами

|  |  |  |  |
| --- | --- | --- | --- |
| Номер связи | Родительская таблица | Дочерняя таблица | Тип связи |
| 1 | Book | Chapter | 1:M |
| 2 | Status | Loan | 1:M |
| 3 | Book | Loan | 1:M |
| 4 | Discipline | BookDisc | 1:M |
| 5 | Book | BookDisc | 1:M |
| 6 | Genre | BookGenre | 1:M |
| 7 | Book | BookGenre | 1:M |
| 8 | Specialty | BookSpec | 1:M |
| 9 | Book | BookSpec | 1:M |
| 10 | User | Loan | 1:M |
| 11 | Work |  |  |
| 12 | User | Role-User | 1:M |
| 13 | Role | Role-User | 1:M |

3.3.4 ER-диаграмма

Одним из наиболее удобных инструментов унифицированного представления данных, независимого от реализующего его программного обеспечения, является модель "сущность-связь" (ER - model).

ER-модель используется при высокоуровневом (концептуальном) проектировании баз данных. С её помощью можно выделить ключевые сущности и обозначить связи, которые могут устанавливаться между этими сущностями.

Во время проектирования баз данных происходит преобразование ER-модели в конкретную схему базы данных на основе выбранной модели данных (реляционной, объектной, сетевой или др.).

ER-модель представляет собой формальную конструкцию, которая сама по себе не предписывает никаких графических средств её визуализации. В качестве стандартной графической нотации, с помощью которой можно визуализировать ER-модель, была предложена **диаграмма сущность-связь (ER-диаграмма)**.

Понятия ER-модель и ER-диаграмма часто ошибочно не различают, хотя для визуализации ER-моделей предложены и другие графические нотации [27].

Таблица 3.14 – Обозначения на ER-диаграммах



Инфологическая модель данных представлена в приложении В.

3.4 Физическая модель

Физическая модель отображает состав таблиц БД. Для каждого поля таблицы указывается размер поля (количество символов), тип. Для первичных ключей необходимо ввести запрет неопределенных значений. Для остальных полей возможность запрета неопределенных значений определяется семантикой предметной области.

Таблица 3.15 – Атрибуты таблицы «User»

|  |  |  |  |
| --- | --- | --- | --- |
| Имя столбца | Тип данных | Разрешить значение NULL | Тип ключа |
| FirstName | nvarchar(max) | NULL |  |
| LastName | nvarchar(max) | NULL |  |
| UsernameChangeLimit | int | NOT NULL |  |
| NumderReadTicket | nvarchar(max) | NULL |  |
| ProfilePicture | nvarchar(max) | NULL |  |
| GroupNumber | nvarchar(max) | NULL |  |

Таблица 3.16 – Атрибуты таблицы «Status»

|  |  |  |  |
| --- | --- | --- | --- |
| Имя столбца | Тип данных | Разрешить значение NULL | Тип ключа |
| Id | int | NOT NULL | Первичный ключ |
| StatusName | nvarchar(max) | NULL |  |

Таблица 3.17 – Атрибуты таблицы «Book»

|  |  |  |  |
| --- | --- | --- | --- |
| Имя столбца | Тип данных | Разрешить значение NULL | Тип ключа |
| Id | int | NOT NULL | Первичный ключ |
| Name | nvarchar(max) | NULL |  |
| Author | nvarchar(max) | NULL |  |
| Description | nvarchar(max) | NULL |  |
| Year | int | NOT NULL |  |
| Articl | nvarchar(max) | NULL |  |
| BookPicture | nvarchar(max) | NULL |  |
| Count | int | NOT NULL |  |

Таблица 3.18 – Атрибуты таблицы «BookDisc»

|  |  |  |  |
| --- | --- | --- | --- |
| Имя столбца | Тип данных | Разрешить значение NULL | Тип ключа |
| Id | int | NOT NULL | Первичный ключ |
| BookId | int | NOT NULL | Внешний ключ |
| DisciplineId | int | NOT NULL | Внешний ключ |

Таблица 3.19 – Атрибуты таблицы «Loan»

|  |  |  |  |
| --- | --- | --- | --- |
| Имя столбца | Тип данных | Разрешить значение NULL | Тип ключа |
| Id | int | NOT NULL | Первичный ключ |
| BookId | int | NOT NULL | Внешний ключ |
| UserId | int | NOT NULL | Внешний ключ |
| StatusId | int | NOT NULL | Внешний ключ |
| Date | datetime2(7) | NOT NULL |  |

Таблица 3.20 – Атрибуты таблицы «Discipline»

|  |  |  |  |
| --- | --- | --- | --- |
| Имя столбца | Тип данных | Разрешить значение NULL | Тип ключа |
| Id | int | NOT NULL | Первичный ключ |
| NameDiscipline | nvarchar(max) | NULL |  |

Таблица 3.21 – Атрибуты таблицы «Genre»

|  |  |  |  |
| --- | --- | --- | --- |
| Имя столбца | Тип данных | Разрешить значение NULL | Тип ключа |
| Id | int | NOT NULL | Первичный ключ |
| GenreName | nvarchar(max) | NULL |  |

Таблица 3.22 – Атрибуты таблицы «BookGenre»

|  |  |  |  |
| --- | --- | --- | --- |
| Имя столбца | Тип данных | Разрешить значение NULL | Тип ключа |
| Id | int | NOT NULL | Первичный ключ |
| BookId | int | NOT NULL | Внешний ключ |
| GenreId | int | NOT NULL | Внешний ключ |

Таблица 3.23 – Атрибуты таблицы «BookSpec»

|  |  |  |  |
| --- | --- | --- | --- |
| Имя столбца | Тип данных | Разрешить значение NULL | Тип ключа |
| Id | int | NOT NULL | Первичный ключ |
| BookId | int | NOT NULL | Внешний ключ |
| SpecialtyId | int | NOT NULL | Внешний ключ |

Таблица 3.24 – Атрибуты таблицы «Specialty»

|  |  |  |  |
| --- | --- | --- | --- |
| Имя столбца | Тип данных | Разрешить значение NULL | Тип ключа |
| Id | int | NOT NULL | Первичный ключ |
| NameSpecialty | nvarchar(max) | NULL |  |

Таблица 3.25 – Атрибуты таблицы «Chapter»

|  |  |  |  |
| --- | --- | --- | --- |
| Имя столбца | Тип данных | Разрешить значение NULL | Тип ключа |
| Id | int | NOT NULL | Первичный ключ |
| BookId | int | NOT NULL | Внешний ключ |
| NumberChapter | int | NOT NULL |  |
| Name | nvarchar(max) | NULL |  |

Таблица 3.26 – Атрибуты таблицы «Work»

|  |  |  |  |
| --- | --- | --- | --- |
| Имя столбца | Тип данных | Разрешить значение NULL | Тип ключа |
| Id | int | NOT NULL | Первичный ключ |
| Key | nvarchar(max) | NULL |  |
| Value | nvarchar(max) | NULL |  |

3.4.1 Диаграмма связи по полям

**Схема (диаграмма)** **базы данных** − структура базы данных, описанная на формальном языке, поддерживаемом СУБД. В реляционных базах данных схема определяет таблицы, поля в каждой таблице (обычно с указанием их названия, типа, обязательности), и ограничения целостности (первичный, потенциальные и внешние ключи и другие ограничения) [28].

Схемы в общем случае хранятся в словаре данных. Хотя схема определена на языке базы данных в виде текста, термин часто используется для обозначения графического представления структуры базы данных.

Основными объектами графического представления схемы являются таблицы и связи, определяемые внешними ключами.

Диаграмма базы данных представлена в приложении Г.

3.5 Описание структурных частей приложения

Приложение разделяется на три основных компонента, каждый из которых отвечает за различные задачи (принцип единой ответственности).

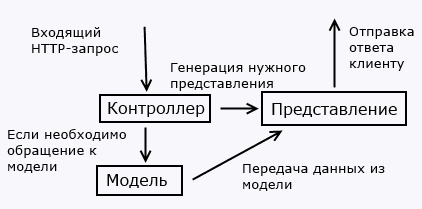
Контроллер управляет запросами пользователя (получаемые в виде запросов HTTP GET, POST или PUT, когда пользователь нажимает на элементы интерфейса для выполнения различных действий). Его основная функция - вызывать и координировать действие необходимых ресурсов и объектов, нужных для выполнения действий, задаваемых пользователем [29].

Модель — это данные и правила, которые используются для работы с данными, которые представляют концепцию управления приложением. Модель дает контроллеру представление данных, которые запросил пользователь.

Модель содержит наиболее важную часть логики нашего приложения, логики, которая решает задачу, с которой мы имеем дело. Контроллер содержит в основном организационную логику для самого приложения.

Вид обеспечивает различные способы представления данных, которые получены из модели. Он может быть шаблоном, который заполняется данными. Может быть несколько различных видов, и контроллер выбирает, какой подходит наилучшим образом для текущей ситуации.

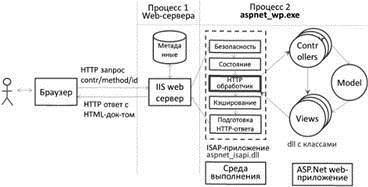
Приложение состоит из набора контроллеров, моделей и видов. Идея, лежащая в основе управления работы приложения представлена на рисунке 3.1.



**Рисунок 3.1 – Управление потоком данных в приложении**

В кратком виде обработка запроса в приложении выглядит следующим образом (рисунок 3.2):

* На сервер приходит запрос от пользователя;
* IIS запускает ASP.NET Core модуль, который будет запускать приложение;
* Система маршрутизации определяет каким контроллером должен обрабатываться запрос;
* Выполняется нужный Action (действие);
* Воспроизводит View и отдаѐтся пользователю.



**Рисунок 3.2 – Обработка запроса в приложении**

Проект приложения содержит простую структуру: проект приложения, проект базы данных, проект для тестирования приложения. Для .Net Core - необходимый минимум для запуска приложения является наличие:

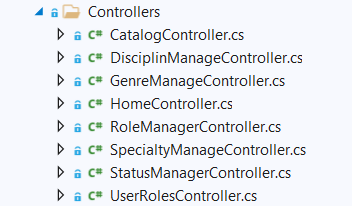
* Connected Services: подключенные сервисы из Azure.
* Dependencies: все добавленные в проект пакеты и библиотеки, иначе говоря зависимости.
* Properties: узел, который содержит некоторые настройки проекта. В
* частности, в файле launchSettings.json описаны настройки запуска проекта, например, адреса, по которым будет запускаться приложение.
* appsettings.json: файл конфигурации проекта в формате json.
* Program.cs: главный файл приложения, с которого и начинается его выполнение. Код этого файла настривает и запускает веб-хост, в рамках которого разворачивается приложение.
* Startup.cs: файл, который определяет класс Startup и который содержит
* логику обработки входящих запросов.
* Controllers – узел, содержащий контроллеры проекта.
* Migrations – узел, содержащий миграции(изменения) базы данных.
* Models – узел, содержащий модели (сущности) проекта.
* Views – узел, содержащий представления (визуализацию) проектных элементов.
* wwwroot – узел, содержащий статические файлы проекта.

3.5.1 Организация и управление данными

Центральным звеном в архитектуре ASP.NET Core MVC является контроллер. При получении запроса система маршрутизации выбирает для обработки запроса нужный контроллер и передает ему данные запроса. Контроллер обрабатывает эти данные и посылает обратно результат обработки.

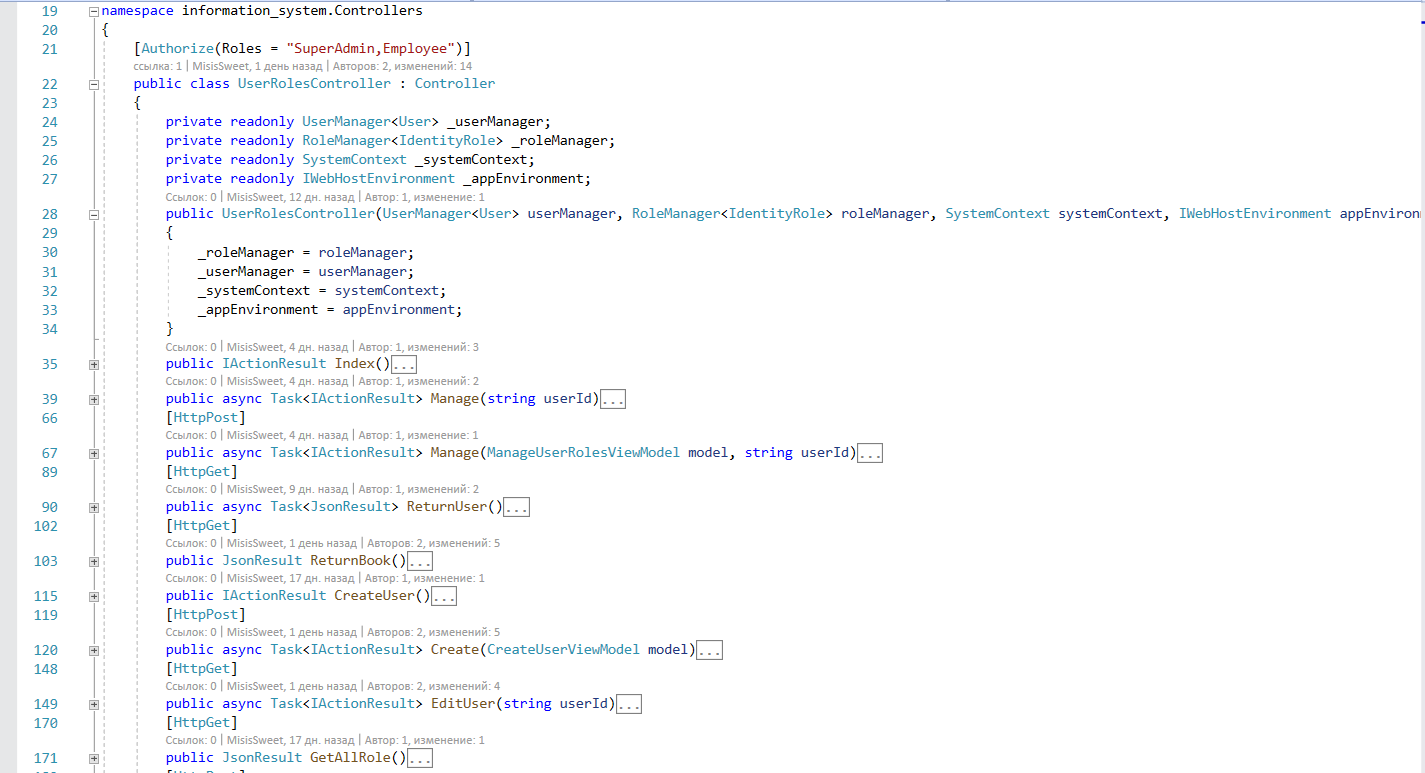
В ASP.NET Core MVC контроллер представляет обычный класс на языке C#, который наследуется от абстрактного базового класса Microsoft.AspNetCore.Mvc.Controller. По умолчанию проект ASP.NET Core MVC содержит как минимум один контроллер – HomeController.

Контроллеры в приложении показаны на рисунке 3.3.



**Рисунок 3.3 – Контроллеры приложения**

Пример структуры контроллера «UserRolesController», представлен на рисунке 3.4 с базовым минимумом методов, соответствующих представлениям приложения: Index, CreateBook, EditBook, CreateUser, EditUser, EditP, Manage, ManageDiscipline, ManageSpecialty, ManageGenre, ManageStatus. Так же есть методы, которые получают данные и возвращают их через Get запрос на страницу.

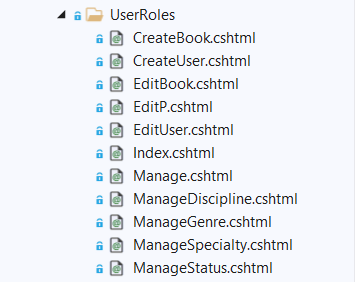


**Рисунок 3.4 – Структура контролера на примере «UserRolesController»**

В моделях проекта имеется классы, которые соответствуют описанию сущностям базы данных.

Представления отвечают за отображение контекста в пользовательском интерфейсе. Они используют [инструмент представления Razor](https://dotnet.today/ru/aspnet5-vnext/mvc/overview.html#razor), чтобы включить .NET код в HTML разметку. В представлениях должно быть по минимуму логики, и эта логика должна касаться отображаемого контента.

В каждом представлении имеется набор отображений, которые представлены на рисунке 3.5.

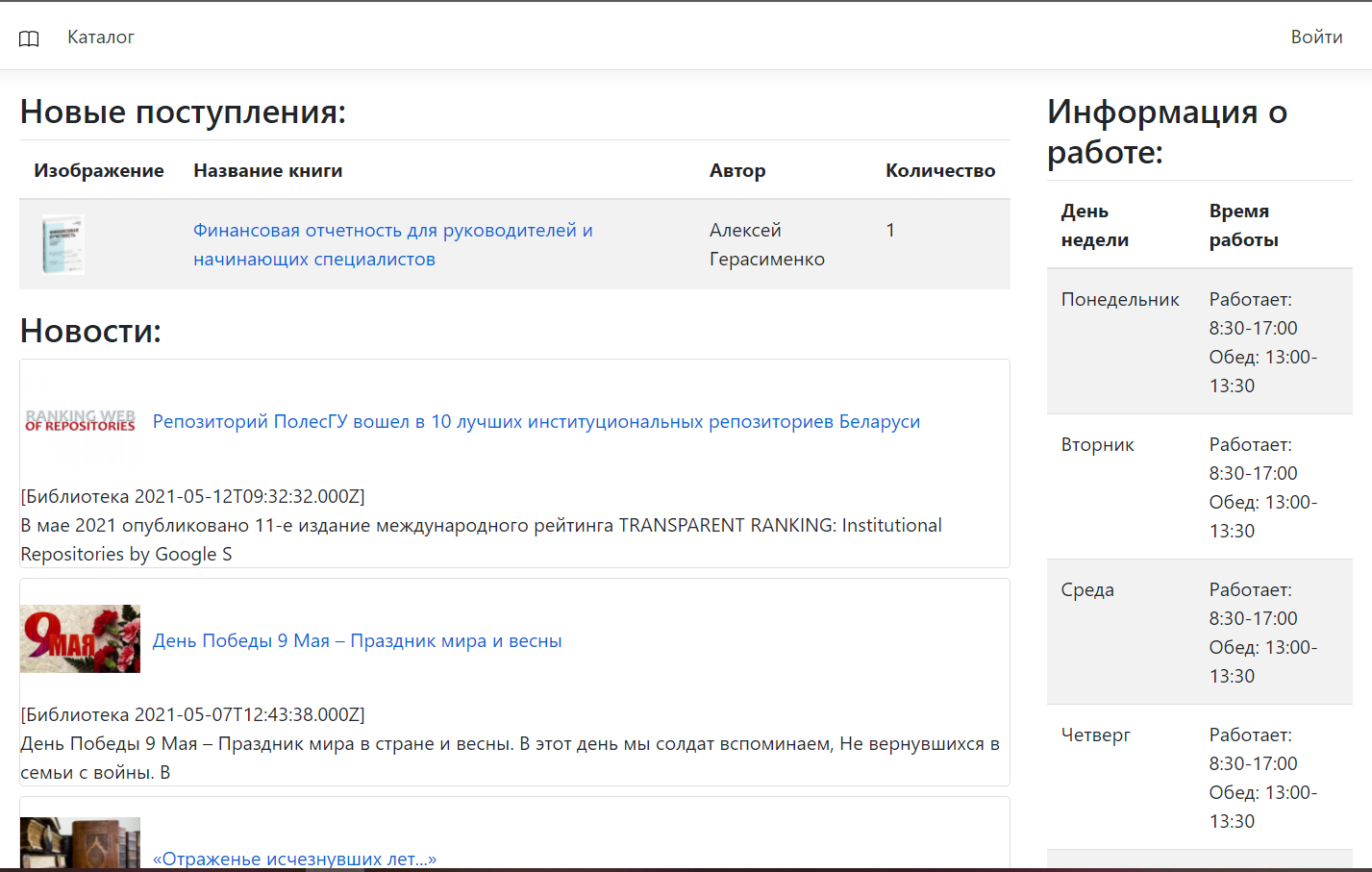


**Рисунок 3.5 – Набор отображений соответствующего контроллера**

3.5.2 Интерфейс информационной системы

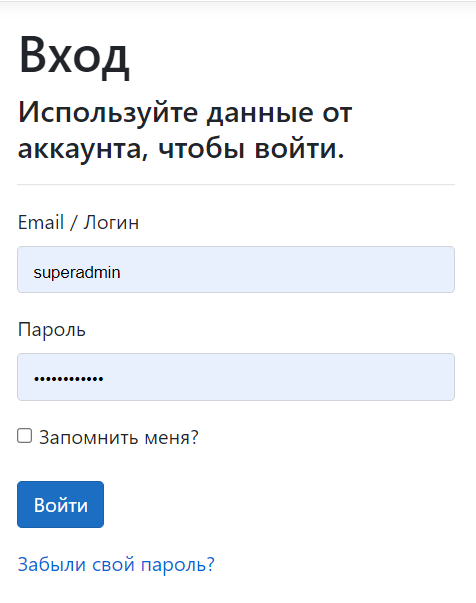
Интерфейс систем разработан с помощью библиотеки Bootstrap. Учитывая тот факт, что разрабатываемое программное приложение не является веб-сайтом, а представляет собой веб-приложение с клиентской частью в браузере, его интерфейс будет содержать поля для ввода, отображения, корректировки данных в стилях библиотеки.

Пользователь, попадая на сайт видит главную страницу, на которой представлены: Новые поступление, информация о работе, новости. На шапке сайта имеются несколько ссылок: каталог, кнопка войти и изображения открытой книги для перехода на главную страницу. В новых поступлениях показываются 5 новых книг по дате добавления. Новости парсятся из RSS на JS с сайта ПолесГУ. В информации о работе показано время работы библиотеки и санитарные дни. Главная страница представлена на рисунке 3.6.



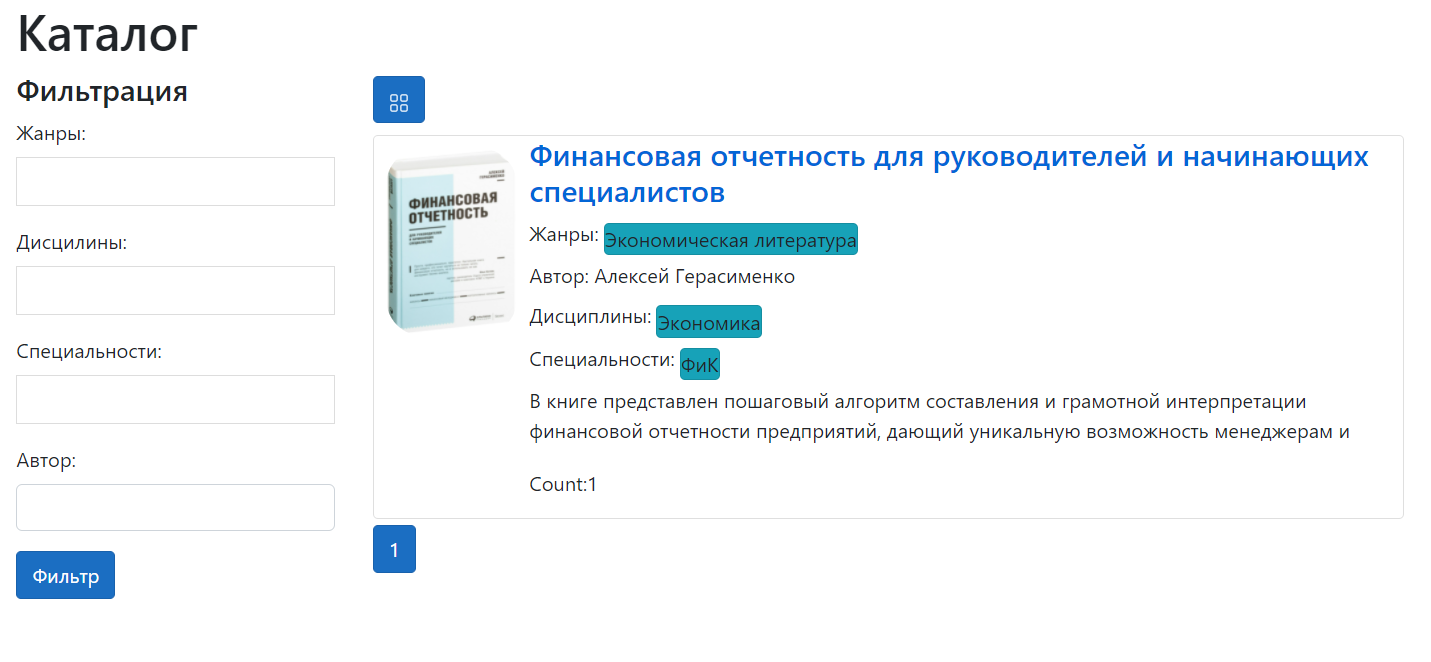
**Рисунок 3.6 – Главная страница приложения**

При нажатии на кнопку войти, необходимо ввести свой логин или email и пароль, есть поле запомнить пароль. После введения данных нажать кнопку «Вход» и, если всё успешно будет произведено перенаправления на главную страницу. Если пользователь забыл пароль его есть возможность восстановить через email. Страница Входа изображена на рисунке 3.7.



**Рисунок 3.7 – Страница входа**

Пользователю доступна ссылка на каталог, в котором он может просмотреть список книг. В каталоге можно отфильтровать книги по жанрам, дисциплинам, специальностям и авторам. Книги можно просмотреть в двух форматах: ввиду таблицы и в виде списка. Страница каталога представлена на рисунке 3.8.



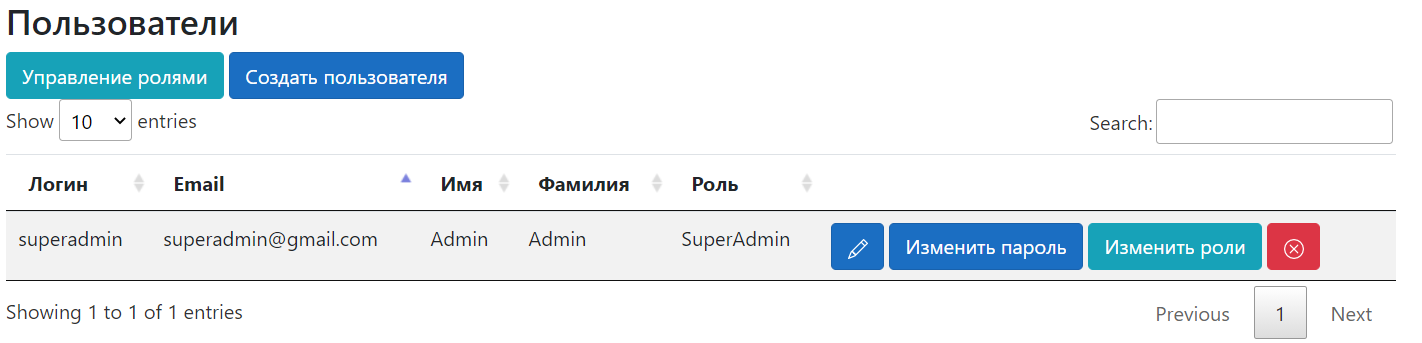
**Рисунок 3.8 – Страница каталога**

Если мы входим под SuperAdminim, то вверху страницы появляется дополнительная кнопка «Управление» представлена на рисунке 3.9.

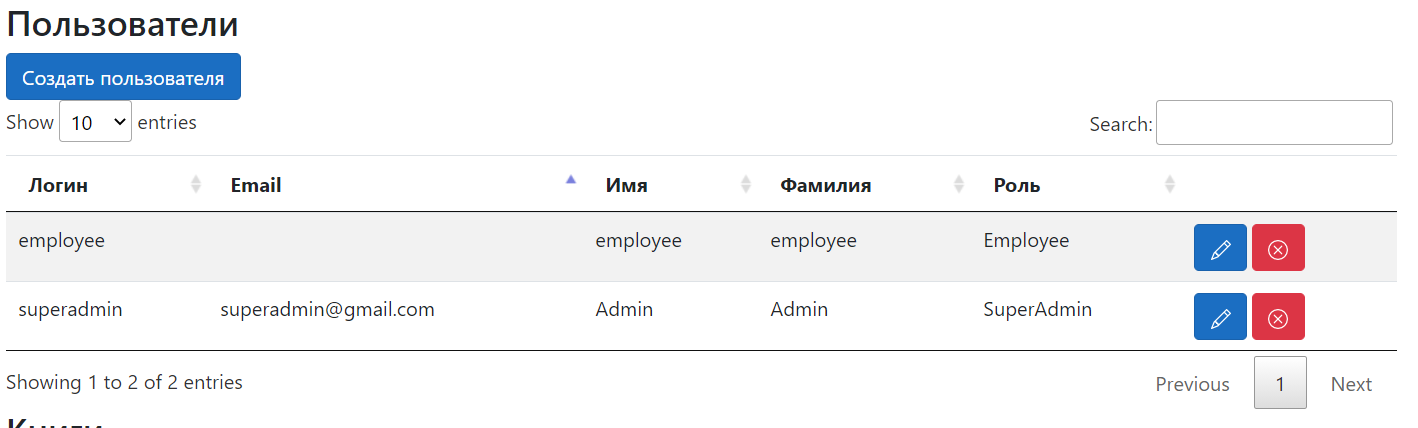


**Рисунок 3.9 – Шапка сайта с новыми параметрами**

На странице управления у админа есть три поля: Пользователи, Книги и заявки. Поле пользователи представлено на рисунке 3.10.1-3.10.2. На данном поле отображаются пользователи, Супер админ имеет доступ на создание новых ролей, назначению роли пользователю. Супер админ и сотрудник имеют доступ к созданию, редактированию и удалению пользователей. Супер админ также имеет возможность изменять логин и пароль пользователей.

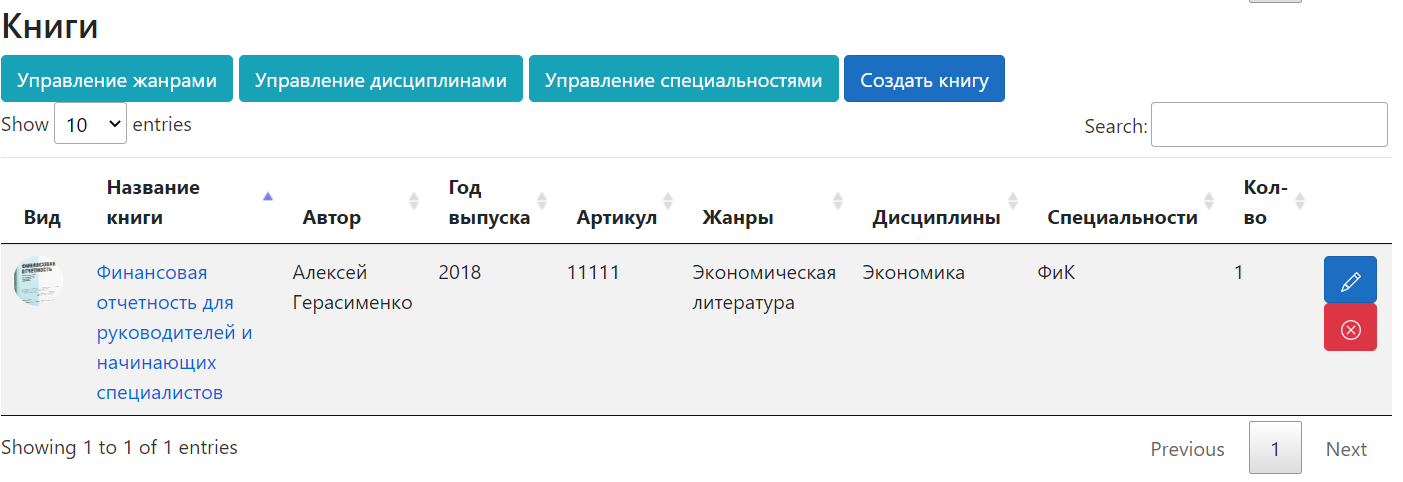


**Рисунок 3.10.1 – Панель управления пользователями SuperAdmin**

****

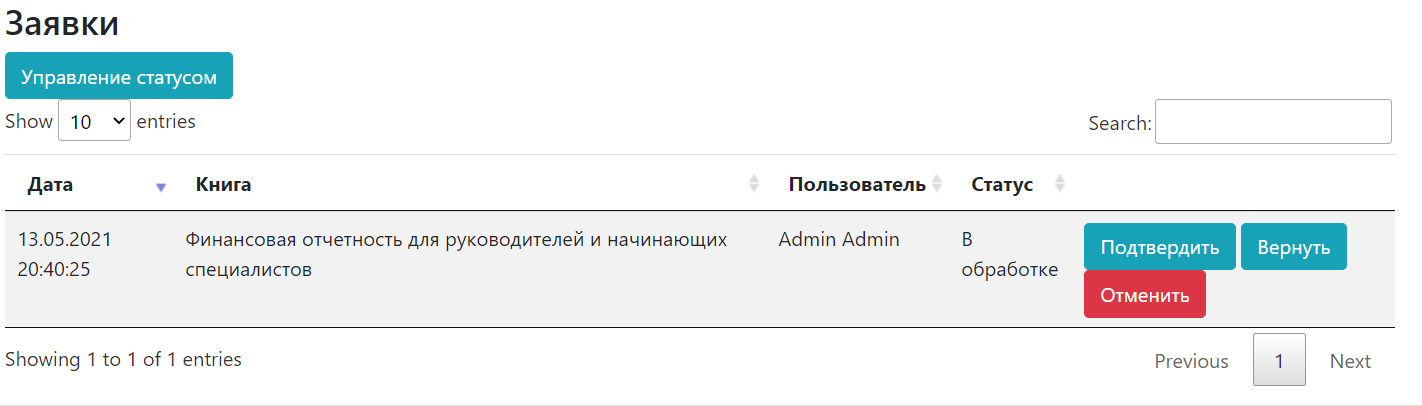
**Рисунок 3.10.2 – Панель управления пользователями Employee**

Отображение панели управления книгами изображено на рисунке 3.11. На данной панели можно управлять жанрами, дисциплинами и специальностями, создавать, редактировать и удалять книги.



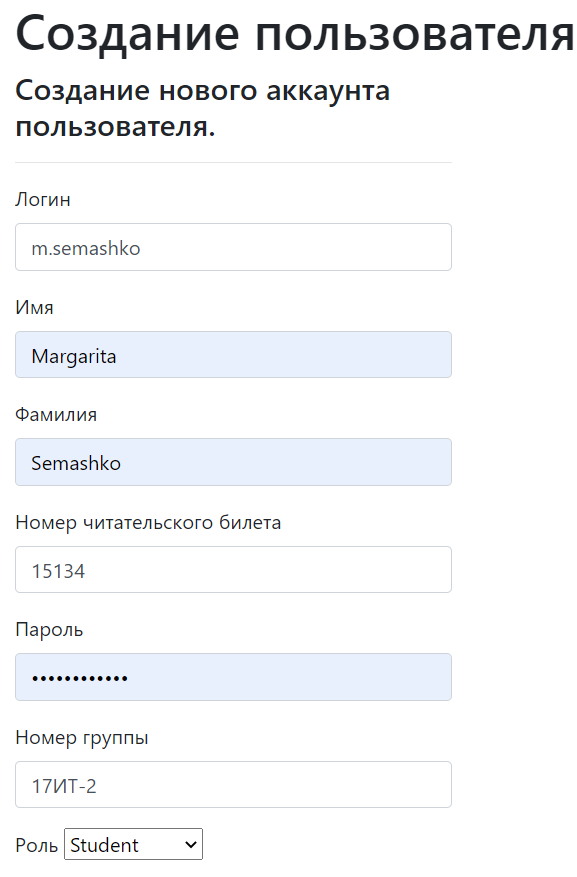
**Рисунок 3.11 – Панель управления книгами**

Так же есть панель управления заявками, она показана на рисунке 3.12. У каждой заявки есть свой статус, который можно изменить. Администратор может принять, отклонить заявку или если заявка принята, то по возвращению книги перевести заявку в статус «Возвращено».



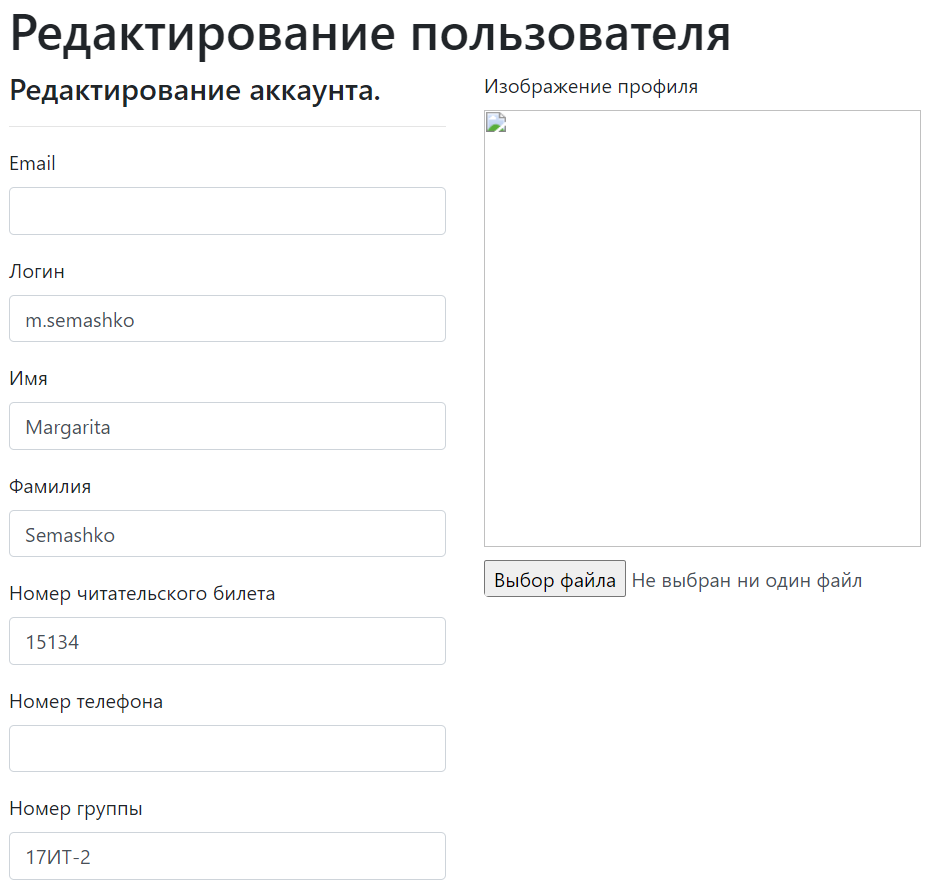
**Рисунок 3.12 – Панель управления заявками**

При нажатии на кнопку «Создать пользователя» администратор или сотрудник перемещаются на страницу создания пользователя (рисунок 3.13.). Администратор заполняет такие данные как: Логин, Имя, Фамилия, Номер читательского билета, Пароль, Номер группы и присваивает роль пользователю, после чего нажимает кнопку «Создать». Новый пользователь отображается в таблице на панели управления пользователями (рисунок 3.10). В данной системе, пользователя создавать может только администратор или сотрудник, сам пользователь не имеет доступа на регистрацию.



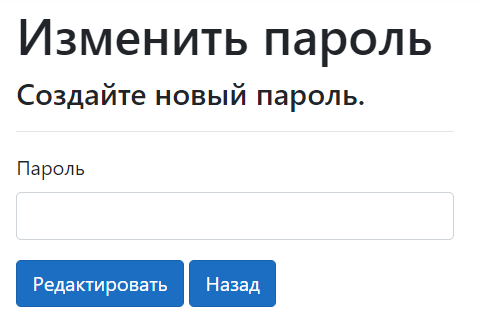
**Рисунок 3.13 – Создание пользователя**

Когда администратор нажимает кнопку «Редактировать» (она обозначена на панели карандашом), то переходит на страницу редактирования пользователя (рисунок 3.14). На данной странице представлены следующие поля: Email, логин, Имя, Фамилия, Номер читательского билета, группа, номер телефона, а также можно установить картинку. При изменении данные администратор нажимает кнопку «Редактировать».



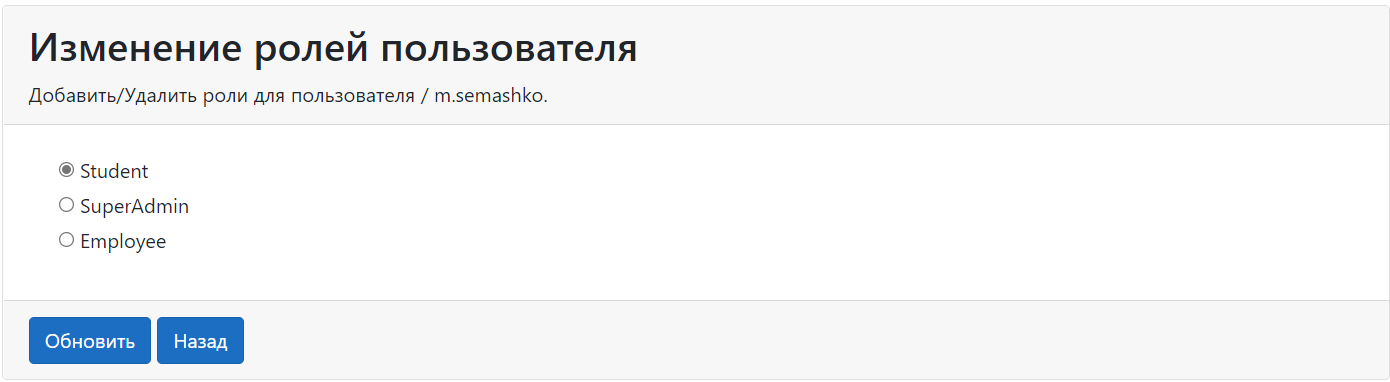
**Рисунок 3.14 – Редактирование пользователя**

Т.к. Логин можно изменить на странице редактирование пользователя, то для пароля существует отдельная страница (рисунок 3.15). Администратор вводит новый пароль и нажимает кнопку «Редактировать».



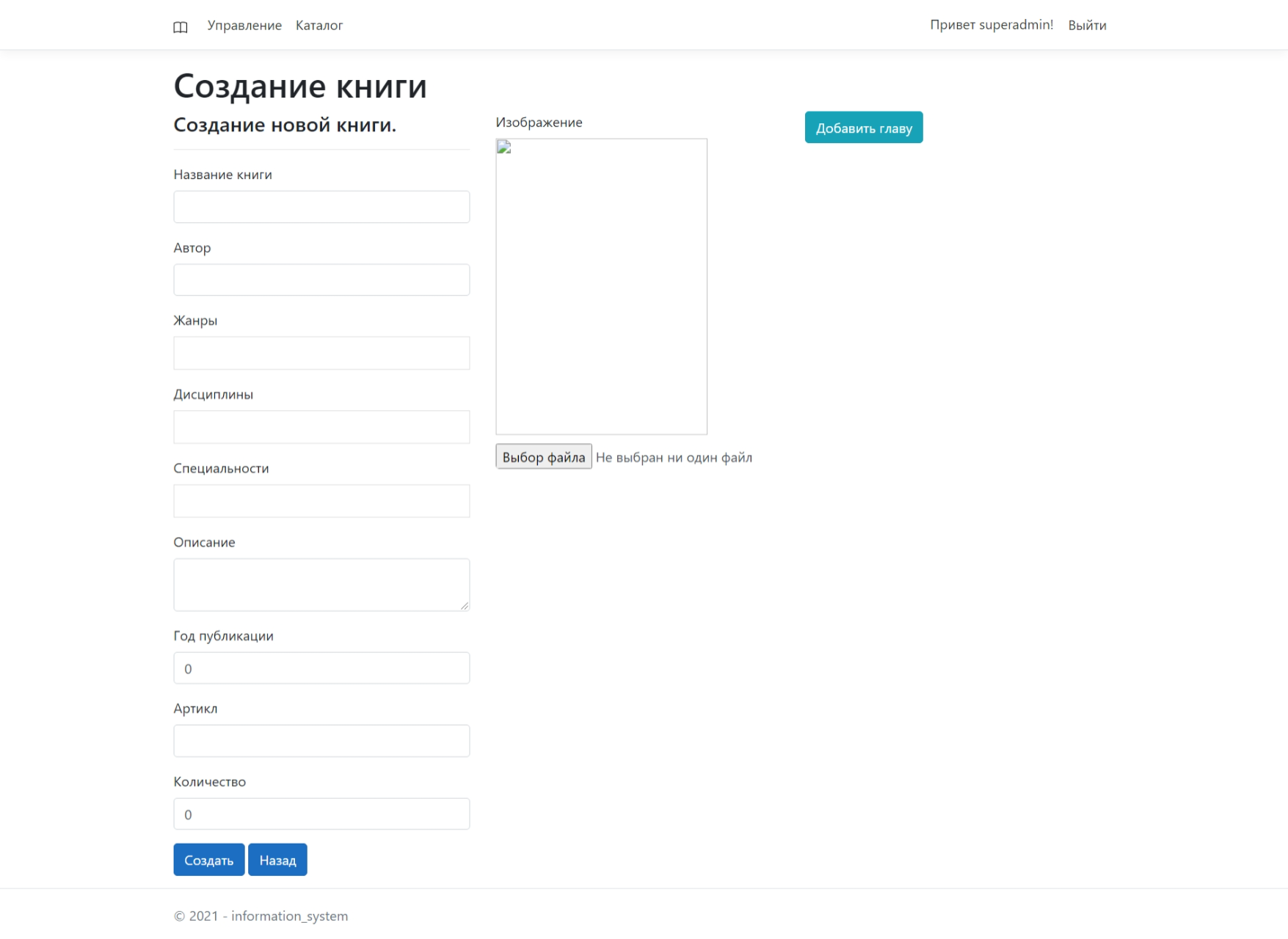
**Рисунок 3.15 – Изменение пароля**

Также существует страница изменения ролей (рисунок 3.16). На данной странице можно изменить роль, можно выбрать одну роль из списка.



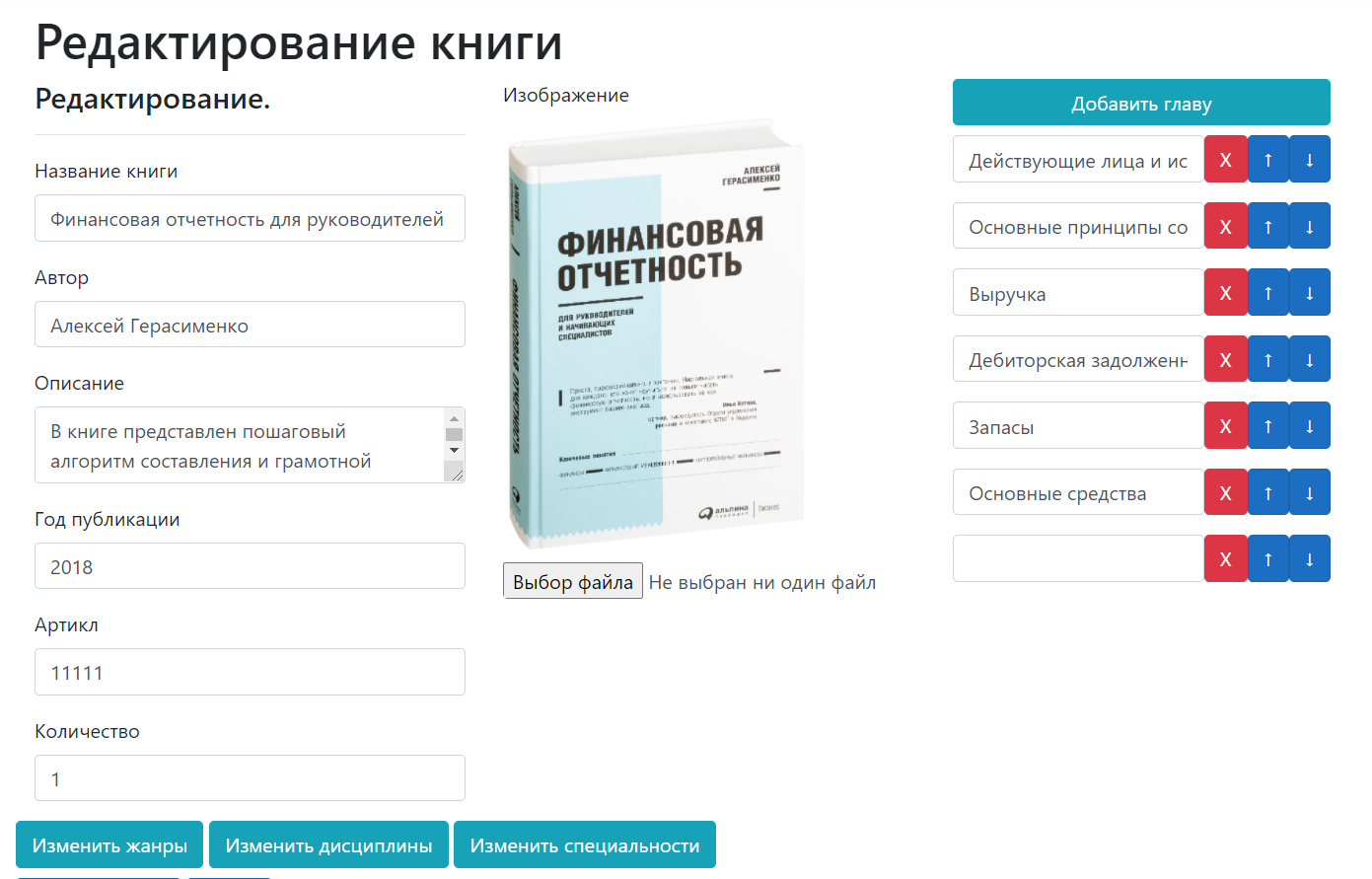
**Рисунок 3.16 – Изменение роли**

Создание книги схоже с созданием пользователя (рисунок 3.17). Имеются поля: Название книги, автор, жанры, дисциплины, специальности, описание, год публикации, артикул, количество, изображение. Так же можно нажать кнопку «Добавить главу» и появятся поля для добавления названия главы.



**Рисунок 3.17 – Создание книги**

Для того, что бы перейти к редактированию книги необкодимо нажать на кнопку «Редактировать». Страница редактирования книги идентична созданию книги, за исключением: жанров, дисциплин и специальностей (рисунок 3.18). Эти три поля вынесены на редактирование на отдельные страници и можно выбирать данные как на рисунке 3.16, но в данном случаее поддердимается мноественная выборка.

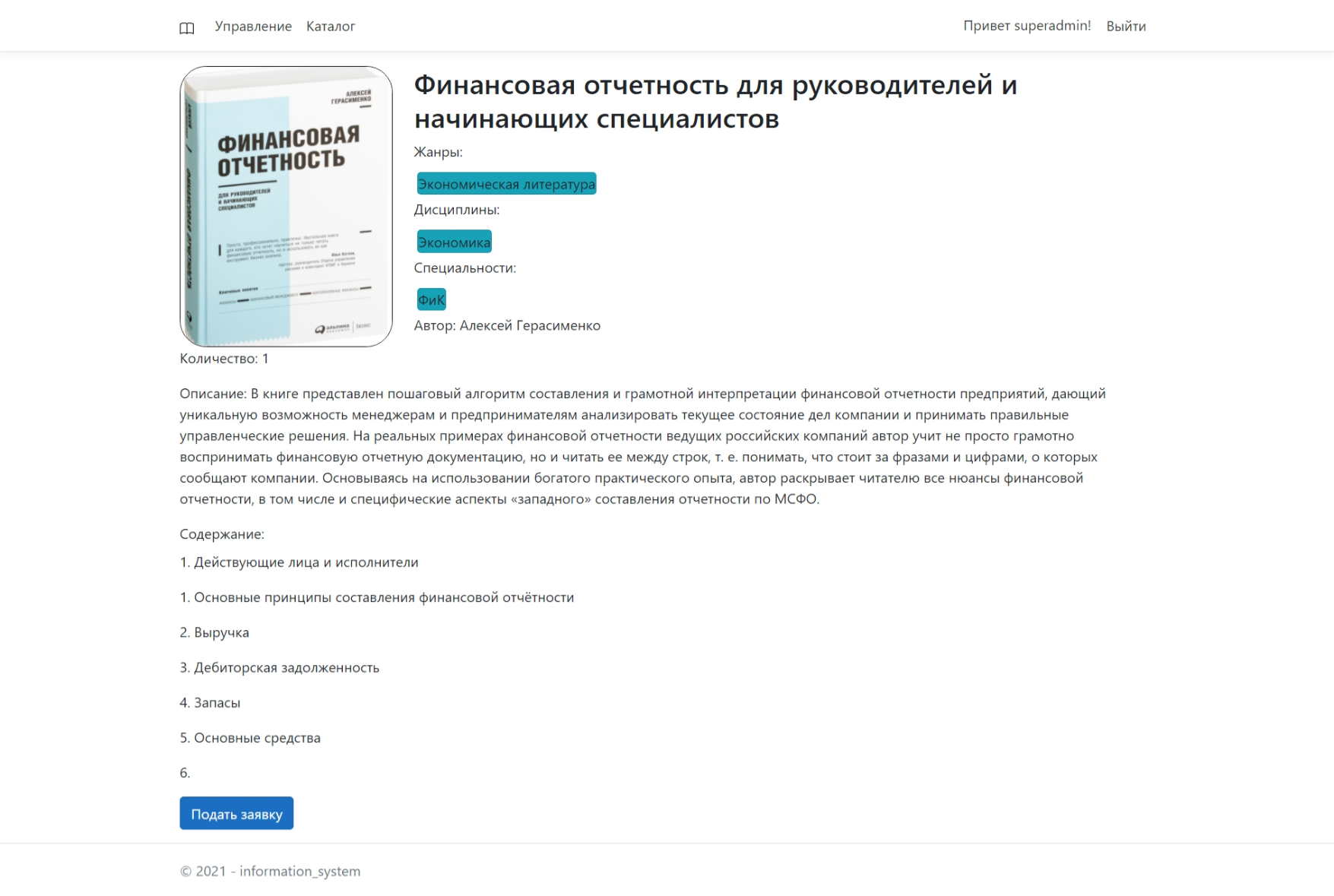


**Рисунок 3.18 – Редактирование книги**

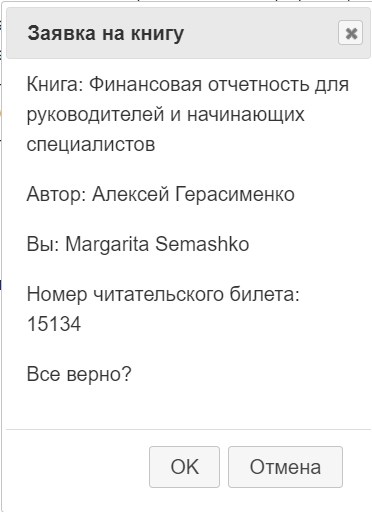
Перейти на страницу просмотра книгу можно тремя способами:

1. С главной страницы, нажав на название книги из новых поступлений;
2. Из каталога, отфильтровав книги;
3. Или с панели администратора, нажав на название книги.

Страница просмотра книги представлена на рисунке 3.19. На данной странице отображается изображение книги, название, жанры, дисциплины, специальности, автор, количество, описание и содержание, если были добавлены главы. При нажатии кнопки «Подать заявку» появляется диалоговое окно (рисунок 3.20). В диалоговом окне отображается информация о книги и о пользователе. При нажатии кнопки «Ок» создаётся заявка.

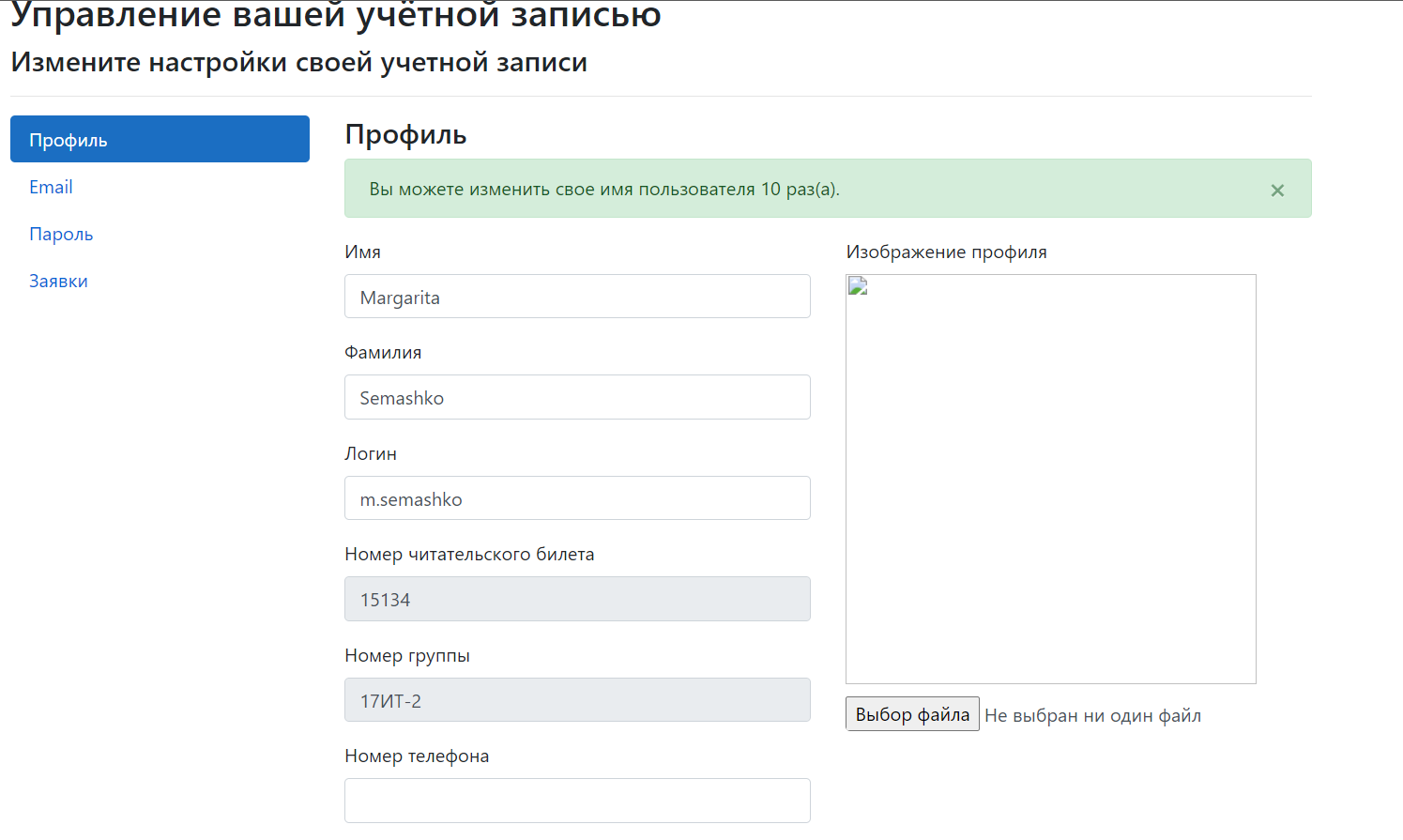


**Рисунок 3.19 – Просмотр книги**

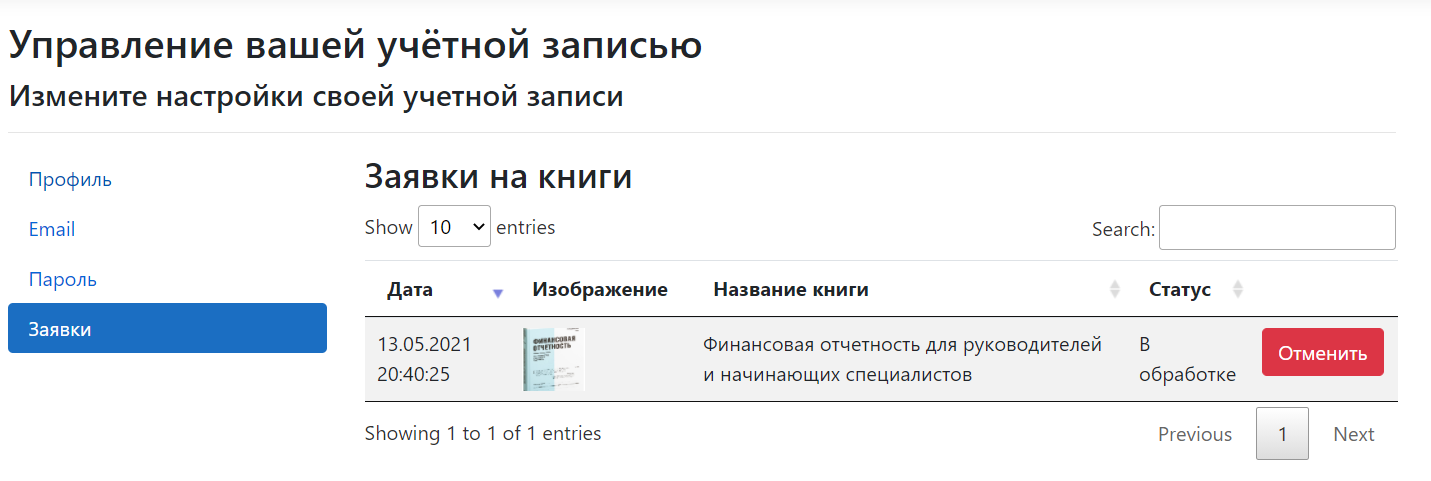


**Рисунок 3.20 – Диалоговое окно**

Пользователь свои заявки может просмотреть в профиле. Для того, чтобы перейти в профиль на шапке приложения необходимо нажать на «Привет [Логин пользователя]!». Страница пользователя представлена на рисунке 3.21. На данной странице пользователь управляет своим аккаунтом, может изменить данные, email, пароль, а также просмотреть список своих заявок (рисунок 3.22). Пока заявка в обработке пользователь может отменить её. История всех заявок хранится в панели управления заявками.



**Рисунок 3.21 – Страница пользователя**



**Рисунок 3.22 – Просмотр заявок пользователя**

3.5.3 Тестирование информационной системы

# ГЛАВА 4 ОЦЕНКА ЭКОНОМИЧЕСКОЙ ЭФФЕКТИВНОСТИ РАЗРАБОТКИ ПРОГРАММНОГО ПРИЛОЖЕНИЯ

## **Оценка трудоемкости программного продукта на основе метода вариантов использования**

## **Оценка трудоемкости программного продукта укрупненным методом**

## **Определение себестоимости создания программного продукта**

## **Расчет суммарной экономии затрат после внедрения ПП по упрощенной модели**

## **Расчет ожидаемого прироста прибыли по укрупненной модели**

## **Расчет показателей эффективности использования программного обеспечения**

# ГЛАВА 5 ОХРАНА ТРУДА ПРИ ЭКСПЛУАТАЦИИ ПРОГРАММНОГО ПРОДУКТА

5.1 Общая информация

Охрана труда –– важнейшая государственная задача, т. к. в ее основе лежит забота о здоровье и жизни людей. Значимость этой функции государства подтверждается принятым и действующим в нашей стране «Законом об охране труда».

1. Понятие охраны труда содержится в Законе Республики Беларусь от 23 июня 2008 г. «Об охране труда», а также в СТБ 18001 «Системы управления охраной труда. Общие требования» [30].

Политика в области охраны труда направлена на:

* соблюдение действующего законодательства по охране труда;
* предупреждение несчастных случаев на производстве и профессиональных заболеваний;
* улучшение условий и охраны труда;
* обучение, повышение квалификации работников в области охраны труда, подготовку персонала к локализации и ликвидации аварий;
* организацию контроля за соблюдением требований охраны труда;
* информирование работников об условиях труда, случаях производственного травматизма и профессиональных заболеваний, льготах и компенсациях по условиям труда;
* привлечение всех работников организаций, к участию в формировании и реализации политики, деятельности по улучшению условий труда, профилактике несчастных случаев и профессиональных заболеваний [31].

5.2 Охрана труда инженера-программиста

Инструкция по охране труда инженера-программиста при работе с персональным компьютером, разработана с учетом условий работы, текст приведен в приложении М и включает в себя 5 глав:

Глава 1. «Общие требования безопасности» содержит: опасные и вредные производственные факторы; условия допуска к работе.

К работе программистом допускаются: прошедшие вводный инструктаж по охране труда; прошедшие обучение безопасным приемам и методам труда и прошедшие проверку знаний, в том числе по электробезопасности; прошедшие курс обучения на персональном компьютере с использованием конкретного программного обеспечения; прошедшие инструктаж по охране труда на конкретном рабочем месте по данной инструкции.

Глава 2. «Требования безопасности перед началом работы» включает в себя обязанности инженера-программиста перед началом работы и условия, запрещающие приступать к работе.

Перед началом работы программист обязан: осмотреть и привести в порядок рабочее место; проверить правильность подключения оборудования в электросеть; протереть специальной салфеткой поверхность экрана; при включении компьютера соблюдать правила электробезопасности.

Глава 3. «Требования безопасности во время работы» отражает обязанности программиста во время работы и перечень запрещенных действий.

Во время работы запрещается: прикасаться к задней панели системного блока (процессора) при включенном питании; переключать разъемы интерфейсных кабелей периферийных устройств при включенном питании; загромождать верхние панели устройств бумагами и посторонними предметами; производить отключение питания во время выполнения активной задачи; производить частые переключения питания; допускать попадание влаги на поверхность системного блока (процессора), монитора, рабочую поверхность клавиатуры, дисководов, принтеров и других устройств.

Глава 4. «Требования безопасности в аварийных ситуациях» содержит список действий при возникновении различных аварийных и экстремальных ситуаций.

В случае обнаружения обрыва проводов питания, неисправности заземления и других повреждений электрооборудования, появления запаха гари немедленно прекратить работу, отключить питание и сообщить об аварийной ситуации руководителю.

Глава 5. «Требования безопасности по окончанию работ» содержит последовательность действий по окончанию работ.

По окончании работ программист обязан осмотреть и привести в порядок рабочее место и вымыть с мылом руки и лицо.

5.3 Охрана труда при работе на ПК

Инструкция по охране труда при работе на персональных компьютерах, в том числе на портативных, текст приведен в приложении Н и содержит схожие требования на инструкцию по охране труда инженера-программиста, однако более подробный и развернутый.

Инструкция включает в себя 5 глав:

Глава 1. «Общие требования по охране труда» содержит условия допуска к работе; особенности режима работы беременных и кормящих грудью женщин; опасные и вредные производственные факторы; обязанности работника; обязанности работника при несчастном случае на производстве; особенности поведения при нахождении в цехах; требования к освещению и рабочему месту.

Глава 2. «Требования по охране труда перед началом работы» содержит правила поведения работника до начала работы.

Перед началом работы необходимо осмотреть рабочее место и убедиться: в устойчивости положения оборудования на рабочем столе; в отсутствии видимых повреждений оборудования; в исправности и целостности питающих и соединительных кабелей, разъемов и штепсельных соединений; в исправности общего и местного освещения, мебели.

Глава 3. Требования по охране труда при выполнении работ» содержит: обязанности работника при выполнении работ; регламентированные перерывы; рекомендации по проведению перерывов; запрещенные действия при работе с ПК.

Глава 4. «Требования по охране труда по окончанию работы» содержит перечень действий после завершения работы.

По окончанию работы работник обязан выключить ПК; осмотреть и привести в порядок рабочее место; при необходимости протереть поверхность периферийных устройств.

Глава 5. «Требования по охране труда в аварийных ситуациях» содержит правила поведения работника при возникновении аварийных ситуаций.

При любых случаях сбоя в работе оборудования необходимо прекратить работу и вызвать специалиста, осуществляющего техническое обслуживание оборудования.

# ЗАКЛЮЧЕНИЕ

В результате работы над дипломным проектом были решены следующие задачи:

* исследована предметная область;
* разработана база данных;
* разработано программное приложение;
* рассчитаны показатели экономической эффективности программного приложения;
* протестирована автоматизированная система;
* сформулированы требования охраны труда;

На основании построенных моделей была создана база данных, хранящая необходимые для работы программного приложения данные.

Для описания основных принципов работы программного обеспечения были разработаны бизнес требования, составлены UML-диаграммы, диаграммы IDEF-нотации, которые более детально показывают логику работы приложения.

По результатам моделирования было создано программное приложение, которое полностью реализует все поставленные перед ним задачи.

Интерфейс программы простой, понятный, удобный в использовании, что исключает возможные затруднения при работе пользователя с данной программой.

Основные преимущества, которыми обладает разработанная система: снижение затрат времени на работу с приложением; широкий охват клиентской базы.

Данный дипломный проект является вполне конкурентоспособным на отечественном рынке. Это приводит к необходимости доработки и усовершенствования программы. При правильно построенной работе и усовершенствование программы позволит получить готовый и уникальный продукт, востребованный на рынке.

По результатам разработки были выполнены экономические расчеты, которые показали целесообразность разработки данного программного приложения и выявили, что срок окупаемости продукта составит 2 года.

Отмечается рост трудоемкости решения задачи, сокращение затрат на зарплату и социальные выплаты, прирост условной прибыли, рост производительности труда. В последних разделах работы были изучены и описаны принципы охраны труда на предприятии.

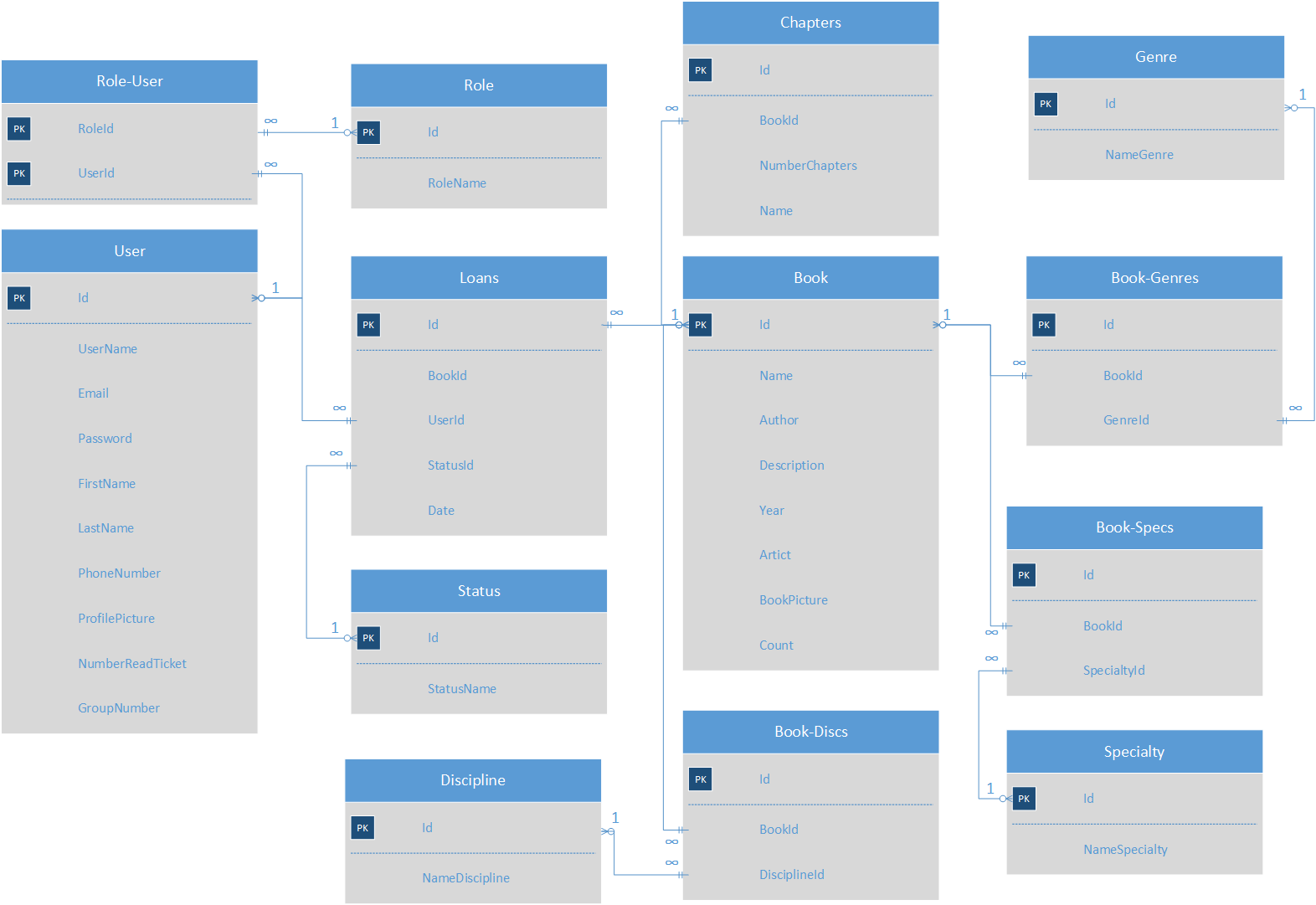
# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Библиотечно-информационное обслуживание: тенденции развития [Электронный ресурс] – Режим доступа: [https://myfilology.ru/194/bibliotechno-informaczionnoe-obsluzhivanie-tendenczii-razvitiya/#](https://myfilology.ru/194/bibliotechno-informaczionnoe-obsluzhivanie-tendenczii-razvitiya/) – Дата доступа: 24.04.2021.
2. Информационные системы, базы знаний, базы данных [Электронный ресурс] – Режим доступа: <https://www.sites.google.com/site/inftech11/home/sam/informacionnye-sistemy-bazy-znanij-bazy-dannyh> – Дата доступа: 24.04.2021.
3. Информационная система [Электронный ресурс] – Режим доступа: [https://ru.wikipedia.org/wiki/Информационная\_система](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0) – Дата доступа: 24.04.2021.
4. WEB-учебник "Информационные технологии в профессиональной деятельности" [Электронный ресурс] – Режим доступа: <http://xn80a44a.1gb.ru/UMK/p3aa1.html> – Дата доступа: 24.04.2021.
5. Что такое: Электронные библиотеки [Электронный ресурс] – Режим доступа: <https://www.ikirov.ru/news/10408-chto-takoe-elektronnye-biblioteki> – Дата доступа: 24.04.2021.
6. Электронные библиотека [Электронный ресурс] – Режим доступа: <https://ozlib.com/818337/informatika/elektronnye_biblioteki> – Дата доступа: 24.04.2021.
7. Электронная библиотека [Электронный ресурс] – Режим доступа: <https://wikichi.ru/wiki/Digital_library> – Дата доступа: 24.04.2021.
8. Rational Rose [Электронный ресурс] / Режим доступа: https://itteach.ru/rational-rose/. –– Дата доступа: 03.12.2020.
9. UML диаграммы в Rational Rose [Электронный ресурс] / Режим доступа: <http://www.caseclub.ru/articles/rose2.html> – Дата доступа: 26.04.2021.
10. Диаграмма вариантов использования [Электронный ресурс] / Режим доступа: <https://ru.wikipedia.org/wiki/диаграмма_прецедентов> – Дата доступа: 26.04.2021.
11. Диаграмма последовательности [Электронный ресурс] / Режим доступа: <https://ru.wikipedia.org/wiki/диаграмма_последовательности> – Дата доступа: 26.04.2021.
12. Диаграмма кооперации [Электронный ресурс] / Режим доступа: https://ru.wikipedia.org/wiki/диаграмма\_ кооперации – Дата доступа: 26.04.2021.
13. Диаграмма классов [Электронный ресурс] / Режим доступа: <https://ru.wikipedia.org/wiki/диаграмма_классов> – Дата доступа: 26.04.2021.
14. Диаграмма состояний [Электронный ресурс] / Режим доступа: https://ru.wikipedia.org/wiki/диаграмма\_ состояний – Дата доступа: 26.04.2021.
15. Диаграммы состояния [Электронный ресурс] / Режим доступа: <http://itteach.ru/diagrammi-sostoyaniy/vse-stranitsi> – Дата доступа: 26.04.2021.
16. Диаграмма развёртывания [Электронный ресурс] / Режим доступа: <https://ru.wikipedia.org/wiki/диаграмма_развёртывания> – Дата доступа: 26.04.2021.
17. Общие сведения о платформе .Net [Электронный ресурс] – Режим доступа: <https://docs.microsoft.com/ru-ru/dotnet/framework/get-started/overview> – Дата доступа: 26.04.2021.
18. Visual Studio [Электронный ресурс] – Режим доступа: https://visualstudio.microsoft.com/ru/vs/ – Дата доступа: 26.04.2021.
19. ADO.NET Entity Framework [Электронный ресурс] Режим доступа: https://docs.microsoft.com/ru-ru/dotnet – Дата доступа: 26.04.2021.
20. Введение в ASP.NET Core [Электронный ресурс] – Режим доступа: https://docs.microsoft.com/ru-ru/ASPNET/Core/?view=aspnetcore- 3.1&viewFallbackFrom=aspnetcore-2.1 – Дата доступа: 26.04.2021.
21. Фаулер, М. Шаблоны корпоративных приложений. – Россия: Вильямс, 2011. – 544 с.
22. Что такое ORM [Электронный ресурс] Режим доступа: https://ru.wikipedia.org/wiki/ORM – Дата доступа: 26.04.2021.
23. Т. Коннолли. Базы данных. Проектирование, реализация и сопровождение. Теория и практика. / Т. Коннолли, К. Бегг – М.: «Вильямс», 2003. – 179c.
24. СУБД [Электронный ресурс] – Режим доступа: [https://studopedia.org/4-](https://studopedia.org/4-158599.html) [158599.html](https://studopedia.org/4-158599.html) – Дата доступа: 03.05.20.
25. Белокопытов А., Володченков А. Автоматизированная обработка агроэкономической информации средствами MS Access. – Смоленск, 2006. – 23 с.
26. Т.С. Карпова. Базы данных: модели, разработка, реализация./ Т.С. Карпова - Сб. :Питер, 2001. - 304с.
27. Прамодкумар Дж. Рефакторинг баз данных: эволюционное проектирование. / Дж. Прамодкумар – М.: «Вильямс», 2007. – 130с.
28. Схема базы данных [Электронный ресурс] – Режим доступа: https://ru.wikipedia.org/wiki/Схема\_базы\_данных – Дата доступа: 26.04.2021.
29. Концепция MVC [Электронный ресурс] – Режим доступа: <https://ruseller.com/lessons.php?id=666>– Дата доступа: 26.04.2021.
30. Обязательное страхование [Электронный ресурс] – Режим доступа: http://www.ssf.gov.by/priside/statesocialinsurance/norm\_doc/uc/~page\_\_m17=1~news\_\_m17=1491 – Дата доступа: 26.04.2021.
31. СТБ 18001-2009 "Системы управления охраной труда. Требования" [Электронный ресурс] – Режим доступа: https://www.novation.by/articles/stb-18001-2009-sistemy-upravleniya-okhranoy-truda-trebovaniya/ – Дата доступа: 26.04.2021.

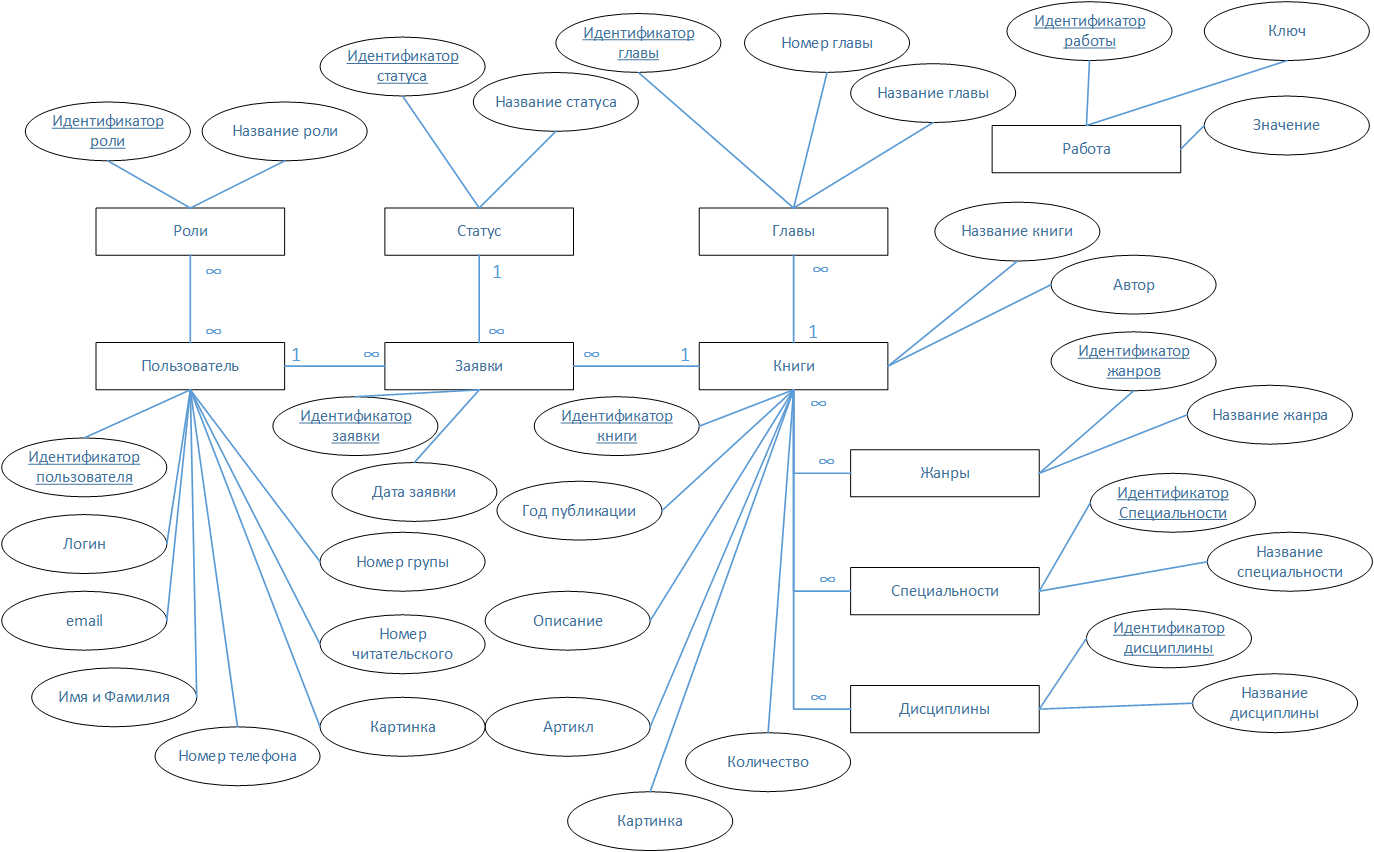
# ПРИЛОЖЕНИЕ А



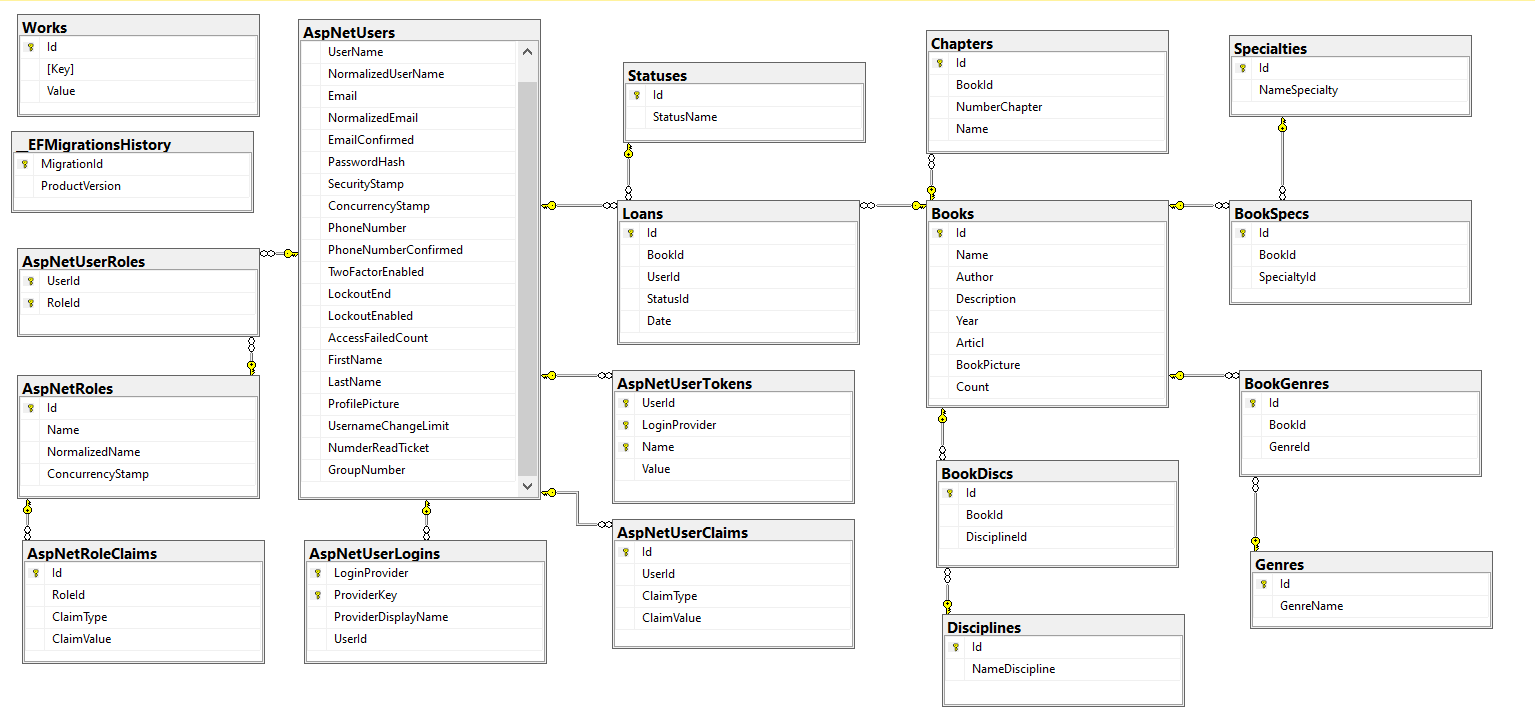
# ПРИЛОЖЕНИЕ Б



# ПРИЛОЖЕНИЕ В



# ПРИЛОЖЕНИЕ Г



# ПРИЛОЖЕНИЕ М

**Охрана труда инженера-программиста**

**1. Общие требования безопасности**

Настоящая инструкция по охране труда инженера-программиста, занятого эксплуатацией персональных электронно-вычислительных машин (ПЭВМ) разработана с учетом условий его работы.

На программиста могут воздействовать опасные и вредные производственные факторы:

***1. Физические:***

* Повышенные уровни электромагнитного излучения.
* Повышенные уровни рентгеновского излучения.
* Повышенные уровни ультрафиолетового излучения.
* Повышенный уровень инфракрасного излучения.
* Повышенный уровень статического электричества.
* Повышенные уровни запыленности воздуха рабочей зоны.
* Пониженная или повышенная влажность воздуха рабочей зоны.
* Пониженная или повышенная подвижность воздуха рабочей зоны.
* Повышенный уровень шума.
* Повышенный или пониженный уровень освещенности.
* Повышенный уровень прямой блесткости.
* Повышенный уровень отраженной блесткости.
* Повышенный уровень ослепленности:
* неравномерность распределения яркости в поле зрения.
* повышенная яркость светового изображения;
* повышенный уровень пульсации светового потока;
* повышенное значение напряжения в электрической цепи, замыкание которой может произойти через тело человека;

***2. Химические:***

Повышенное содержание в воздухе рабочей зоны двуокиси углерода, озона, аммиака, фенола, формальдегида и полихлорированных бифенилов.

***3. Биологические:***

Повышенное содержание в воздухе рабочей зоны микроорганизмов.

***4. Психофизиологические:***

* Напряжение зрения.
* Напряжение внимания.
* Интеллектуальные нагрузки.
* Эмоциональные нагрузки.
* Длительные статические нагрузки.
* Монотонность труда.
* Большой объем информации, обрабатываемой в единицу времени.
* Нерациональная организация рабочего места.

К работе программистом допускаются:

* Лица не моложе 18 лет, прошедшие обязательный при приеме на работу и ежегодные медицинские освидетельствования на предмет пригодности для работы.
* Прошедшие вводный инструктаж по охране труда.
* Прошедшие обучение безопасным приемам и методам труда по программе, утвержденной руководителем организации и прошедшие проверку знаний, в том числе по электробезопасности.
* Прошедшие курс обучения на персональном компьютере с использованием конкретного программного обеспечения.
* Прошедшие инструктаж по охране труда на конкретном рабочем месте по данной инструкции.

**2. Требования безопасности перед началом работы**

Перед началом работы программист обязан:

* Осмотреть и привести в порядок рабочее место.
* Отрегулировать освещенность на рабочем месте, убедиться в достаточности освещенности, отсутствии отражений на экране, отсутствии встречного светового потока.
* Проверить правильность подключения оборудования в электросеть.
* Протереть специальной салфеткой поверхность экрана.
* Убедиться в отсутствии дисков в дисководах процессора персонального компьютера.
* Проверить правильность установки стола, стула, подставки для ног, положения оборудования, угла наклона экрана, положение клавиатуры и, при необходимости, произвести регулировку рабочего стола и кресла, а также расположение элементов компьютера в соответствии с требованиями эргономики и в целях исключения неудобных поз и длительных напряжений тела.
* При включении компьютера соблюдать правила электробезопасности.

Программисту запрещается приступать к работе при:

* Обнаружение неисправности оборудования.
* Отсутствие защитного заземления устройств.
* Нахождении на рабочем месте в состоянии алкогольного или наркотического опьянения.
* Плохом самочувствии (головная боль, головокружение, тошнота, боль в глазах и т.п.).

**3. Требования безопасности во время работы**

Программист во время работы обязан:

* Выполнять только ту работу, которая ему была поручена, и по которой он был проинструктирован.
* В течение всего рабочего дня содержать в порядке и чистоте рабочее место.
* Держать открытыми все вентиляционные отверстия устройств.
* При необходимости прекращения работы на некоторое время корректно закрыть все активные задачи.
* Выполнять санитарные нормы и соблюдать режимы работы и отдыха.
* Соблюдать правила эксплуатации вычислительной техники в соответствии с инструкциями по эксплуатации.
* Соблюдать установленные режимом рабочего времени регламентированные перерывы в работе и выполнять в физкультпаузах и физкультминутках рекомендованные упражнения для глаз, шеи, рук, туловища, ног.
* Соблюдать расстояние от глаз до экрана в пределах 60 - 80 см.

Программисту во время работы запрещается:

* Прикасаться к задней панели системного блока (процессора) при включенном питании.
* Переключать разъемы интерфейсных кабелей периферийных устройств при включенном питании;
* Загромождать верхние панели устройств бумагами и посторонними предметами;
* Производить отключение питания во время выполнения активной задачи;
* Производить частые переключения питания;
* Допускать попадание влаги на поверхность системного блока (процессора), монитора, рабочую поверхность клавиатуры, дисководов, принтеров и др. устройств;
* Включать сильноохлажденное (принесенное с улицы в зимнее время) оборудование;
* Производить самостоятельно вскрытие и ремонт оборудования, без согласия руководителя.

**4. Требования безопасности в аварийных ситуациях**

Программист обязан:

* Во всех случаях обнаружения обрыва проводов питания, неисправности заземления и других повреждений электрооборудования, появления запаха гари немедленно отключить питание и сообщить об аварийной ситуации руководителю и дежурному электрику.
* При обнаружении человека, попавшего под напряжение, немедленно освободить его от действия тока путем отключения электропитания и до прибытия врача оказать потерпевшему первую медицинскую помощь.
* При любых случаях сбоя в работе технического оборудования или программного обеспечения немедленно поставить в известность руководителя.
* В случае появления рези в глазах, резком ухудшении видимости, - невозможности сфокусировать взгляд или навести его на резкость, появлении боли в пальцах и кистях рук, усилении сердцебиения немедленно покинуть рабочее место, сообщить о происшедшем руководителю работ и обратиться к врачу.
* При возгорании оборудования отключить питание и принять меры к тушению очага пожара при помощи углекислотного или порошкового огнетушителя, вызвать пожарную команду и сообщить о происшествии руководителю работ.

**5. Требования безопасности по окончании работы**

По окончании работ программист обязан соблюдать следующую последовательность выключения вычислительной техники:

* Произвести закрытие всех активных задач.
* Убедиться, что в дисководах нет дисков.
* Выключить питание системного блока (процессора).
* Выключить питание всех периферийных устройств.
* Отключить блок питания.

По окончании работ программист обязан осмотреть и привести в порядок рабочее место и вымыть с мылом руки и лицо.

# ПРИЛОЖЕНИЕ Н

**Охрана труда при работе на ПК**

1. **Общие требования по охране труда**

Настоящей инструкцией устанавливаются требования по охране труда при работе на персональных компьютерах (далее ПК) в том числе портативных (нетбуки, ноутбуки и др.).

К работе на ПК допускаются лица:

* прошедшие медицинский осмотр и не имеющие медицинских противопоказаний;
* прошедшие вводный инструктаж по охране труда;
* прошедшие обучение безопасным приемам и методам труда, и прошедшие проверку знаний по вопросам охраны труда;
* получившие первичный инструктаж по охране труда на конкретном рабочем месте по данной инструкции.

Женщинам со дня установления беременности и в период кормления ребенка грудью следует ограничить время работы на ПК до 3 часов за рабочую смену с учетом обеспечения оптимальных условий труда и регламентированных перерывов.

При невозможности ограничить время работы на ПК до 3 часов за рабочую смену с учетом обеспечения оптимальных условий труда и регламентированных перерывов по причинам, связанным с особенностями технологического процесса, женщины со времени установления беременности и в период кормления ребенка грудью должны быть переведены на работы, не связанные с использованием ПК.

Не допускается нахождение работающих в состоянии алкогольного опьянения либо в состоянии, вызванном употреблением наркотических средств, психотропных или токсичных веществ, а также распитие спиртных напитков, употребление наркотических средств, психотропных или токсических веществ на рабочем месте или в рабочее время, курение в неустановленном месте.

Опасными и вредными производственными факторами, действующими на работающих при работе на ПК, являются:

* повышенный уровень электромагнитных излучений;
* повышенный уровень ионизирующих излучений;
* повышенный уровень статического электричества;
* повышенная напряженность электростатического поля;
* повышенная или пониженная ионизация воздуха;
* повышенная яркость света;
* прямая и отраженная блесткость;
* повышенное значение напряжения в электрической цепи, замыкание которой может произойти через тело человека;
* статические перегрузки костно-мышечного аппарата и динамические локальные перегрузки мышц кистей рук;
* перенапряжение зрительного анализатора;
* умственное перенапряжение;
* эмоциональные перегрузки;
* монотонность труда.

В зависимости от условий труда, в которых применяются ПК, и характера работы на работающих могут воздействовать также другие вредные и (или) опасные производственные факторы.

Работающие с учетом воздействующих на них вредных и (или) опасных производственных факторов обеспечиваются в соответствии с законодательством средствами индивидуальной защиты.

Работающие обязаны:

* знать и соблюдать требования эксплуатационных документов организаций-изготовителей, используемых ПК;
* проходить и выходить с территории предприятия только через контрольно-пропускные пункты;
* соблюдать режим труда и отдыха, установленный законодательством, правилами внутреннего трудового распорядка организации, трудовую дисциплину, выполнять требования по охране труда, правила личной гигиены;
* выполнять требования пожарной безопасности;
* курить только в специально предназначенных для курения местах;
* заботиться о личной безопасности и личном здоровье, а также о безопасности окружающих в процессе выполнения работ либо во время нахождения на территории организации;
* содержать рабочее место в порядке и чистоте;
* знать местонахождение аптечки первой медицинской помощи универсальной;
* сообщать непосредственному руководителю, иному уполномоченному должностному лицу нанимателя или лицам, осуществляющим техническое обслуживание оборудования о неисправности ПК и периферийных устройств (принтера, сканера, клавиатуры ПК, электрических компьютерных сетевых устройств, блока бесперебойного питания и других устройств) (далее - оборудование) и иных неполадках, препятствующих выполнению работы, и не приступать к работе до их устранения;
* немедленно сообщать непосредственному руководителю или иному уполномоченному должностному лицу нанимателя о любой ситуации, угрожающей жизни или здоровью работающих и окружающих;
* исполнять другие обязанности, предусмотренные законодательством.

Работающие имеют право отказаться от выполнения порученной работы в случае возникновения непосредственной опасности для жизни и здоровья их и окружающих до устранения этой опасности.

При несчастном случае на производстве работающий обязан:

* принять меры по предотвращению воздействия травмирующих факторов на потерпевшего, оказанию потерпевшему первой помощи, вызову на место происшествия медицинских работников или доставке потерпевшего в организацию здравоохранения;
* немедленно сообщить о несчастном случае непосредственному руководителю или иному уполномоченному должностному лицу;
* обеспечить до начала расследования сохранность обстановки на месте происшествия, если не существует угрозы жизни и здоровью окружающих.

Во время работы необходимо быть внимательным, не отвлекаться самому и не отвлекать других.

Перед приемом пищи мыть руки, прием пищи производить только в столовых, буфетах и специально отведенных и оборудованных для этого местах.

Оконные проемы должны быть оборудованы регулируемыми устройствами типа жалюзи, занавесей, внешних козырьков и другое.

Высота рабочей поверхности стола должна регулироваться в пределах 680 - 800 мм. При отсутствии такой возможности высота рабочей поверхности стола должна составлять 725 мм.

Рабочий стол должен иметь пространство для ног высотой не менее 600 мм, шириной — не менее 500 мм, глубиной на уровне колен — не менее 450 мм и на уровне вытянутых ног — не менее 650 мм.

Конструкция рабочего стола должна обеспечивать оптимальное размещение на рабочей поверхности используемого оборудования с учетом его количества и конструктивных особенностей, характера выполняемой работы. При этом допускается использование рабочих столов различных конструкций, отвечающих современным требованиям эргономики.

Рабочие столы следует размещать таким образом, чтобы экраны ПК были ориентированы боковой стороной к световым проемам (исключение составляет периметральная расстановка рабочих мест), чтобы естественный свет падал преимущественно слева.

Рабочий стул (кресло) должен быть подъемно-поворотным, регулируемым по высоте и углам наклона сиденья и спинки, а также расстоянию спинки от переднего края сиденья, при этом регулировка каждого параметра должна быть независимой, легко осуществляемой и иметь надежную фиксацию.

Конструкция рабочего стула (кресла) должна обеспечивать:

* ширину и глубину поверхности сиденья не менее 400 мм;
* регулировку высоты поверхности сиденья в пределах 400 - 550 мм и угла наклона вперед до 15 градусов и назад до 5 градусов;
* высоту опорной поверхности спинки 300 +/- 20 мм, ширину не менее 380 мм и радиус кривизны горизонтальной плоскости — 400 мм;
* угол наклона спинки в вертикальной плоскости в пределах 0 +/- 30 градусов;
* регулировку расстояния спинки от переднего края сиденья в пределах 260 - 400 мм;
* стационарные или съемные подлокотники длиной не менее 250 мм и шириной — 50 - 70 мм;
* регулировку подлокотников по высоте над сиденьем в пределах 230 +/- 30 мм и внутреннего расстояния между подлокотниками в пределах 350 - 500 мм.

Поверхность сиденья, спинки и других элементов рабочего стула (кресла) должна быть полумягкой, с нескользящим, слабо электризующимся и воздухопроницаемым покрытием, обеспечивающем легкую очистку от загрязнений. Поверхность сиденья должна иметь закругленный передний край.

Рабочее место при работе на ПК следует оборудовать подставкой для ног, имеющей ширину не менее 300 мм, глубину не менее 400 мм, регулировку по высоте в пределах до 150 мм и по углу наклона опорной поверхности подставки до 20 градусов. Поверхность подставки должна быть рифленой и иметь по переднему краю бортик высотой 10 мм.

Не допускается:

* устанавливать системный блок в закрытых объемах мебели, непосредственно на полу;
* использовать для подключения оборудования розетки, удлинители, не оснащенные заземляющим контактом (шиной);
* касаться находящихся в движении частей механизмов;
* прикасаться к токоведущим частям, электрическим проводам (изолированным и не изолированным), кабелям, шинам, клеммам, патронам освещения и пр.

За невыполнение требований настоящей инструкции работающие несут ответственность в соответствии с законодательными актами.

**2. Требования по охране труда перед началом работы**

Осмотреть рабочее место и убедиться:

* в устойчивости положения оборудования на рабочем столе;
* в отсутствии видимых повреждений оборудования;
* в исправности и целостности питающих и соединительных кабелей, разъемов и штепсельных соединений;
* в исправности общего и местного освещения, мебели.

При невозможности устранить нарушения, доложить об этом непосредственному руководителю, до устранения нарушений к работе не приступать.

Расположить клавиатуру на поверхности стола на расстоянии 100 - 300 мм от края, обращенного к взрослому пользователю, или на специальной, регулируемой по высоте рабочей поверхности, отделенной от основной столешницы.

Разместить экран видеомонитора на расстоянии 600 - 700 мм от глаз, но не ближе 500 мм с учетом размеров алфавитно-цифровых знаков и символов (далее — оптимальное расстояние от экрана видеомонитора до глаз) так, чтобы уровень глаз при вертикально расположенном экране видеомонитора приходился на центр или 2/3 высоты экрана. Линия взора должна быть перпендикулярна центру экрана, и оптимальное ее отклонение от перпендикуляра, проходящего через центр экрана в вертикальной плоскости, не должно превышать +/-5 градусов, допустимое - +/-10 градусов.

Убедиться в отсутствии бликов (отражений) на экране видеомонитора, встречного светового потока. Возможные мешающие отражения и отблески на экране видеомонитора и другом оборудовании устраняются путем соответствующего их размещения, расположения светильников местного освещения. Для снижения яркости в поле зрения при естественном освещении необходимо применить регулируемые жалюзи, плотные шторы

При необходимости включить местное освещение, протереть поверхность экрана видеомонитора сухой мягкой тканевой салфеткой, проветрить помещение.

Отрегулировать подъемно-поворотный стул (кресло) по высоте и углам наклона сиденья и спинки, а также расстоянию спинки от переднего края сиденья, с учетом роста. Регулировка каждого параметра подъемно-поворотного стула (кресла) должна быть независимой, легко осуществляемой и иметь надежную фиксацию.

Отрегулировать положение подставки для ног (в случае ее использования).

Включить оборудование в электрическую сеть, соблюдая следующую последовательность: стабилизатор напряжения (если он используется), блок бесперебойного питания, периферийные устройства (принтер, видеомонитор, сканер и другие устройства), системный блок

Работающему не допускается:

* устанавливать системный блок в закрытых объемах мебели, непосредственно на полу;
* использовать для подключения оборудования розетки, удлинители, не оснащенные заземляющим контактом (шиной);
* включать охлажденное (принесенное с улицы в зимнее время) оборудование;
* располагать экраны видеомониторов навстречу друг другу при рядном размещении рабочих столов в целях исключения их взаимного отражения;
* приступать к работе на ПК при мелькании изображения на экране видеомонитора, в случае обнаружения неисправности оборудования, кабелей или проводов, разъемов, штепсельных соединений, при отсутствии или неисправности защитного заземления (зануления) оборудования.

**3. Требования по охране труда при выполнении работы**

При работе на ПК работающий обязан:

* выполнять только ту работу, которая ему поручена;
* содержать в порядке и чистоте свое рабочее место;
* держать открытыми вентиляционные отверстия оборудования;
* соблюдать оптимальное расстояние от экрана видеомонитора до глаз;
* поддерживать рациональную рабочую позу и оптимальное размещение на рабочей поверхности используемого оборудования с учетом его количества и конструктивных особенностей, характера выполняемой работы;
* осуществлять систематическое проветривание помещения после каждого часа работы на ПК.

Работу на ПК следует периодически прерывать на регламентированные перерывы, которые устанавливаются для обеспечения работоспособности и сохранения здоровья, или заменять другой работой.

Продолжительность непрерывной работы на ПК без регламентированного перерыва не должна превышать 2 часов.

При восьмичасовом рабочем дне и работе на ПК регламентированные перерывы следует устанавливать:

* при выполнении работ по считыванию информации с экрана ПК с предварительным запросом до 20000 знаков (работа по вводу информации до 15000 знаков либо творческая работа в режиме диалога с ПК до 2 часов) - через 2 часа от начала рабочего дня (смены) и через 2 часа после обеденного перерыва продолжительностью 15 минут каждый;
* при выполнении работ по считыванию информации с экрана ПК с предварительным запросом до 40000 знаков (работа по вводу информации до 30000 знаков либо творческая работа в режиме диалога с ПК до 4 часов) - через 2 часа от начала рабочего дня (смены) и через 1,5 - 2 часа после обеденного перерыва продолжительностью 15 минут каждый или продолжительностью 10 минут через каждый час работы;
* при выполнении работ по считыванию информации с экрана ПК с предварительным запросом до 60000 знаков (работа по вводу информации до 40000 знаков либо творческая работа в режиме диалога с ПК до 6 часов) - через 1,5 - 2 часа от начала рабочего дня (смены) и через 1,5 - 2 часа после обеденного перерыва продолжительностью 20 минут каждый или продолжительностью 15 минут через каждый час работы.

Во время регламентированных перерывов с целью снижения нервно-эмоционального напряжения, утомления зрительного анализатора, устранения влияния гиподинамии и гипокинезии, предотвращения развития статического утомления необходимо выполнять физические упражнения и упражнения для глаз.

С целью уменьшения отрицательного влияния монотонности труда целесообразно применять чередование операций.

При работе на ПК не допускается:

* прикасаться к панелям с разъемами оборудования, разъемам питающих и соединительных кабелей, экрану видеомонитора при включенном питании;
* натягивать, перекручивать и перегибать кабель, ставить на него предметы;
* загромождать рабочее место;
* ограничивать доступ работающих к первичным средствам пожаротушения, аптечкам первой медицинской помощи универсальным;
* производить переключения, отключение питания во время выполнения активной задачи;
* допускать попадание влаги на поверхность оборудования;
* производить самостоятельно вскрытие и ремонт оборудования;
* вытирать пыль на включенном оборудовании;
* допускать нахождение вблизи оборудования посторонних лиц;
* оставлять оборудование включенным без наблюдения.

**4. Требования по охране труда по окончании работы**

По окончании работы на ПК работающий обязан:

* корректно закрыть все активные задачи;
* извлечь съемные носители;
* выключить питание системного блока;
* выключить питание всех периферийных устройств;
* отключить блок бесперебойного питания;
* отключить стабилизатор напряжения (если он используется);
* отключить питающий кабель от сети;
* осмотреть и привести в порядок рабочее место;
* при необходимости протереть поверхности периферийных устройств и вымыть с мылом руки.

Протирание периферийных устройств производить мягкой ветошью с применением специальных или бытовых чистящих средств, не содержащих кислот и отбеливателей, при выключенном оборудовании методом и средствами, не влияющими на работоспособность данных устройств, не реже 1 раза в неделю.

О неисправности оборудования и других замечаниях по работе сообщить непосредственному руководителю или лицам, осуществляющим техническое обслуживание оборудования.

**5. Требования по охране труда в аварийных ситуациях**

Во всех случаях обнаружения обрыва проводов питания, возникновении необычного шума и других повреждений электрооборудования, появления запаха гари немедленно отключить питание и сообщить руководителю и лицу, осуществляющему техническое обслуживание оборудования.

При любых случаях сбоя в работе оборудования или программного обеспечения вызвать специалиста, осуществляющего техническое обслуживание данного оборудования для устранения неполадок.

В случае возникновения возгорания или пожара работающий обязан отключить от электросети оборудование, принять меры по эвакуации работающих в безопасное место, вызвать подразделение по чрезвычайным ситуациям по телефону 101, указав адрес объекта и участок возгорания, сообщить о происшедшем непосредственному руководителю или иному уполномоченному должностному лицу нанимателя, приступить к тушению пожара имеющимися средствами пожаротушения. Действовать согласно Плану ликвидации аварий.

Применение воды и пенных огнетушителей для тушения находящегося под напряжением электрооборудования не допускается. Для этих целей используются углекислотные и порошковые огнетушители.

В случае внезапного ухудшения здоровья (усиления сердцебиения, появления головной боли, рези в глазах, резком ухудшении видимости, невозможности сфокусировать взгляд или навести его на резкость, появлении боли в пальцах и кистях рук, усилении сердцебиения и других) прекратить работу, выключить оборудование, сообщить об этом руководителю и при необходимости обратиться к врачу.

# ВЕДОМОСТЬ ОБЪЁМА ДИПЛОМНОГО ПРОЕКТА

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Формат | Обозначение | | Наименование | | | Количество листов | | | Примечание | |
| А4 | - | | Задание по выполнению  дипломного проекта | | | 1 | | |  | |
|  |  | |  | | |  | | |  | |
| А4 | ПолесГУ ДП 1-40 05  01-02 02 012 ПЗ | | Пояснительная записка | | | 119 | | |  | |
| А4 | - | | Отзыв руководителя | | | 1 | | |  | |
| А4 | - | | Рецензия | | | 1 | | |  | |
|  |  | |  | | |  | | |  | |
|  |  | |  | | |  | | |  | |
|  |  |  |  |  | ПолесГУ ДП 1-40 05 01-02 02 012 | | | | | |
|  |  |  |  |  |
|  |  | Ф.И.О. | Подпись | Дата |
| Дипломник | | М. Г. Семашко |  |  | Ведомость объёма дипломного проекта | | Стадия | Лист | | Листов |
| Руководитель ДП | | П. Б. Пигаль |  |  | ДП | 119 | | 119 |
| Консультант | |  |  |  | Кафедра ИТ и ИС, гр. 17ИТ-2 | | | |