**РОСЖЕЛДОР**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ**

**ВЫСШЕГО ОБРАЗОВАНИЯ**

**СИБИРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**ПУТЕЙ СООБЩЕНИЯ (СГУПС)**

|  |  |  |  |
| --- | --- | --- | --- |
| **К защите:** |  |  | |
| **Заведующий кафедрой** | **Информационные** | |
| **технологии транспорта** | | |
|  | д-р техн. наук, проф. | |
|  |  | В. И. Хабаров | |
| *подпись* |  | *инициалы, фамилия* | |
|  |  |  | |
| *дата* |  |  | |

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Тема:** | Разработка программного обеспечения учёта почасовой | | | | | |
|  | нагрузки преподавателя в университете | | | | | |
|  | |  | БР.БИСТ.07.2022 |  |  |
|  | |  | *шифр документа* |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | **Выполнил** |  |  |  | **Руководитель** |
|  |  | В. И. Гонец |  |  |  | канд. техн. наук, доц.  А. А. Уланов |
| *подпись* |  | *инициалы, фамилия* |  | *подпись* |  | *инициалы, фамилия* |
|  |  |  |  |  |  |  |
| *дата* |  |  |  | *дата* |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
| Нормоконтролер работы |  |  |  | ст. преп.  Т. А. Распопина |
|  |  | *подпись* |  |  |
|  |  |  |  |  |
|  |  | *дата* |  |  |

**2022 г.**

**СИБИРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**ПУТЕЙ СООБЩЕНИЯ (СГУПС)**

Факультет: Бизнес-информатики

Кафедра: Информационные технологии транспорта

Направление: 09.03.02 «Информационные системы и технологии»

Профиль: Интеллектуальные транспортные системы

|  |  |
| --- | --- |
|  | ***УТВЕРЖДАЮ****: зав. кафедрой «Информационные технологии транспорта»*  д-р техн. наук, проф.  В. И. Хабаров |
|  | *«\_\_\_\_»\_\_\_\_\_\_\_\_\_\_ 2022 г.* |

**З А Д А Н И Е**

**на выполнение выпускной квалификационной работы**

|  |  |  |  |
| --- | --- | --- | --- |
| студенту | Гонец Виктору Ивановичу | | |
|  |  | | |
| 1. Тема «Разработка программного обеспечения учёта почасовой нагрузки преподавателя в университете» утверждена приказом №203/с от «30» мая 2022 г. | | | |
| 2. Задание выдано «12» мая 2022 г. | | | |
| 3. Срок сдачи законченной работы на кафедру «17» июня 2022 г. | | | |
| 4. Исходные данные: данные, полученные в ходе прохождения преддипломной практики | | | |
| 5. Содержание расчетно-пояснительной записки | | | |
| Наименование разделов и вопросов | | Примерное количество страниц | График (сроки) выполнения |
| Введение | | 2 | 20.05.2022 |
| Аналитическое исследование | | 5 | 25.05.2022 |
| Проектирование информационной системы | | 18 | 30.05.2022 |
| Создание приложения | | 9 | 04.06.2022 |
| Заключение | | 1 | 06.06.2022 |

6. Содержание и объемы графической части

|  |  |  |
| --- | --- | --- |
| Наименование графического документа (чертежа, схемы, графика) | Количество  листов  формата А1 | График  (сроки)  выполнения |
| Презентация PowerPoint | 15 | 08.06.2022 |

|  |  |  |
| --- | --- | --- |
| Руководитель |  | А. А. Уланов |
|  | *(подпись, фамилия, И.О.)* |  |
| Задание к использованию принял |  | В. И. Гонец |
|  | *(подпись студента)* |  |

УДК 004.41

**АННОТАЦИЯ**

В работе 42 страницы, 27 рисунков, 7 таблиц, 15 источников, 1 приложение.

Ключевые слова: *программное обеспечение, почасовая нагрузка преподавателя, учет почасовой нагрузки.*

Объектом исследования является бизнес-процесс подачи почасовой нагрузки преподавателя в университете.

Цель работы – разработать программное обеспечение учета почасовой нагрузки преподавателя в университете.

Проведен анализ предметной области и аналогов программного обеспечения, смоделированы бизнес-процессы предметной области, выбраны инструментальные средства для разработки системы, создана модель данных системы, разработано программное обеспечение учета почасовой нагрузки преподавателя в университете.

**ABSTRACT**

The work contains 42 pages, 27 figures, 7 tables, 15 sources, 1 appendix.

Keywords: software, hourly teacher workload, hourly workload accounting.

The object of the study is the business process of submitting the hourly workload of a teacher at the university.

The purpose of the work is to develop software for accounting for the hourly workload of a teacher at the university.

The analysis of the subject area and software analogues was carried out, the business processes of the subject area were modeled, tools for system development were selected, a data model of the system was created, software for accounting for the hourly workload of a teacher at the university was developed.

**ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ**

ПО – программное обеспечение.

UML (Unified Modeling Language) – язык графического описания для моделирования бизнес-процессов.

СУБД – система управления базами данных.

ОС – операционная система.

IDE (Integrated development environment) – комплекс программных средств, используемый программистами для разработки программного обеспечения.

SQL (Structured Query Language) – декларативный язык программирования, применяемый для управления данными в реляционной базе данных.

C# – объектно-ориентированный язык программирования высокого уровня.

.NET – программная платформа компании Microsoft.

ER (Entity Relationship) – модель «сущность-связь».

Visual Studio – среда разработки от компании Microsoft.

**СОДЕРЖАНИЕ**

[Введение 6](#_Toc105353351)

[1 Аналитическое исследование 8](#_Toc105353352)

[1.1 Наименование и область применения 8](#_Toc105353353)

[1.2 Анализ предметной области 8](#_Toc105353354)

[1.3 Анализ существующих аналогов программного обеспечения учета индивидуальной нагрузки преподавателя 10](#_Toc105353355)

[1.4 Выводы по первому разделу 12](#_Toc105353356)

[2 Проектирование информационной системы 13](#_Toc105353357)

[2.1 Моделирование бизнес-процессов 13](#_Toc105353358)

[2.2 Язык программирования 19](#_Toc105353359)

[2.3 Выбор среды разработки 20](#_Toc105353360)

[2.4 Система управления базами данных (СУБД) 23](#_Toc105353361)

[2.5 Модель данных (ER-модель) 23](#_Toc105353362)

[2.6 Проектирование интерфейса информационной системы 27](#_Toc105353363)

[2.7 Выводы по второму разделу 30](#_Toc105353364)

[3 Создание приложения 31](#_Toc105353365)

[3.1 Описание функциональных особенностей информационной системы 31](#_Toc105353366)

[3.2 Условия обеспечения работоспособности информационной системы 32](#_Toc105353367)

[3.3 Выполнение операций в программе 33](#_Toc105353368)

[3.4 Выводы по третьему разделу 39](#_Toc105353369)

[Заключение 40](#_Toc105353370)

[Список использованных источников 41](#_Toc105353371)

[Приложение А Шаблоны документов 43](#_Toc105353372)

ВВЕДЕНИЕ

С увеличением объема информации, задействованной в деятельности университета, возникает потребность в автоматизации некоторых бизнес-процессов, для освобождения сотрудников от повторяющихся действий. Одним из шагов, необходимых для этого – переход к электронному документообороту. Кроме того, в условиях усложнения бизнес-процессов современных вузов для повышения их конкурентоспособности требуется оптимизировать деятельность структурных подразделений и при необходимости организационных структур, а также усилить их взаимодействие с целью оптимизации времени принятия управленческих решений.

Актуальность выбранной темы обусловлена отсутствием на рынке программных средств, ограничивающихся необходимым диапазоном функциональных средств, и весьма большими трудовыми и временными затратами, необходимыми для составления документов.

Объект исследования – бизнес-процесс подачи почасовой нагрузки преподавателя в университете.

Предмет исследования – программное обеспечение для документоведа, необходимое для составления документов, содержащих отчетность по учету нагрузки преподавателя.

Цель данной выпускной квалификационной работы – разработать программное обеспечение учёта почасовой нагрузки преподавателя в университете. Программное обеспечение учёта почасовой нагрузки преподавателя в университете позволит облегчить и автоматизировать работу документоведа, занимающегося подготовкой различной документации.

Для достижения поставленной цели необходимо выполнить следующие задачи:

* проанализировать предметную область;
* ознакомиться с аналогами программного обеспечения;
* смоделировать бизнес процессы предметной области;
* выбрать и обосновать инструменты для разработки приложения;
* реализовать программное обеспечение учёта почасовой нагрузки преподавателя в университете.

В соответствии с намеченной целью и задачами работы определены следующие методы: анализ предметной области и существующих аналогов программного обеспечения учета индивидуальной нагрузки преподавателя, моделирование бизнес-процессов с помощью языка UML, сравнение различных сред разработки, метод графического представления модели данных в виде ER-диаграммы.

Практическая значимость работы состоит в возможности внедрения разработанного продукта в деятельность университета.

В качестве источников информации использовались сайт университета, локальные нормативные документы, учебная и периодическая литература, техническая документация и другая информация.

Выполненная работа состоит из введения, основной части из 3 разделов (аналитическая, проектная, прикладная), заключения, списка использованных источников и приложений.

1 Аналитическое исследование

В первом разделе проанализирована предметная область, в которой разрабатывается информационная система; проведен анализ существующих аналогов программного обеспечения учета индивидуальной нагрузки преподавателя; а также сформулированы цели проектирования и разработки информационной системы.

1.1 Наименование и область применения

Наименование предметной области – «Нагрузка преподавателя».

Область применения разрабатываемого приложения – кафедра «Информационные технологии транспорта» СГУПС.

Кафедра «Информационные технологии транспорта» (ИТТ) является ответственной за выпуск направлений «Информационные системы и технологии», а также «Прикладная информатика»[1].

Объектами профессиональной деятельности специалистов кафедры являются: информационные системы и сети, их математическое, информационное и программное обеспечение, способы и методы проектирования и эксплуатации программных средств информационных систем в различных областях народного хозяйства[2].

Разрабатываемое программное обеспечение «Учет почасовой нагрузки преподавателя в университете» позволит облегчить и автоматизировать процесс заполнения отчетов, связанных с учетом почасовой нагрузки преподавателя.

1.2 Анализ предметной области

Преподаватель – работник Университета из числа профессорско-преподавательского состава, то есть из работников Университета, имеющих право на выполнение обязанностей по обучению, воспитанию обучающихся и организации образовательной деятельности[3].

Педагогическая нагрузка преподавателя – это нагрузка первой и второй половины дня, включающая все направления деятельности, выполняемые преподавателем в соответствии с условиями трудового договора и индивидуальным планом работы: учебная, учебно-методическая, учебно-организационная, научная, общественная, издательская, инновационная и воспитательная работа[3].

Индивидуальная учебная нагрузка планируется в индивидуальном плане учебно-методической работы преподавателя, рассчитывается заведующим кафедрой/руководителем департамента, согласовывается с главным экспертом Управления организации учебного процесса, для штатного преподавателя университета может составлять до 900 часов в учебный год, является обязательным документом при оформлении на работу и прохождении конкурса, количество часов планируемой нагрузки должно соответствовать занимаемой должности претендента[4].

Объем часов контактной работы рассчитывается в академических часах на основе утвержденного учебного плана данного направления подготовки (специальности) и включает в себя:

* часы из учебного плана, отводимые на лекции, практические (семинарские) занятия и лабораторные работы;
* часы, определяемые нормами времени для расчета объема учебной нагрузки профессорско-преподавательского состава и отводимые на: проведение консультаций; прием зачетов; прием экзаменов; проверку, консультации и прием расчетно-графических работ, домашних заданий, рефератов и других работ; рецензирование и консультирование рефератов по программам подготовки магистров; курсовых работ; руководство, консультации, рецензирование и прием зашиты руководство, консультации, рецензирование и прием защиты курсовых проектов; руководство подготовкой студента в магистратуре; руководство программой специализированной подготовки в магистратуре; руководство научно-исследовательской работой студентов; руководство учебной практикой; руководство производственной, педагогической практиками; руководство производственной практикой, проводимой по индивидуальным планам студентов; другие виды работ, определяемые нормами времени для расчета объема учебной нагрузки профессорско-преподавательского состава, утверждаемыми ежегодно ректором университета.

Контактная работа при проведении текущего контроля успеваемости, обучающегося проводится в рамках аудиторных занятий или во внеаудиторное время, пределах норм времени, выделяемых на индивидуальные консультации в семестре[5].

Для учета почасовой нагрузки преподавателя составляются следующие документы:

* лист учета работы преподавателя-почасовика (составляется для штатных, внутренних и сторонних преподавателей);
* акт сдачи-приемки услуг преподавателя (составляется для преподавателей, нанятых по гражданско-правовому договору);
* сводка о выполнении учебной нагрузки (содержит данные листа учета работы и акта сдачи-приемки услуг).

1.3 Анализ существующих аналогов программного обеспечения учета индивидуальной нагрузки преподавателя

Существует несколько приложений, используемых для автоматизации управленческой деятельности вуза, в том числе имеющих функционал для учета индивидуальной нагрузки преподавателя, например, «БИТ:Учёт нагрузки преподавателей», «1C:Университет», автоматизированная система управления вузом «Галактика». Далее приведена краткая характеристику каждого из этих программных продуктов.

Программа «БИТ:Учёт нагрузки преподавателей» предназначена для автоматизации планирования и учёта выполнения нагрузки профессорско-преподавательского состава. Реализует процессы планирования и учёта нагрузки по преподавателям и кафедрам, формирования учебных планов по утверждённым государством стандартам, а также всей необходимой отчетности[6].

«1C:Университет» – решение автоматизации управленческой деятельности вуза. Продукт охватывает все уровни деятельности основных подразделений учреждения высшего профессионального образования и интегрируется с типовыми решениями фирмы «1С» для бухгалтерии и отдела кадров[7].

Автоматизированная система управления вузом «Галактика» – это современное комплексное ИТ-решение является базой для функционирования системы «Галактика ERP 9.1. Управление учебным процессом» и предназначено для эффективного управления всей финансово-хозяйственной деятельностью вуза. «Галактика Управление вузом» позволяет решать широкий спектр управленческих задач современного образовательного учреждения[8].

Обзор имеющихся на рынке программных продуктов показал, что основными направлениями сравниваемых разработок являются планирование учебного процесса в целом, проведение приемной кампании, организация внутреннего электронного документооборота, экономическое сопровождение высшего учебного заведения[9].

Анализ программных продуктов, предназначенных для автоматизации организации учебного процесса в высших учебных заведениях, позволяет сделать вывод о том, что помимо достоинств, заключающихся в широком диапазоне функциональных возможностей, они также не лишены недостатков, к которым можно отнести:

* высокую стоимость программного обеспечения;
* необходимость обучение персонала;
* необходимость обращения к разработчикам в случае возникновения неполадок системы;
* платное послегарантийное техническое обслуживание;
* отсутствие гарантий постоянной поддержки программных продуктов (в случае прекращение или приостановление разработки);
* часть функций программных комплексов не востребована ВУЗом.

Можно сделать вывод, что проанализированные программные средства не подходят из-за широкого диапазона функций, поэтому и разрабатывается программное обеспечение, которое будет включать узкий диапазон функций для учета индивидуальной нагрузки преподавателя и генерации необходимых документов, применимых конкретно на кафедре ИТТ.

1.4 Выводы по первому разделу

В первом разделе была проанализирована предметная область «Нагрузка преподавателя», в которой разрабатывается программное обеспечение «Учёт почасовой нагрузки преподавателя в университете»; а также проведен анализ существующих аналогов программного обеспечения учета индивидуальной нагрузки преподавателя.

Основными целями проектирования и разработки системы являются:

* повышение эффективности работы документоведа;
* повышение качества работы (снижение количества ошибок при подготовке отчетов, связанных с учетом нагрузки преподавателя; ускорение процесса составления отчетов).

2 Проектирование информационной системы

Во втором разделе выбрана и описана методология моделирования бизнес-процессов; смоделированы бизнес-процессы предметной области; выбраны современные инструментальные средства для проектирования и разработки информационной системы, а именно: язык программирования, среда разработки и система управления базами данных; создана модель данных информационной системы; а также спроектирован интерфейс информационной системы.

2.1 Моделирование бизнес-процессов

*Выбор методологии моделирования бизнес-процессов.*

Программное обеспечение «Учёт почасовой нагрузки преподавателя в университете» будет помогать документоведу подготавливать различные отчеты. Основные бизнес-процессы, автоматизированные в приложении:

* загрузка данных в приложение;
* составление документа с помощью приложения (одного из следующих: акта сдачи-приемки услуг преподавателя, листа учета работы преподавателя-почасовика и сводки о выполнении учебной нагрузки).

Для моделирования выделенных бизнес-процессов оптимальной методологией является UML, позволяющей описать их последовательность выполнения и все возможные состояния с объектной стороны. Для описания оптимизированного бизнес-процесса необходимо построение таких диаграмм UML, как: диаграмма вариантов использования, диаграмма последовательности, диаграмма коммуникации, диаграмма активностей, диаграмма состояний, диаграмма обзора взаимодействия и диаграмма классов[10].

*Методология объектного проектирования и анализа на языке UML.*

UML – язык графического описания для объектного моделирования в области разработки программного обеспечения, для моделирования бизнес-процессов, системного проектирования и отображения организационных структур. UML позволяет также разработчикам программного обеспечения достигнуть соглашения в графических обозначениях для представления общих понятий (таких как класс, компонент, обобщение, агрегация и поведение) и больше сконцентрироваться на проектировании и архитектуре.

Диаграммы UML подразделяют на два типа – это структурные диаграммы и диаграммы поведения.

Структурные диаграммы показывают статическую структуру системы и ее частей на разных уровнях абстракции и реализации, а также их взаимосвязь. Элементы в структурной диаграмме представляют значимые понятия системы и могут включать в себя абстрактные, реальные концепции и концепции реализации. Примеры структурных диаграмм: диаграмма классов, диаграмма компонентов, диаграмма композитной структуры, диаграмма объектов.

Диаграммы поведения показывают динамическое поведение объектов в системе, которое можно описать, как серию изменений в системе с течением времени. Примеры поведенческих диаграмм: диаграмма деятельности (активностей), диаграмма конечного автомата (состояний), диаграмма вариантов использования, диаграмма последовательности, диаграмма коммуникации, диаграмма обзора взаимодействия[11].

*Моделирование бизнес-процесса «Загрузка данных в приложение».*

В процессе загрузки данных в приложение принимают участие: документовед и десктопное приложение.

Документовед загружает необходимые данные в приложение (таблицы, формы документов и другие). Приложение проверяет добавленные данные и выводит результат об успешной загрузке или некорректности данных. В случае, если данные корректны, они сохраняются в приложении, иначе – документоводу необходимо провести их загрузку повторно.

На рисунках 2.1-2.7 показаны следующие виды диаграмм для процесса «Загрузка данных в приложение»: вариантов использования, последовательности, коммуникации, активностей, состояний (Десктопное приложение), обзора взаимодействия и классов.

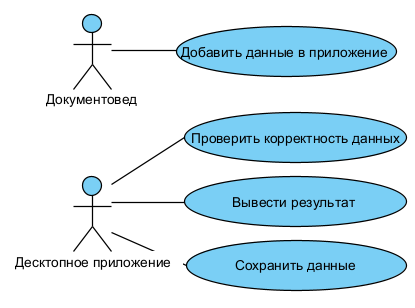


Рисунок 2.1 – Диаграмма вариантов использования «Загрузка данных в приложение»

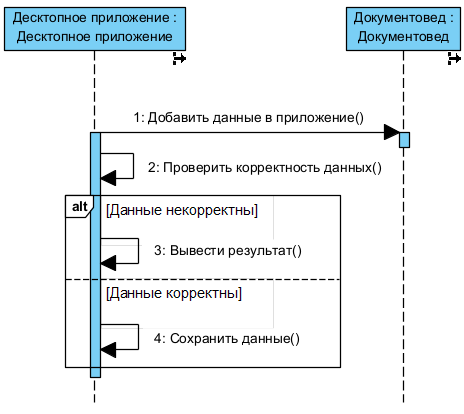


Рисунок 2.2 – Диаграмма последовательности «Загрузка данных в приложение»

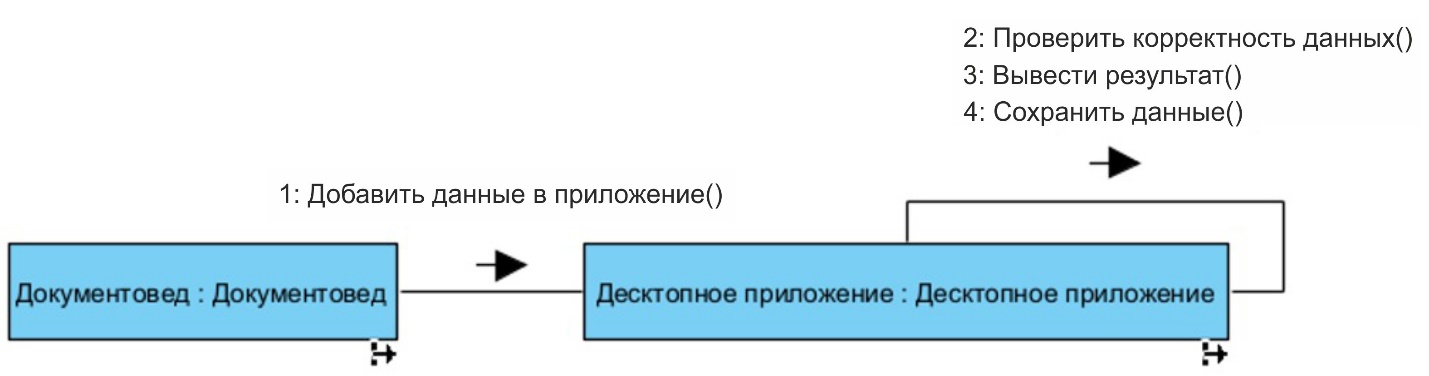


Рисунок 2.3 – Диаграмма коммуникаций «Загрузка данных в приложение»

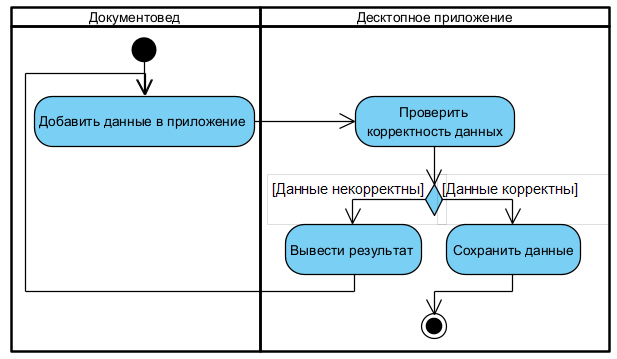


Рисунок 2.4 – Диаграмма активностей «Загрузка данных в приложение»



Рисунок 2.5 – Диаграмма состояний «Десктопное приложение»

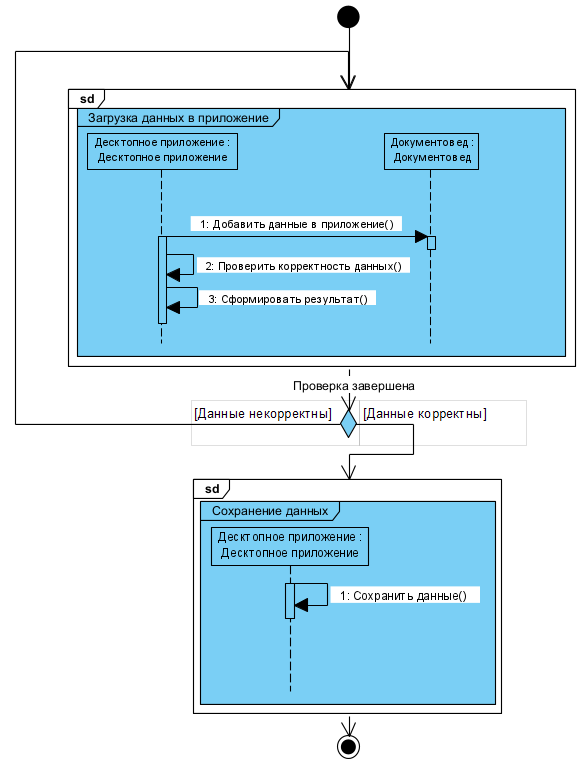


Рисунок 2.6 – Диаграмма обзора взаимодействия «Загрузка данных в приложение»

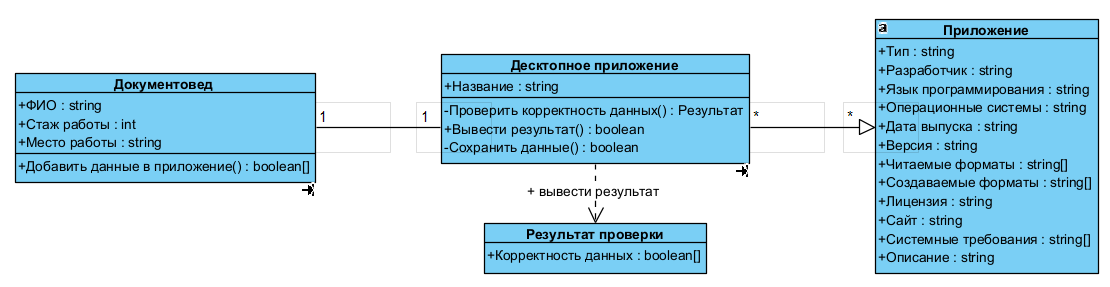


Рисунок 2.7 – Диаграмма классов «Загрузка данных в приложение»

*Моделирование бизнес-процесса «Составление документа с помощью приложения».*

В процессе составления документа с помощью приложения принимают участие: документовед, десктопное приложение, Microsoft Word и Excel.

Процесс начинается с запуска приложения документоведом и импорта в него учебного расписания в виде Excel файла, далее в приложении документовед выбирает преподавателя из расписания, настраивает параметры для генерации отчета и запускает саму генерацию. После того, как приложение создает отчет, происходит его запись в Word и открытие созданного документа для просмотра или редактирования.

На рисунках 2.8-2.13 показаны следующие виды диаграмм для процесса «Составление документа с помощью приложения»: вариантов использования, последовательности, коммуникации, активностей, состояний (Десктопное приложение) и классов.



Рисунок 2.8 – Диаграмма вариантов использования «Составление документа с помощью приложения»

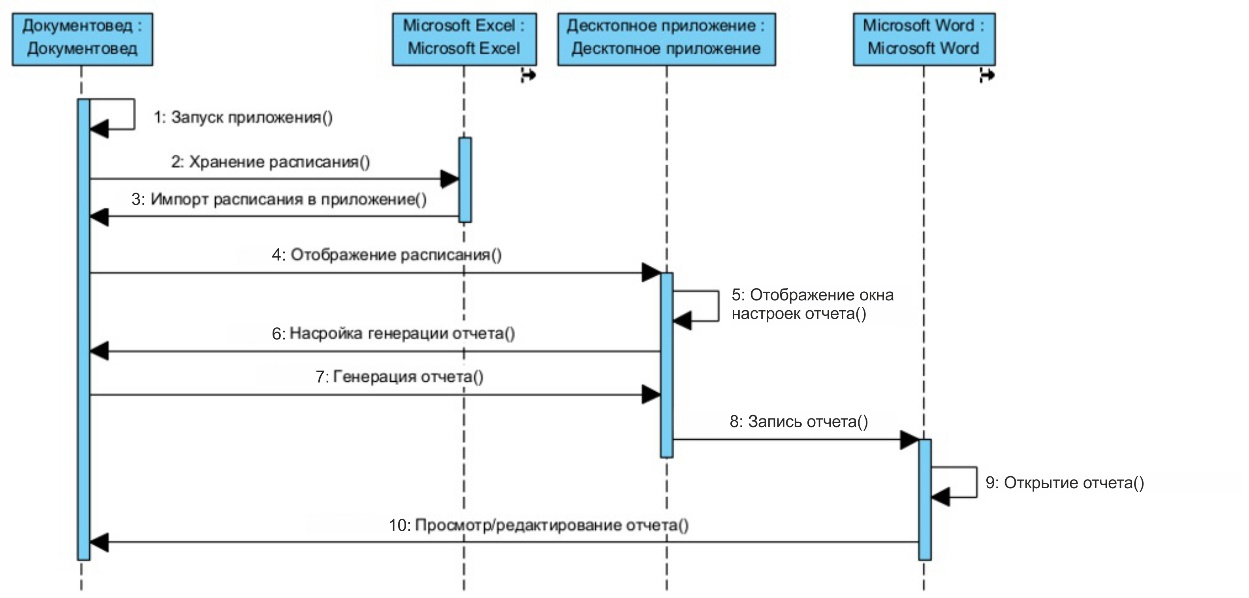


Рисунок 2.9 – Диаграмма последовательности «Составление документа с помощью приложения»

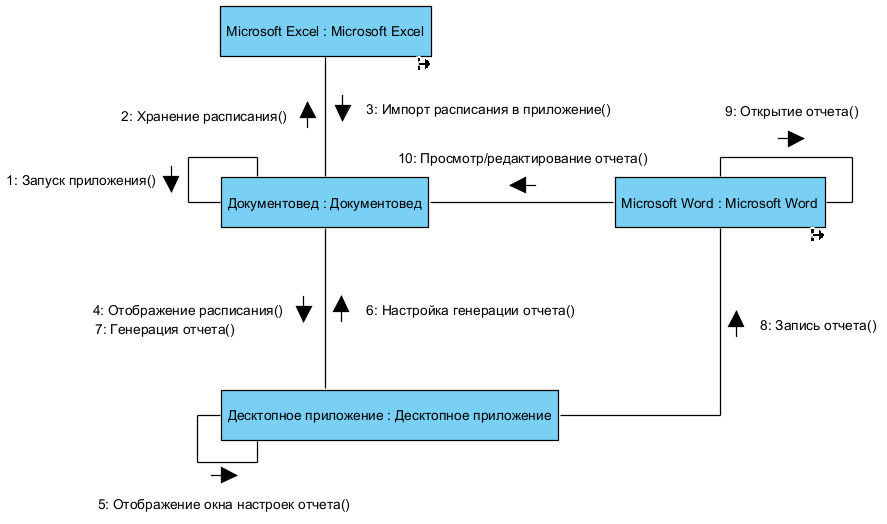


Рисунок 2.10 – Диаграмма коммуникаций «Составление документа с помощью приложения»

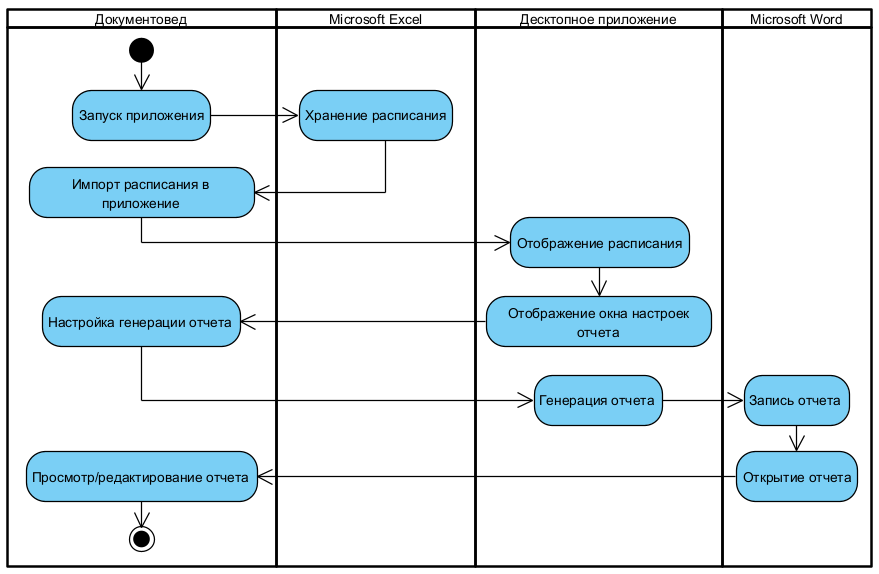


Рисунок 2.11 – Диаграмма активностей «Составление документа с помощью приложения»

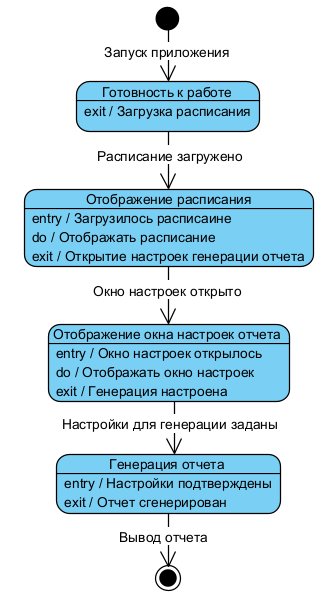


Рисунок 2.12 – Диаграмма состояний «Десктопное приложение»

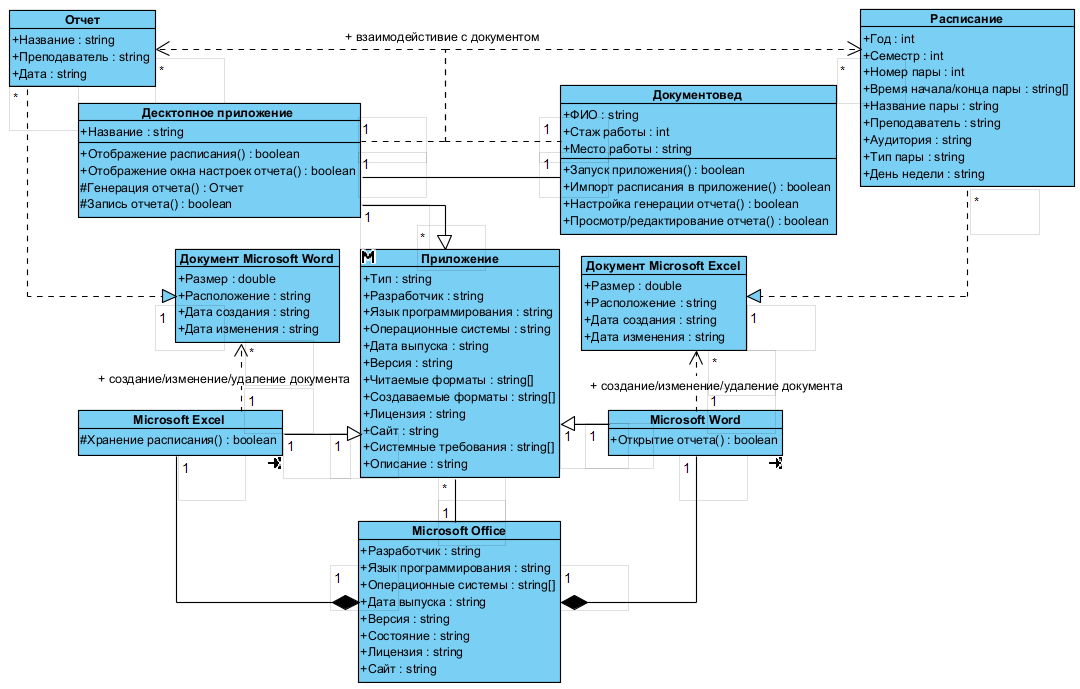


Рисунок 2.13 – Диаграмма классов «Составление документа с помощью приложения»

2.2 Язык программирования

В качестве языка программирования при разработке используется C#, поскольку он:

* объектно-ориентированный язык программирования для платформы .NET;
* основан на строгой компонентной архитектуре и реализует передовые механизмы обеспечения безопасности кода;
* был создан специально для технологии ASP.NET. В то же время на C# полностью написана и сама ASP.NET;
* имеет статическую типизацию,
* поддерживает полиморфизм, перегрузку операторов, делегаты, атрибуты, события, переменные, свойства, обобщённые типы и методы, итераторы, анонимные функции с поддержкой замыканий, LINQ, исключения и комментарии в формате XML;
* имеет поддержку компонентов, надежен и устойчив благодаря использованию «сборки мусора», обработки исключений, безопасности типов[12, 13].

2.3 Выбор среды разработки

На сегодняшний день на рынке информационных технологий существует большое количество различных средств разработки приложений. Ниже приведен список подходящих для разработки на языке программирования C# интегрированных сред:

* Geany;
* Microsoft Visual Studio;
* MonoDevelop;
* JetBrains Rider;
* SharpDevelop.

Geany – среда разработки программного обеспечения, написанная с использованием библиотеки GTK+. Доступна для следующих операционных систем: BSD, GNU/Linux, Mac OS X, Solaris и Windows. Geany распространяется согласно GNU General Public License. Geany не включает в свой состав компилятор. Для создания исполняемого кода используется GNU Compiler Collection или, при необходимости, любой другой компилятор.

Microsoft Visual Studio – линейка продуктов компании Microsoft, включающих интегрированную среду разработки программного обеспечения и ряд других инструментов. Visual Studio включает в себя редактор исходного кода с поддержкой технологии IntelliSense и возможностью простейшего рефакторинга кода. Встроенный отладчик может работать как отладчик уровня исходного кода, так и отладчик машинного уровня. Остальные встраиваемые инструменты включают в себя редактор форм для упрощения создания графического интерфейса приложения, веб-редактор, дизайнер классов и дизайнер схемы базы данных. Visual Studio позволяет создавать и подключать сторонние дополнения для расширения функциональности практически на каждом уровне, включая добавление поддержки систем контроля версий исходного кода, добавление новых наборов инструментов или инструментов для прочих аспектов процесса разработки программного обеспечения[14].

MonoDevelop – свободная среда разработки, предназначенная для создания приложений C#, Java, Boo, Nemerle, Visual Basic .NET, Vala, CIL, C и C++. MonoDevelop является частью проекта Mono. Встроен в дистрибутив Unity3D как средство написания скриптов. С версии Unity 2018.1 перестал поддерживаться.

JetBrains Rider – кроссплатформенная интегрированная среда разработки программного обеспечения для платформы .NET, разрабатываемая компанией JetBrains. Поддерживаются языки программирования C#, VB.NET и F#. В его основе лежит другой продукт JetBrains – ReSharper. Среда поддерживает платформы .NET Framework, .NET и Mono. Работает на операционных системах Windows, macOS, Linux.

SharpDevelop – свободная среда разработки для C#, Visual Basic .NET, Boo, IronPython, IronRuby, F#, C++. Обычно используется как альтернатива Visual Studio .NET. Существует также форк на Mono/GTK+ — MonoDevelop. SharpDevelop 2.0 предоставляет интегрированный отладчик, который использует собственные библиотеки и взаимодействует с исполняющей средой .NET через COM Interop. Хотя SharpDevelop 2.0 использует файлы проекта в формате MSBuild, он по-прежнему может использовать компиляторы от .NET Framework 1.0 и 1.1, а также от Mono. На текущий момент поддерживается версия .Net Frameworks от 2.0 до 4.5.1.

Сравнение интегрированных сред разработки представлено в виде таблице 2.1.

Таблица 2.1 – Сравнение интегрированных сред разработки

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| IDE | Geany | Visual Studio | MonoDevelop | Rider | SharpDevelop |
| Лицензия | GPL | Проприетарная | GPL, LGPL | Проприетарная | LGPL |
| Windows | Да | Да | Да | Да | Да |
| Linux | Да | Нет | Да | Да | Нет |
| maxOS | Да | Да | Да | Да | Нет |
| Проверка синтаксиса | Да | Да | Да | Да | Да |
| Отладчик | Да | Да | Да | Да | Да |
| Разработка GUI | Нет | Да | Да | Да | Да |
| Профилирование | Нет | Да | - | Да | Да |
| Покрытие кода | - | Да | Да | - | - |
| Автодополнение | Да | Да | Да | Да | Да |
| Статический анализ кода | Да | Да | - | Да | - |
| Дизайн на основе GUI | Нет | Да | Да | - | Да |
| Браузер классов | Да | Да | Да | Да | - |

В результате сравнения интегрированных сред разработки можно сказать, что лучшим выбором является Microsoft Visual Studio. Эта среда предоставляет самые передовые функции, которые ищут разработчики: редактор исходного кода с поддержкой технологии IntelliSense и возможностью простейшего рефакторинга кода; встроенный отладчик; редактор форм; веб-редактор; дизайнер классов; дизайнер схемы базы данных.

2.4 Система управления базами данных (СУБД)

В качестве системы управления базами данных в разрабатываемом приложении используется SQLite.

SQLite – компактная встраиваемая система управления базами данных. SQLite не использует парадигму клиент-сервер, то есть движок SQLite не является отдельно работающим процессом, с которым взаимодействует программа, а представляет собой библиотеку, с которой программа компонуется, и движок становится составной частью программы. Таким образом, в качестве протокола обмена используются вызовы функций библиотеки SQLite. Такой подход уменьшает накладные расходы, время отклика и упрощает программу. SQLite хранит всю базу данных в единственном стандартном файле на том компьютере, на котором исполняется программа. Простота реализации достигается за счёт того, что перед началом исполнения транзакции записи весь файл, хранящий базу данных, блокируется[15].

SQLite возможно использовать как на встраиваемых системах, так и на выделенных машинах с гигабайтными массивами данных.

SQLite поддерживает динамическое типизирование данных, то есть тип данных связан со значением, а не с полем таблицы. Возможные типы значений: INTEGER, REAL, TEXT и BLOB. Также поддерживается специальное значение NULL.

2.5 Модель данных (ER-модель)

На основе анализа предметной области «Нагрузка преподавателя» можно спроектировать следующие сущности: Преподаватель, Занятие, Расписание, Кафедра, Период обучения, Нагрузка. В таблицах 2.2-2.7 представлено описание каждой сущности.

Таблица 2.2 – Описание сущности «Преподаватель»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Поле | Тип данных | Обязательное | PK | Описание |
| Id | int | + | + | Уникальный идентификатор |
| Surname | string | + | - | Фамилия |
| Name | string | + | - | Имя |
| Patronymic | string | + | - | Отчество |
| Post | string | - | - | Должность |
| Rank | string | - | - | Ученая степень, звание |
| Group | string | + | - | Группа преподавателя |
| Rate | int | - | - | Почасовая оплата |
| Contract | string | - | - | Номер договора |

Таблица 2.3 – Описание сущности «Занятие»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Поле | Тип данных | Обязательное | PK | Описание |
| Id | int | + | + | Уникальный идентификатор |
| Groups | string | - | - | Шифры групп |
| Discipline | string | - | - | Дисциплина |
| Type | string | - | - | Тип занятия |
| Place | string | - | - | Место проведения занятия |

Таблица 2.4 – Описание сущности «Расписание»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Поле | Тип данных | Обязательное | PK | Описание |
| Id | int | + | + | Уникальный идентификатор |
| TeacherId | int | + | - | Уникальный идентификатор преподавателя |
| LessonId | int | + | - | Уникальный идентификатор занятия |
| Day | string | + | - | День недели |
| LessonNumber | int | + | - | Номер занятия |
| Parity | string | + | - | Недельная паритетность |

Таблица 2.5 – Описание сущности «Кафедра»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Поле | Тип данных | Обязательное | PK | Описание |
| Id | int | + | + | Уникальный идентификатор |
| TeacherId | int | + | - | Уникальный идентификатор заведующего |
| Name | string | + | - | Название кафедры |
| ShortName | string | + | - | Аббревиатура |

Таблица 2.6 – Описание сущности «Период обучения»

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Поле | Тип данных | | Обязательное | | PK | Описание |
| Id | int | | + | | + | Уникальный идентификатор |
| Course | int | | + | | - | Курс |
| Specialization | string | | + | | - | Специальность |
| LessonsStart | | date | | + | - | Начало занятий |
| LessonsWeeks | | int | | + | - | Количество недель занятий |

Продолжение таблицы 2.6

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Поле | Тип данных | Обязательное | PK | Описание |
| SessionStart | date | + | - | Начало сессии |
| SessionWeeks | int | + | - | Количество недель сессии |

Таблица 2.7 – Описание сущности «Нагрузка»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Поле | Тип данных | Обязательное | PK | Описание |
| Id | int | + | + | Уникальный идентификатор |
| TeacherId | int | + | - | Уникальный идентификатор преподавателя |
| EducationForm | string | + | - | Форма обучения |
| EducationLevel | string | + | - | Уровень высшего образования |
| Discipline | string | + | - | Дисциплина |
| Groups | string | + | - | Наименование направления подготовки, специальности |
| Control | string | + | - | Вид контроля |
| Course | string | + | - | Курс |
| Semester | string | + | - | Семестр |
| CountWeeks | int | + | - | Количество недель |
| CountStudents | int | + | - | Количество студентов |
| CountFlows | int | + | - | Количество потоков |
| CountGroups | int | + | - | Количество групп |
| LectureHoursPerFlow | int | - | - | Лекции: часов на поток |
| LectureAllHours | int | - | - | Лекции: всего часов |
| PracticalHoursPerFlow | int | - | - | Практические: часов на поток |
| PracticalAllHours | int | - | - | Практические: всего часов |
| LaboratoryHoursPerFlow | int | - | - | Лабораторные: часов на поток |
| LaboratoryAllHours | int | - | - | Лабораторные: всего часов |
| SeminarHoursPerFlow | int | - | - | Семинарские: часов на поток |
| SeminarAllHours | int | - | - | Семинарские: всего часов |
| Consultations | int | - | - | Консультации |
| Tests | int | - | - | Зачеты |
| Exams | int | - | - | Экзамены |
| PracticeCountDays | int | - | - | Практика: дней |
| PracticeHoursPerGroup | int | - | - | Практика: количество часов на группу в день |

Продолжение таблицы 2.7

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Поле | Тип данных | Обязательное | PK | Описание |
| PracticeAllHours | int | - | - | Практика: всего часов |
| CourseDesign | int | - | - | Курсовое проектирование |
| DiplomaCountStudents | int | - | - | Дипломное: количество дипломников |
| DiplomaHoursPerStudent | int | - | - | Дипломное: количество часов на одного дипломника |
| DiplomaAllHours | int | - | - | Дипломное: всего часов |
| GEKCountMembers | int | - | - | ГЭК: число членов ГЭК |
| GEKAllHours | int | - | - | ГЭК: количество часов |
| RGRorEssay | int | - | - | РГР, рефераты, эссе |
| ManagementMagistrants | int | - | - | Руководство: магистрантами |
| ManagementAspirantsOrInterns | int | - | - | Руководство: аспирантами, стажерами |
| ManagementCathedra | int | - | - | Руководство: кафедрой |
| ManagementFaculty | int | - | - | Руководство: факультетом |
| Aspirantura | int | - | - | Аспирантура |
| FullPartTimeEducation | int | - | - | Очно-заочная форма обучения |
| PartTimeEducation | int | - | - | Заочная форма обучения |
| AdmissionCommittee | int | - | - | Приемная комиссия |
| AllHours | int | + | - | Всего часов |

На рисунке 2.14 показана ER-диаграмма базы данных информационной системы «Учёт почасовой нагрузки преподавателя в университете». С помощью ER-диаграммы были продемонстрированы ключевые сущности и обозначены связи, которые могут устанавливаться между этими сущностями. При реализации базы данных приложения было произведено преобразование схемы, созданной на основе ER-диаграммы, в конкретную схему базы данных на основе реляционной модели данных.

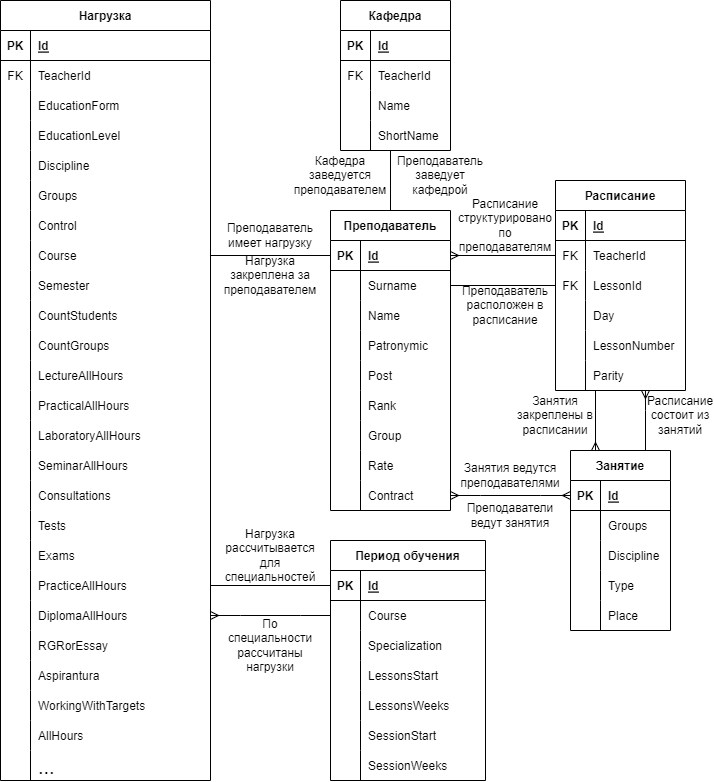


Рисунок 2.14 – ER-диаграмма базы данных информационной системы «Учёт почасовой нагрузки преподавателя в университете»

2.6 Проектирование интерфейса информационной системы

Основными действующими экранами системы являются главное окно (отвечает за отображение расписания и загрузку файлов) и окно создания отчета (отвечает за настройку создаваемого отчета).

Главное окно должно содержит следующие элементы:

* элемент, отображающий расписание;
* кнопки для загрузки расписания, нагрузки, дополнительных данных;
* элементы выбора необходимых листов из загруженного файла с дополнительными данными;
* кнопку перехода на окно настроек отчета.

Окно создания отчета состоит из двух вкладок, предназначенных для генерации разных видов документов, и содержит следующие элементы:

* элемент, с помощью которого выбирается преподаватель;
* блок, содержащий информацию по выбранному преподавателю;
* блок, содержащий информацию по кафедре;
* элементы для выбора периода генерации отчета;
* элементы, отображающие информацию из нагрузки по выбранному преподавателю;
* элемент, отображающий отчет, который будет создан;
* кнопка загрузки отчетов, на основе которых будет создаваться новый документ;
* элемент, отображающий загруженные отчеты;
* кнопка сохранения отчета;
* чекбокс, отвечающий за открытие отчета после сохранения.

При запуске приложения открывается главное окно. Переход с главного окна на окно настроек отчета осуществляется посредством нажатия на соответствующую кнопку. После настройки и сохранения отчета в окне создания отчета, оно остается активно. Передача управления обратно к главному окну переходит только после закрытия окна создания отчета.

Основные пункты используемой графики:

* используемые цвета – белый и серый;
* шрифт и его размер по умолчанию (текущий системный шрифт операционной системы, его базовые значения размера и межстрочного интервала).

На рисунках 2.15-2.17 показаны интерфейсы окон разрабатываемой системы, созданные в Visual Studio.

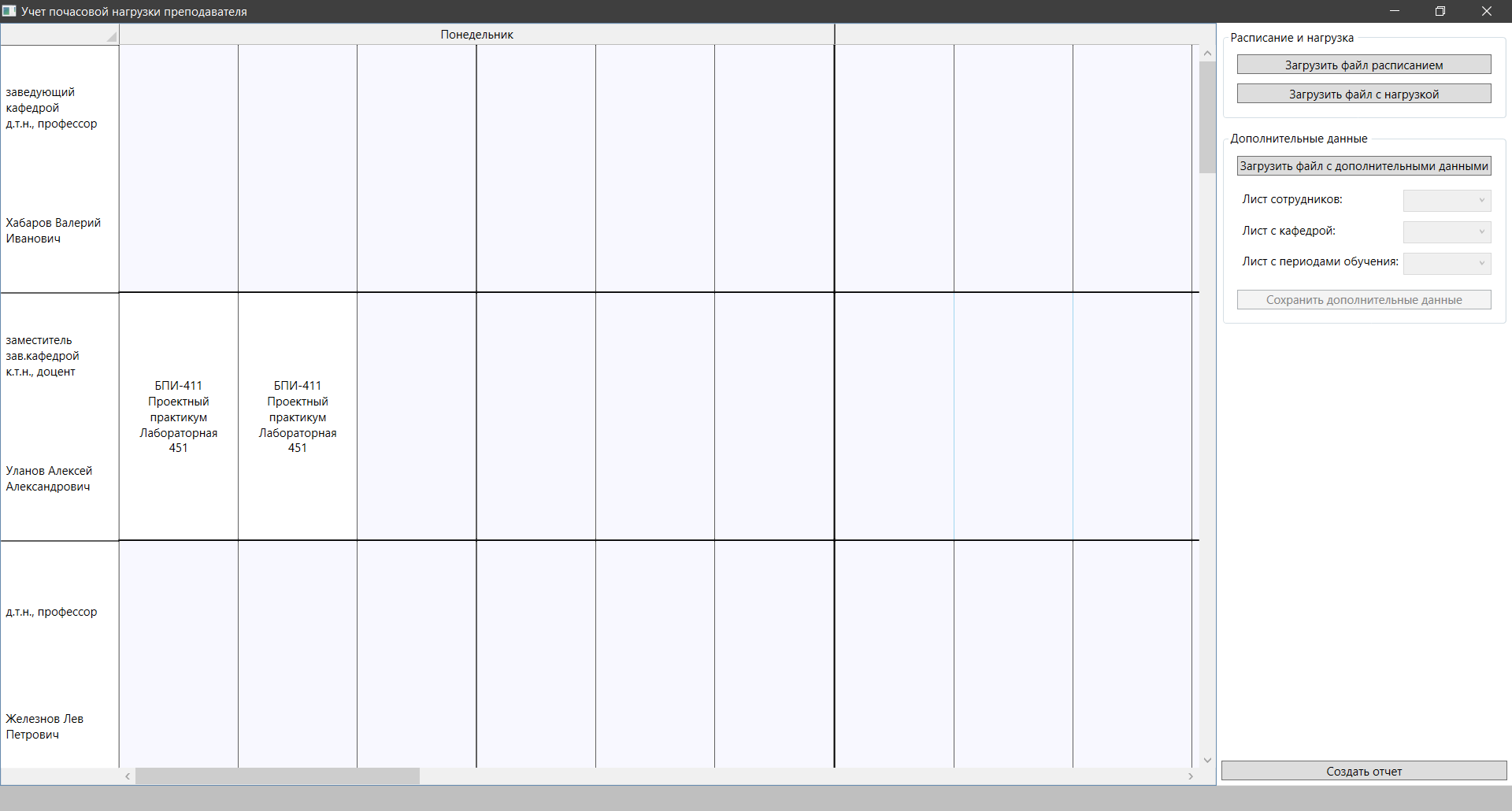


Рисунок 2.15 – Интерфейс главного окна

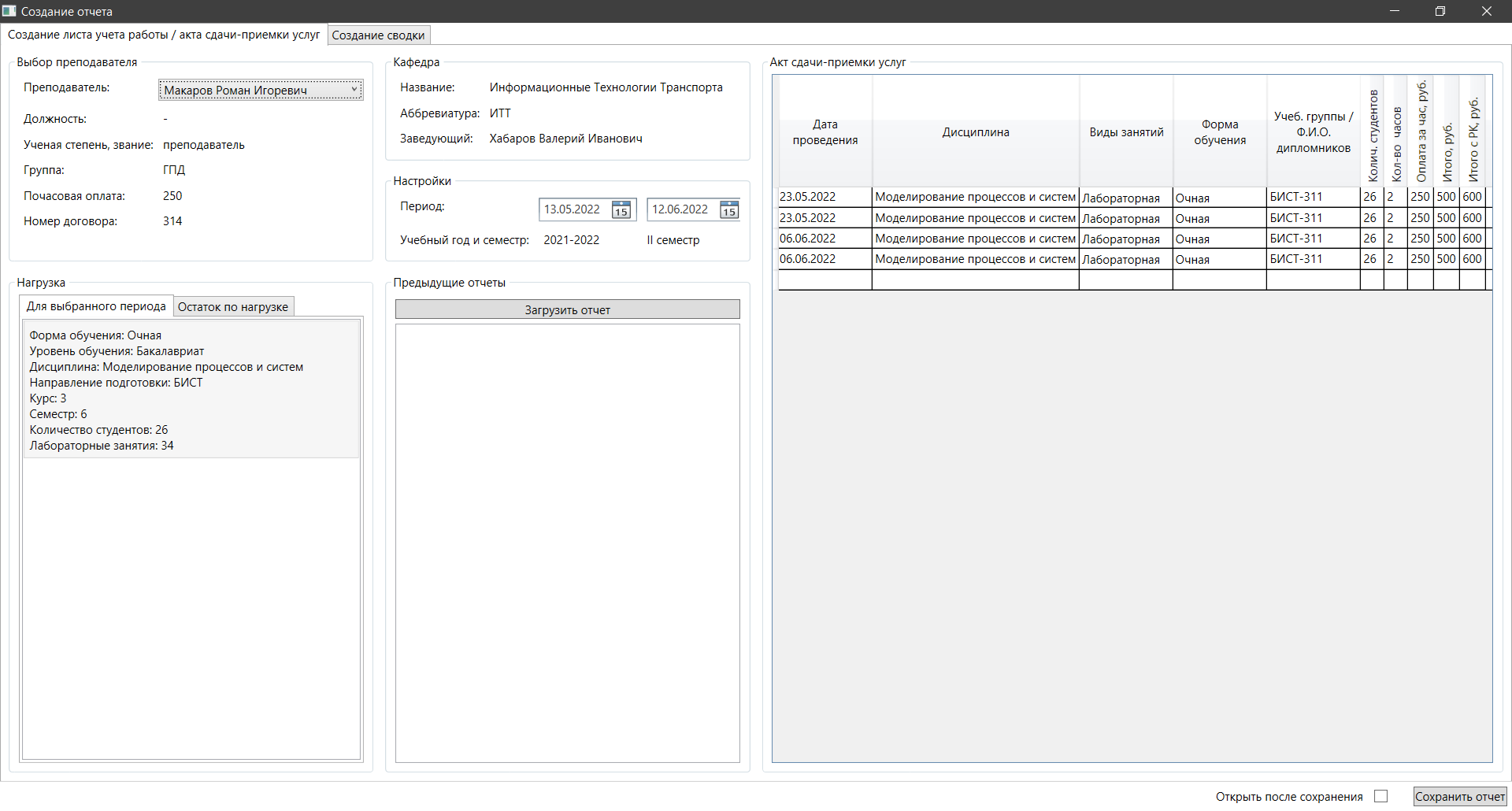


Рисунок 2.16 – Интерфейс окна создания отчета. Вкладка для создания листа учета работы и акта сдачи-услуг

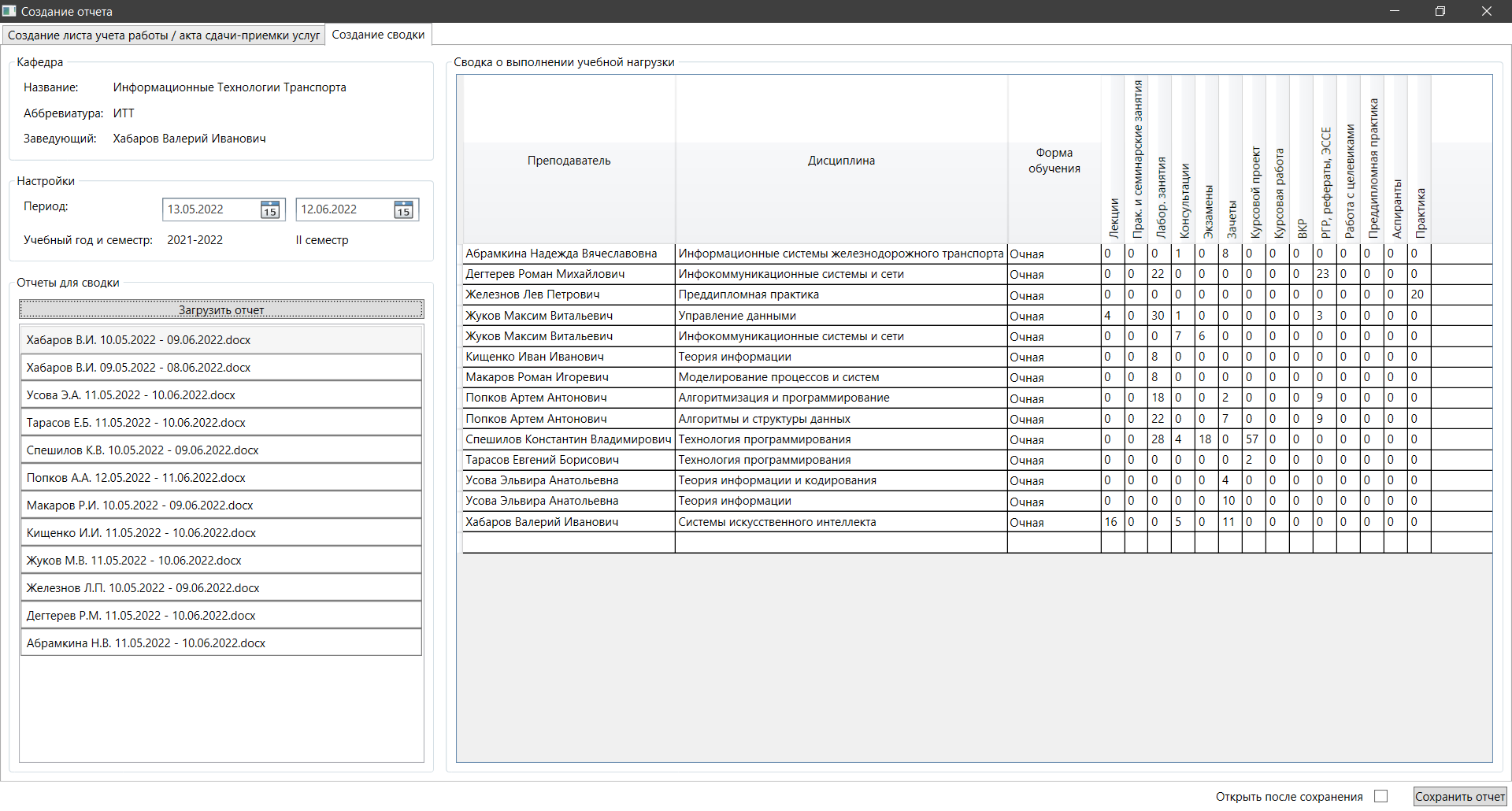


Рисунок 2.16 – Интерфейс окна создания отчета. Вкладка для создания сводки

2.7 Выводы по второму разделу

Во втором разделе были выбрана и описана методология моделирования бизнес-процессов UML; смоделированы бизнес-процессы предметной области: «Загрузка данных в приложение» и «Составление документа с помощью приложения»; выбраны современные инструментальные средства для проектирования и разработки информационной системы, а именно: язык программирования – C#, среда разработки – Microsoft Visual Studio, система управления базами данных – SQLite; создана модель данных информационной системы, описанная в виде ER-диаграммы базы данных; а также спроектирован интерфейс информационной системы.

3 Создание приложения

В третьем разделе проведены разработка, тестирование и апробация системы; описаны ее функциональные особенности и условия обеспечения работоспособности; описаны основные этапы выполнения операций в программе.

3.1 Описание функциональных особенностей информационной системы

*Загрузка данных в приложение.*

Система имеет возможность загрузки следующих документов:

* преподавательское расписание;
* продолжительность периодов обучения групп;
* информация о количестве почасовой учебной работы;
* информация о преподавателях, в том числе норма почасовой оплаты труда.

Документы загружаются в систему в формате Excel-документов. Загружаемые в систему данные проверяются на корректность. При некорректности загружаемых данных происходит информирование пользователя об ошибке. В случае корректности данных – сохранение в базу данных приложения.

*Составление документа с помощью приложения.*

Создание отчета происходит в отдельном окне. Система имеет возможность составлять один из следующих документов:

* лист учета работы преподавателя-почасовика;
* акт сдачи-приемки услуг преподавателя;
* сводка о выполнении учебной нагрузки.

В приложении А представлены шаблоны этих документов.

Для составления документов используется информация, автоматически загружаемая из базы данных приложения. При генерации документа пользователю необходимо настроить различные параметры. После генерации документа он может быть сохранен в Word и открыт после сохранения. Также, в приложении имеется возможность учитывать созданные ранее отчеты при генерации.

3.2 Условия обеспечения работоспособности информационной системы

*Численность и квалификация персонала системы.*

Для работы системы необходимы:

* администратор, производящий настройку системы;
* пользователь, работающий в системе.

Пользователь должен иметь опыт работы с ОС Windows, навык работы с Microsoft Word и Excel, а также обладать знаниям в соответствующей предметной области.

*Информационное обеспечение системы.*

Состав данным по документам, загружаемым в систему:

* преподавательское расписание (семестр, день недели, номер недели (первая или вторая), время пары (номер пары), название занятия, тип занятия, аудитория, группа, преподаватель, квалификация);
* продолжительность периодов обучения групп (курс, группа, начало занятий, количество учебных недель, начало экзаменационной сессии, количество недели в сессии);
* информация о количестве почасовой учебной работы (дисциплина, направление подготовки (специальность), вид контроля, курс, семестр, количество недель, данные к расчету (контингент студентов, количество потоков и групп), количество часов на различные виды деятельности, всего часов, доля ставки, примечание, преподаватель, квалификация, форма обучения, уровень высшего образования, итого по форме обучения, итого по семестру, итого);
* норма почасовой оплаты труда преподавателя (преподаватель, квалификация, норма почасовой оплаты труда).

В систему данные заносятся посредством загрузки в нее Excel-документов и хранится в базе данных приложения. При запуске приложения, происходит загрузка данных из базы данных.

*Программное обеспечение системы.*

Перечень программных средств, необходимых для работы системы:

* Microsoft Word – текстовый процессор, предназначенный для создания, просмотра, редактирования и форматирования документов (входит в состав Microsoft Office);
* Microsoft Excel – программа для работы с электронными таблицами (входит в состав Microsoft Office).

*Техническое обеспечение системы.*

Для функционирования системы необходимо техническое обеспечение со следующими характеристиками.

* процессор с тактовой частотой не менее 1 ГГц;
* оперативная память с 2 ГБ и более;
* жесткий диск с 10 ГБ свободного места;
* видеоадаптер совместимый с DirectX 9 или более поздней версии;
* операционная система – Windows 10;
* .NET Framework 4.8 и выше.

3.3 Выполнение операций в программе

Для запуска системы необходимо открыть .exe файл самой программы. После запуска приложения, на экране отображается главное окно, показанное на рисунке 3.1, в которое выводится последнее загруженное расписание, и из которого можно загружать в систему файлы с расписанием, нагрузкой и дополнительными данными, а также переходить к окну создания отчета.

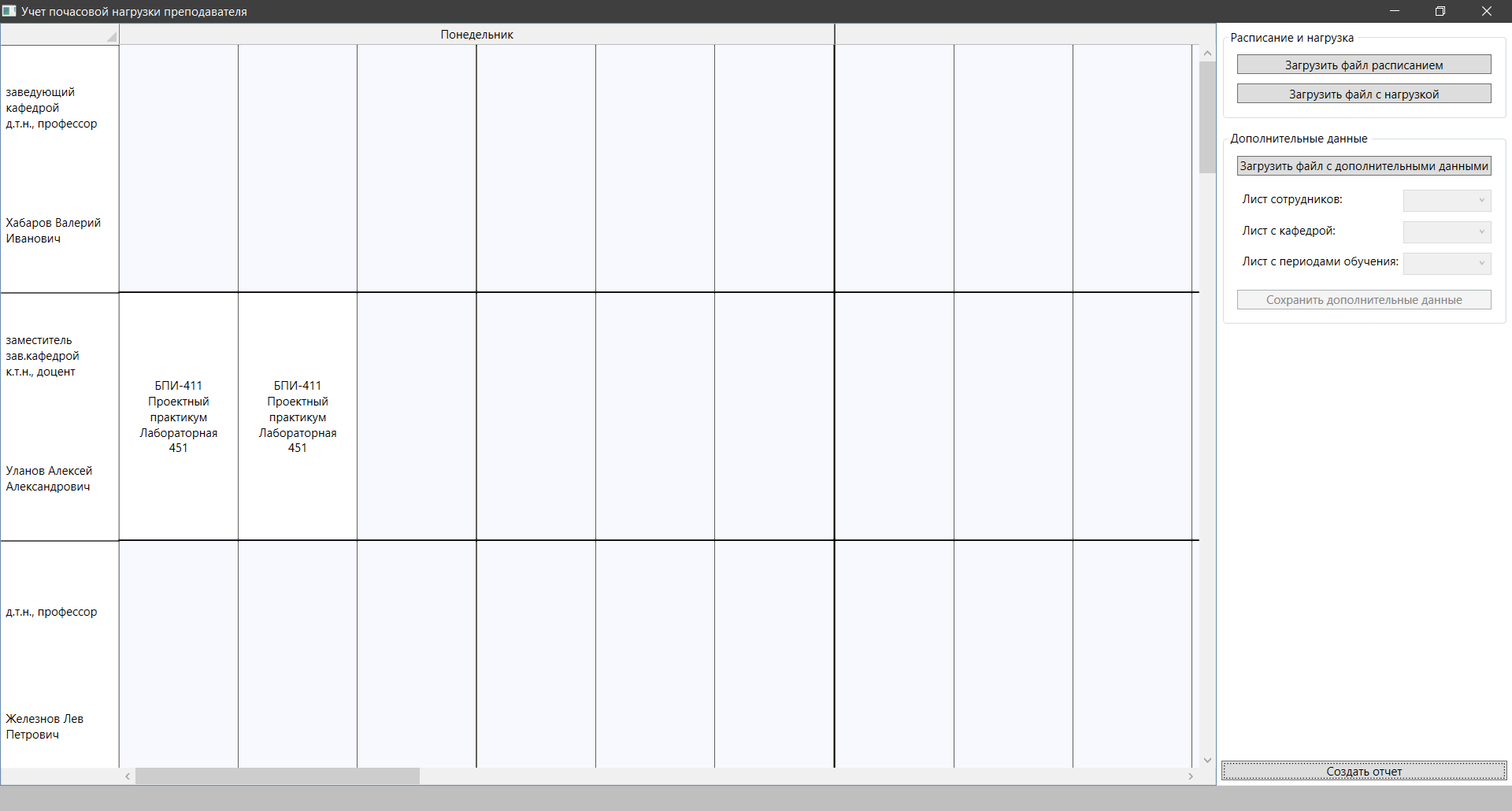


Рисунок 3.1 – Главное окно

Выход из приложения возможен при нажатии на кнопку «Закрыть», а также можно воспользоваться сочетанием клавиш Alt+F4.

Работа с запущенной системой состоит из выполнения следующих этапов.

Загрузка в систему Excel-файлов с расписанием, нагрузкой и дополнительными данными. Кнопки для загрузки соответствующих файлов расположены в правой части главного окна. При загрузке файла с дополнительными данными необходимо выбрать необходимые листы (преподаватели, кафедры и периоды обучения групп).

Для перехода к окну создания отчета необходимо нажать на кнопку «Создать отчет», расположенную в правом нижнем углу главного окна.

Окно создания отчета состоит из двух вкладок – в первой происходит создание листа учета работы преподавателя-почасовика и акта сдачи-приемки услуг преподавателя; во второй – сводки о выполнении учебной нагрузки.

Первая вкладка окна показана на рисунке 3.2. В ней отображаются: данные по выбранному преподавателю, данные по кафедре, выбранный период для создания отчета, данные из нагрузки (для выбранного периода и все), предыдущие загруженные отчеты, сгенерированный отчет.

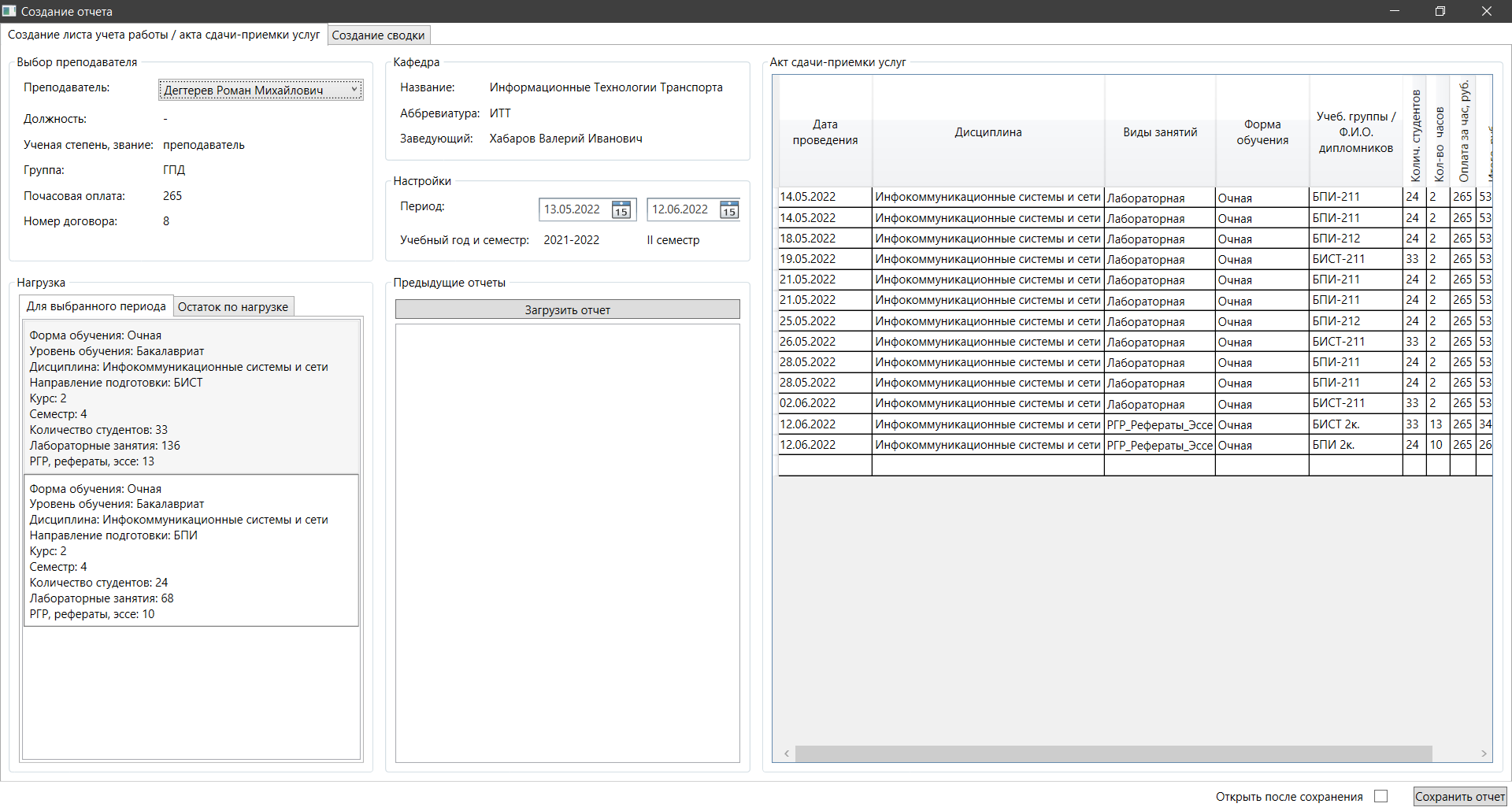


Рисунок 3.2 – Окно создания отчета. Вкладка для создания листа учета работы преподавателя-почасовика и акта сдачи-приемки услуг преподавателя

Вторая вкладка окна показана на рисунке 3.3. В ней отображаются данные по кафедре, выбранный период для создания отчета, загруженные отчеты для формирования сводки, сгенерированный отчет.

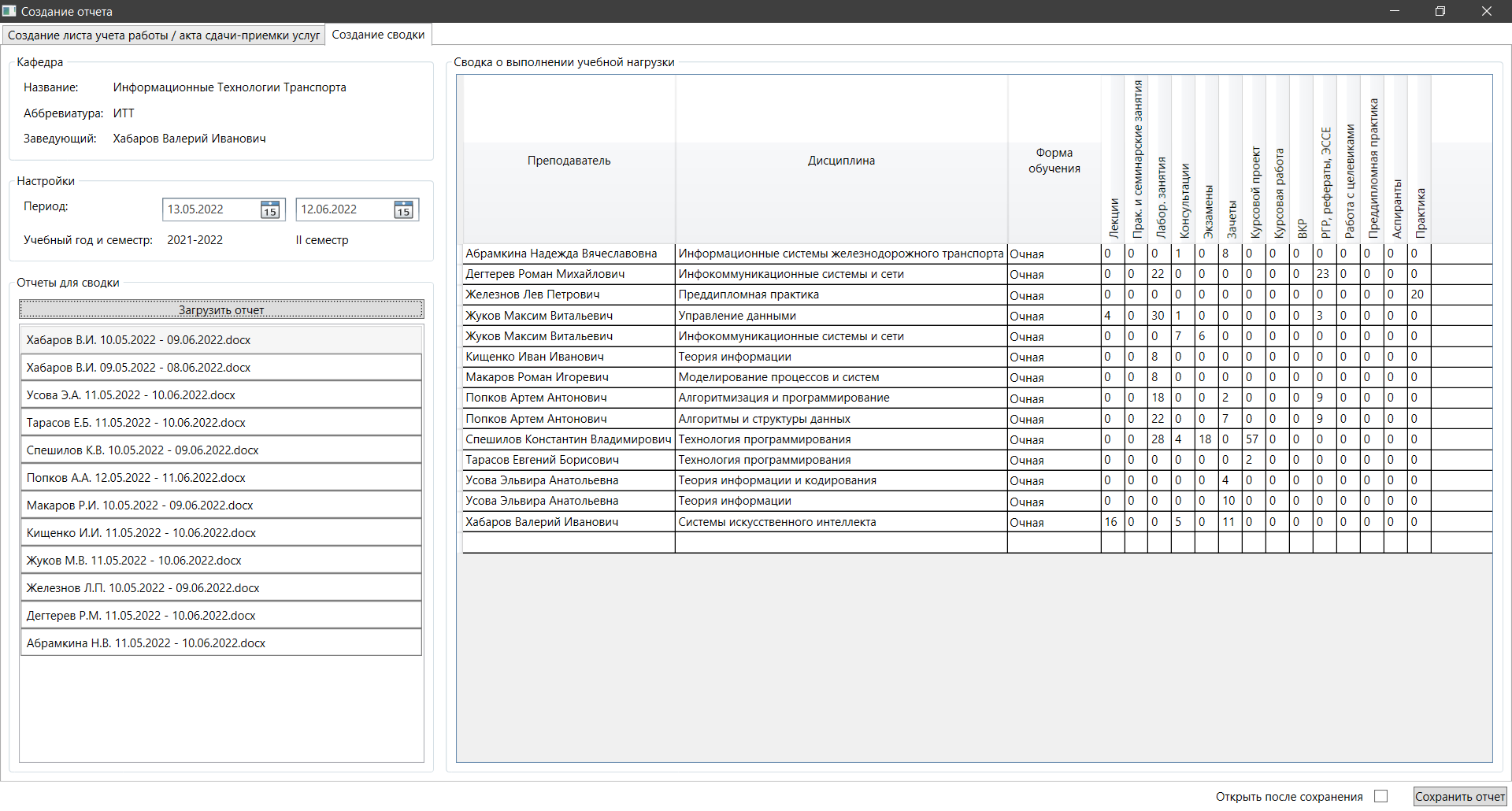


Рисунок 3.3 – Окно создания отчета. Вкладка для сводки о выполнении учебной нагрузки

Для формирования отчета в первой вкладке необходимо в выпадающем списке выбрать преподавателя, для которого будет создаваться отчет, и настроить период. Сразу после этого будет сгенерирован отчет, отображаемый в правой части окна. Вид отчета зависит от того, к какой группе относится выбранный преподаватель – лист учета работы преподавателя-почасовика составляется для штатных, внутренних и сторонних преподавателей; акт сдачи-приемки услуг преподавателя составляется для преподавателей, нанятых по гражданско-правовому договору. В созданном отчете можно изменить данные в ячейках, добавить и удалить строки. Добавление строк в отчет возможно двумя способами: вручную и выбором из нагрузки.

Также, имеется возможность загрузки ранее созданных отчетов. На их основе высчитываются оставшиеся часы нагрузки в разрезе дисциплины и вида работы для преподавателя, которые будут учитываться при создании отчета. На рисунках 3.4-3.7 показано различие при генерации отчета без учета предыдущих отчетов и с их учетом.

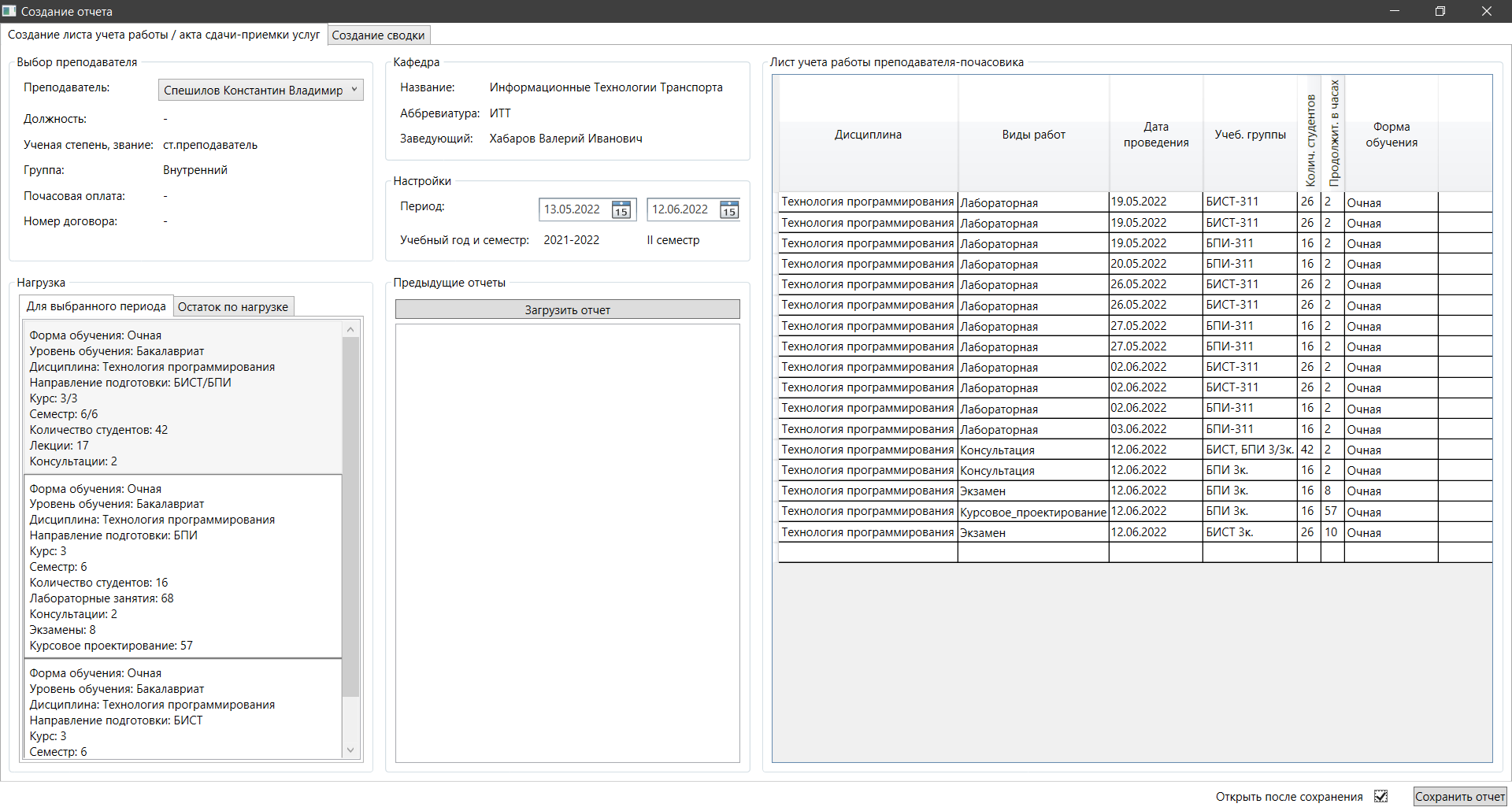


Рисунок 3.4 – Генерация отчета без учета предыдущих

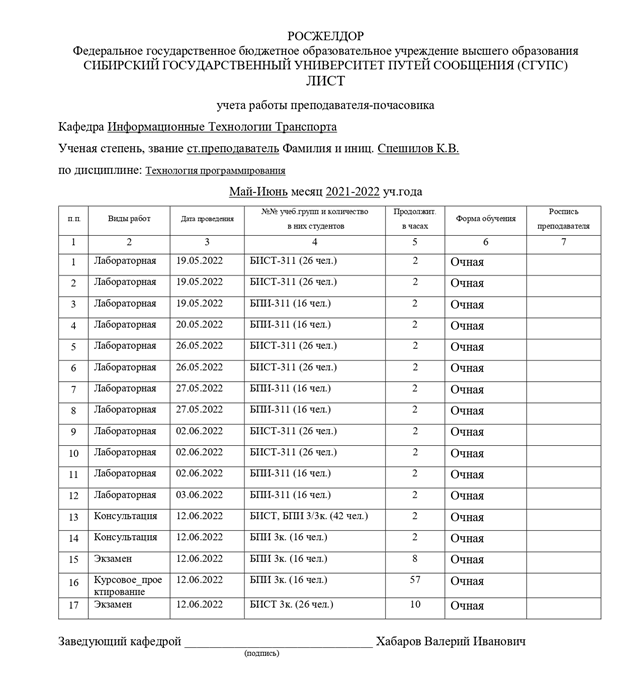


Рисунок 3.5 – Созданный отчет без учета предыдущих

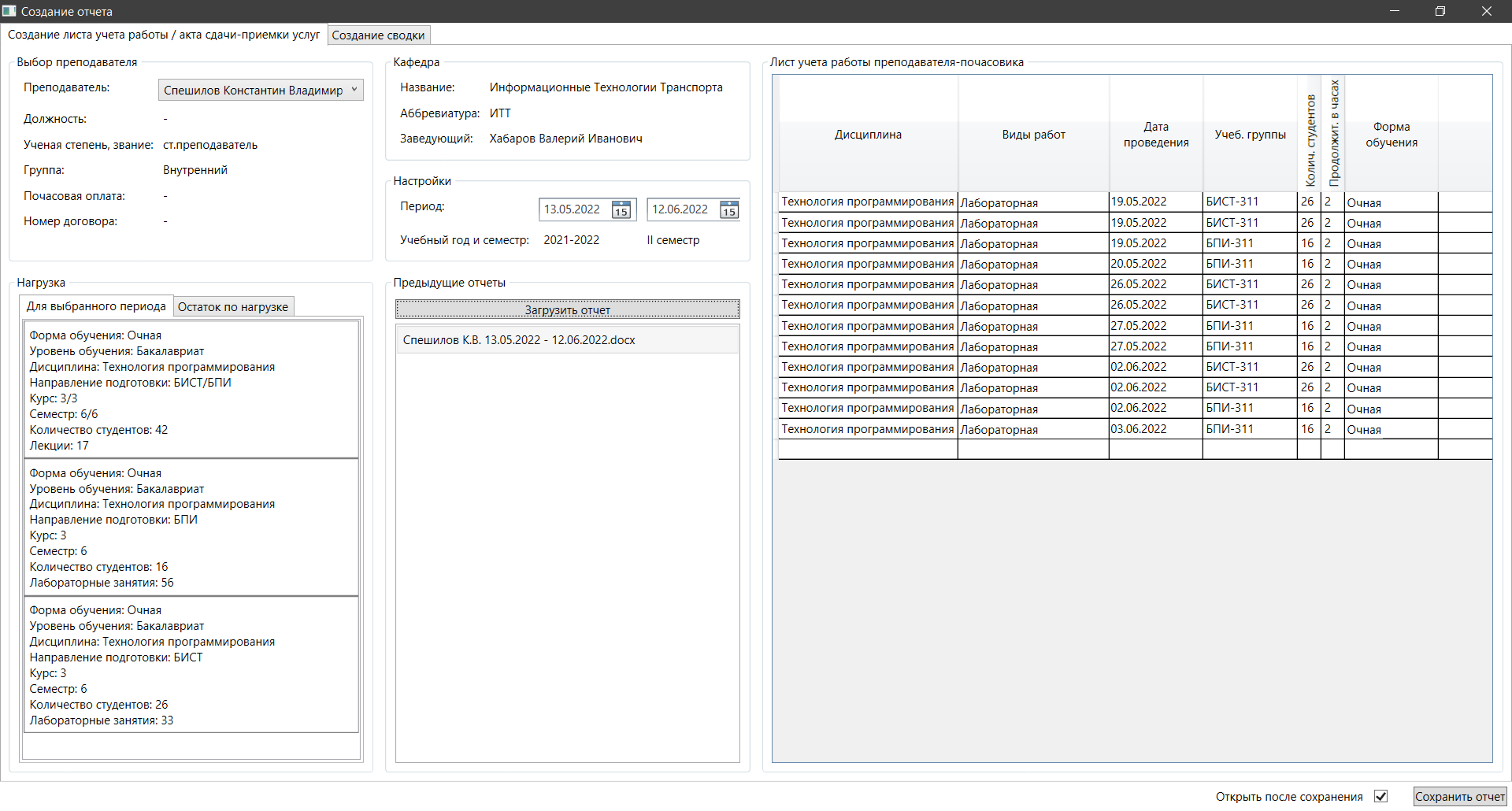


Рисунок 3.6 – Генерация отчета с учетом предыдущих



Рисунок 3.7 – Созданный отчет с учетом предыдущих

Для формирования отчета во второй вкладке необходимо настроить период и загрузить отчеты, на основе которых будет формироваться сводка. Сразу после этого будет сгенерирован отчет, отображаемый в правой части окна. В созданном отчете можно изменить данные в ячейках, добавить и удалить строки.

На рисунках 3.8 и 3.9 показана генерация сводки и созданный документ.

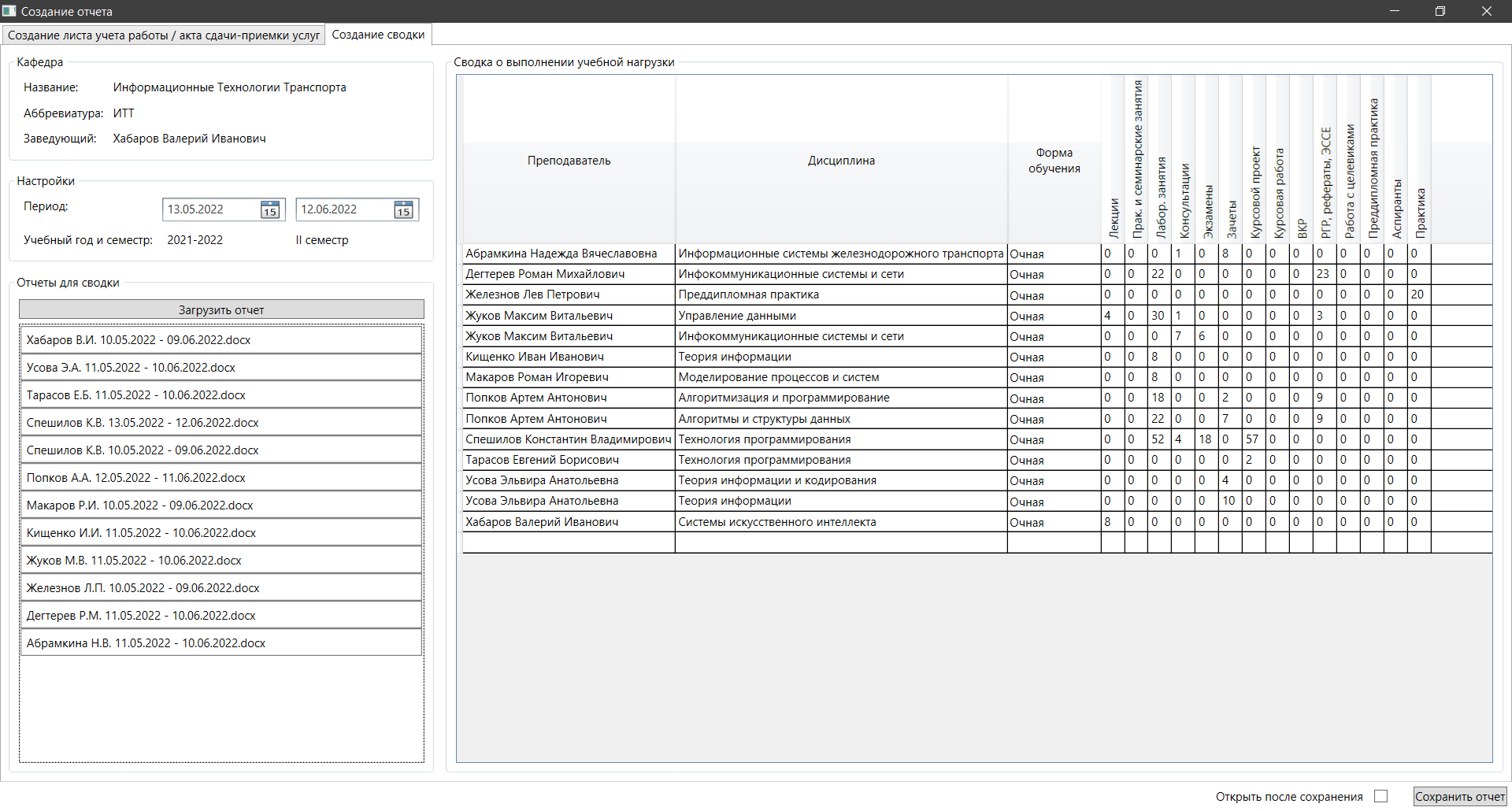


Рисунок 3.8 – Генерация сводки



Рисунок 3.9 – Созданный документ

Сохранение отчета происходит нажатием на кнопку «Сохранить отчет» в правом нижнем углу окна, указав открыть или нет отчет после сохранения.

3.4 Выводы по третьему разделу

В третьем разделе были проведена разработка, тестирование и апробация системы; описаны ее функциональные особенности; условия обеспечения работоспособности, а именно: численность и квалификация персонала системы, информационное обеспечение системы, программное обеспечение системы и техническое обеспечение системы; а также описаны основные этапы выполнения операций в программе.

ЗАКЛЮЧЕНИЕ

В ходе выполнения выпускной квалификационной работы было разработано программное обеспечение учета почасовой нагрузки преподавателя в университете.

В первом разделе была проанализирована предметная область «Нагрузка преподавателя», в которой разрабатывается программное обеспечение «Учёт почасовой нагрузки преподавателя в университете»; а также проведен анализ существующих аналогов программного обеспечения учета индивидуальной нагрузки преподавателя.

Во втором разделе были выбрана и описана методология моделирования бизнес-процессов UML; смоделированы бизнес-процессы предметной области: «Загрузка данных в приложение» и «Составление документа с помощью приложения»; выбраны современные инструментальные средства для проектирования и разработки информационной системы, а именно: язык программирования – C#, среда разработки – Microsoft Visual Studio, система управления базами данных – SQLite; создана модель данных информационной системы, описанная в виде ER-диаграммы базы данных; а также спроектирован интерфейс информационной системы.

В третьем разделе были проведена разработка, тестирование и апробация системы; описаны ее функциональные особенности; условия обеспечения работоспособности, а именно: численность и квалификация персонала системы, информационное обеспечение системы, программное обеспечение системы и техническое обеспечение системы; а также описаны основные этапы выполнения операций в программе.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. stu.ru. Информационные технологии транспорта - О кафедре. [Электронный ресурс]. - URL: https://www.stu.ru/education/index.php?page=236 (дата обращения: 05.05.2022).
2. stu.ru. Направление – Информационные системы и технологии. [Электронный ресурс]. – URL: https://www.stu.ru/abiturient/index.php?page=1388 (дата обращения: 05.05.2022).
3. Регламент планирования, организации и контроля нагрузки [Электронный ресурс]. – URL: https://www.rea.ru/ru/org/managements/uchmetupr/Documents/Регламент%202-ой%20пол%20дня%20июнь%202016.pdf. (Дата обращения: 12.05.2022).
4. Индивидуальная нагрузка преподавателя [Электронный ресурс]. – URL: https://www.hse.ru/studyspravka/indnagruzka (дата обращения: 12.05.2022).
5. Порядок организации контактной работы обучающихся с преподавателем в ФГБОУ ВО «Сибирский государственный университет путей сообщения»: введен в действие приказом ректора СГУПС №168 от 17 октября 2021 г. с изменениями, принятыми методическим советом 13 мая 2021 г. Протокол №5.
6. БИТ:Учет нагрузки преподавателей [Электронный ресурс]. – URL: https://www.tadviser.ru/index.php/Продукт:БИТ:Учет\_нагрузки\_преподавателей. (Дата обращения: 13.05.2022).
7. СГУ-Инфоком. Продукты. [Электронный ресурс]. – URL: https://sgu-infocom.ru/product/. (Дата обращения: 13.05.2022).
8. Экспертный центр корпорации «Галактика». «Галактика Управление вузом» [Электронный ресурс]. – URL: https://galaktika-it.ru/spb/guv. (Дата обращения: 13.05.2022).
9. Каюгина С.М. Автоматизированная система расчета и распределения учебной нагрузки преподавателей кафедры вуза на базе платформы «1С» [Электронный ресурс]. – URL: https://web.snauka.ru/issues/2015/11/58943. (Дата обращения: 13.05.2022).
10. Михайлов, Ю. И. Моделирование бизнес-процессов: учебно-методическое пособие / Ю. И. Михайлов, В. П. Семенов; Санкт-Петербургский гос. электротехнический ун-т «ЛЭТИ» им. В. И. Ульянова (Ленина). - Санкт-Петербург: ЛЭТИ, 2018. - 39 с.
11. Ларман, К. Применение UML 2.0 и шаблонов проектирование / К. Ларман, - М.: Вильямс, 2019 - 736 с.
12. docs.microsoft.com. Документация по C# [Электронный ресурс]. – URL: https://docs.microsoft.com/ru-ru/dotnet/csharp (дата обращения: 31.05.2022).
13. vscode.ru. Программирование на C, C# и Java [Электронный ресурс]. – URL: https://vscode.ru/articles/tehnologiya-razrabotki-po.html (дата обращения: 13.05.2021).
14. Официальный сайт Visual Studio. [Электронный ресурс]. – URL: https://visualstudio.microsoft.com/ru/. (дата обращения: 04.06.2022).
15. Официальный сайт SQLite. [Электронный ресурс]. – URL: https://www.sqlite.org (дата обращения: 04.06.2022).

ПРИЛОЖЕНИЕ А

**Шаблоны документов**



Рисунок А.1 – Сводка о выполнении учебной нагрузки (шаблон)

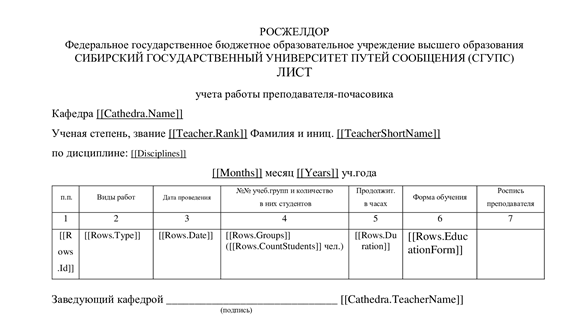


Рисунок А.2 – Лист учета работы преподавателя-почасовика (шаблон)

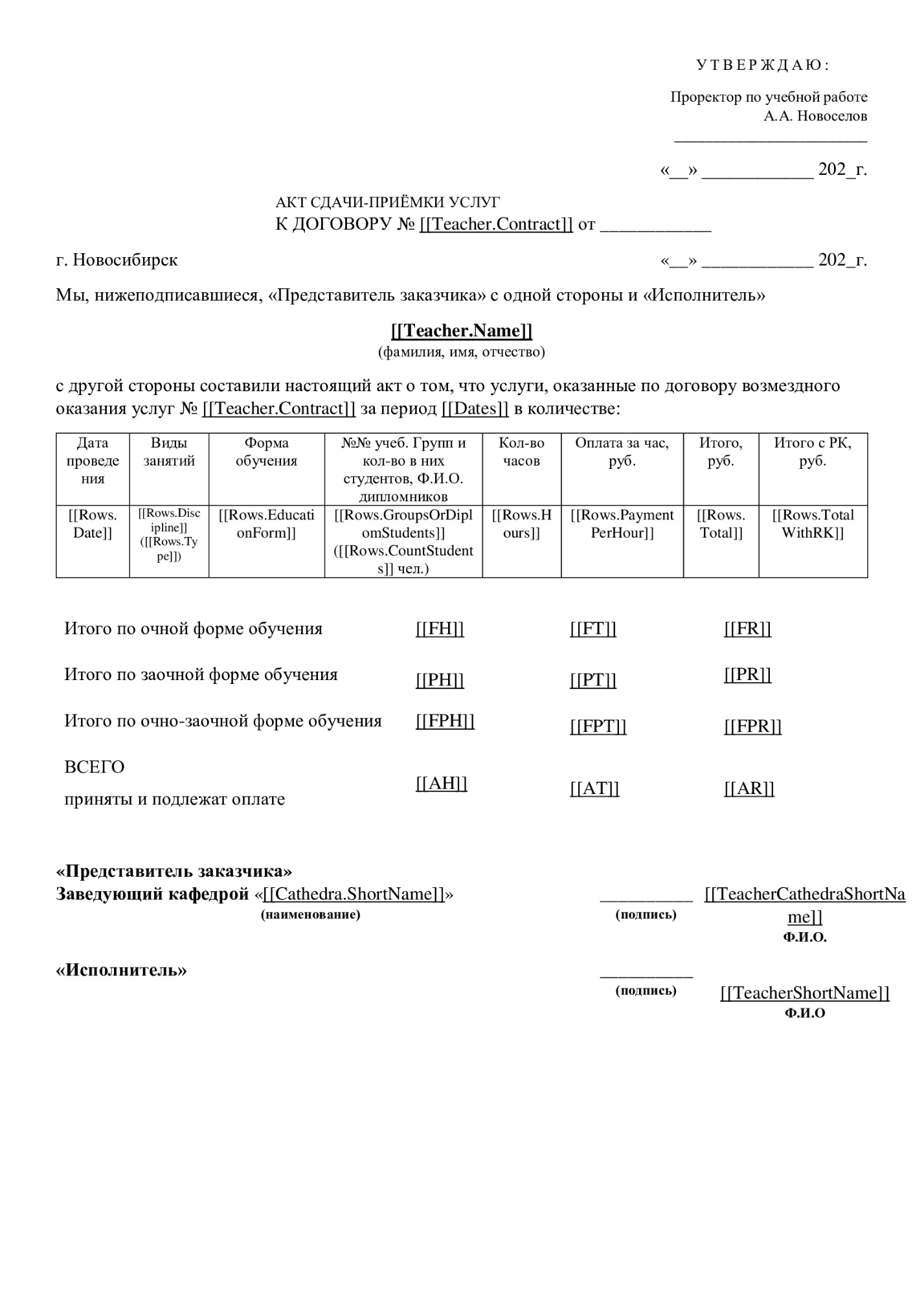


Рисунок А.3 – Акт сдачи-приемки услуг преподавателя (шаблон)