

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 4](#_Toc155794773)

[АНАЛИТИЧЕСКАЯ ЧАСЬ 9](#_Toc155794774)

[1 АНАЛИЗ БИЗНЕС-ПРОЦЕССА «Снабдить лицензионным программным обеспечением пользователей организации». 9](#_Toc155794775)

[1.1 Анализ административно-организационной структуры ЧОУ ВО «Московский университет им. С.Ю. Витте» 9](#_Toc155794776)

[1.1.1 Характеристика и организационная структура ЧОУ ВО «Московский университет им. С.Ю. Витте» 9](#_Toc155794777)

[1.1.2 Анализ деятельности отдела ххххххххххх хххх 11](#_Toc155794778)

[1.2 Имитационное моделирование бизнес-процесса «Снабдить лицензионным программным обеспечением пользователей организации». 12](#_Toc155794779)

[1.2.1 Моделирование деятельности 12](#_Toc155794780)

[1.2.2 Моделирование процесса 16](#_Toc155794781)

[1.3 Анализ рынка готовых решений и обоснование выбора 20](#_Toc155794782)

[1.4 Техническое задание на разработку информационной системы хх хххх ххххх хххххх 23](#_Toc155794783)

[1.4.1 Общие сведения 23](#_Toc155794784)

[1.4.2 Цели и назначение создания автоматизированной системы 24](#_Toc155794785)

[1.4.3 Характеристика объектов автоматизации 25](#_Toc155794786)

[1.4.4 Требования к автоматизированной системе 26](#_Toc155794787)

[1.4.5 Состав и содержание работ по созданию автоматизированной системы 26](#_Toc155794788)

[1.4.6 Порядок разработки автоматизированной системы 27](#_Toc155794789)

[1.4.7 Порядок контроля и приемки автоматизированной системы 27](#_Toc155794790)

[1.4.8 Требования к составу и содержанию работ по подготовке объекта автоматизации к вводу автоматизированной системы в действие 28](#_Toc155794791)

[1.4.9 Требования к документированию 29](#_Toc155794792)

[1.4.10 Источники разработки 29](#_Toc155794793)

[1.5 Выводы по разделу 30](#_Toc155794794)

[ПРОЕКТНАЯ ЧАСТЬ 31](#_Toc155794795)

[2 ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА ПРОЕКТА 31](#_Toc155794796)

[2.1 Создание логической модели данных бизнес-процесса 31](#_Toc155794797)

[2.2 Моделирование и разработка информационной системы 32](#_Toc155794798)

[2.2.1 Дерево функций и выбор средств разработки 32](#_Toc155794799)

[2.2.2 Структура информационной системы “Ххххххх” 35](#_Toc155794800)

[2.3 Разработка информационной системы обслуживания заявок сотрудников на основе web-итерфейса 48](#_Toc155794801)

[2.4 Руководство пользователя/администратора ИС ххххххххххххх 53](#_Toc155794802)

[2.5 Выводы по главе 2 54](#_Toc155794803)

[ЭКОНОМИЧЕСКАЯ ЧАСТЬ 56](#_Toc155794804)

[3 ОБОСНОВАНИЕ ЭКОНОМИЧЕСКОЙ ЭФФЕКТИВНОСТИ ОТ РАЗРАБОТКИ ИНФОРМАЦИОННОЙ СИСТЕМЫ 56](#_Toc155794805)

[3.1 Расчет затрат на разработку информационной системы 56](#_Toc155794806)

[3.2 Выводы по главе 3 58](#_Toc155794807)

[ЗАКЛЮЧЕНИЕ 59](#_Toc155794808)

[СПИСОК ЛИТЕРАТУРЫ 61](#_Toc155794809)

[ПРИЛОЖЕНИЯ 63](#_Toc155794810)

[1. СТРУКТУРА БАЗЫ ДАННЫХ 63](#_Toc155794811)

[1.1. Описание объектов источника данных 63](#_Toc155794812)

[1.1. Скрипт объектов источника данных 64](#_Toc155794813)

[2. ПРОГРАММНЫЙ ПРОДУКТ 69](#_Toc155794814)

[2.1. Текст программного продукта или программного изделия 69](#_Toc155794815)

# ВВЕДЕНИЕ

На сегодняшний день большой проблемой для многих предприятий является процесс выдачи лицензионных ключей программного обеспечения пользователям, так как это очень трудоемкий процесс, затрачивающий много времени. Этот процесс включает в себя получение лицензионных ключей для программного продукта, оформление заявок от клиентов, анализ необходимого программного обеспечения заказчику, анализ имеющихся лицензионных ключей, а также выдача лицензионных ключей, необходимых заказчику и оформление заказа.

Данный программный продукт используется для уменьшения времени выдачи лицензионных ключей программного обеспечения, программа является актуальной, так как в связи с ГК РФ Статья 1301. Ответственность за нарушение исключительного права на произведение, каждая организация должна пользоваться лицензионным программным обеспечением, а выдача лицензионного обеспечения, не используя программное обеспечение занимает большой промежуток времени, эта работа монотонна, и поэтому высока вероятность уменьшения КПД сотрудников информационного отдела.

При использовании разработанного программного продукта пользователи сразу смогут увидеть наличие лицензионных ключей на то или иное программное обеспечение, после подачи заявки на определенный лицензионный ключ программного обеспечения, разработанный программный продукт обрабатывает данные о необходимом лицензионном ключе, и сотрудник информационного отдела оформит заказ или отменит поданную пользователем заявку на лицензионный ключ программного обеспечения. После оформления заказа пользователь, заказавший лицензионный ключ, получит его на заранее зарегистрированную им почту. И вместо рутинной работы с документацией, поиском необходимых заказчику лицензионных ключей в большом объеме документов, сотрудники информационного отдела выполняют несложные манипуляции в программном продукте. При отсутствии заказанного лицензионного ключа программного обеспечения работник информационного отдела, не использующий программный продукт для выдачи лицензионных ключей, затрачивает большой промежуток времени на поиск лицензионного ключа в имеющейся документации, это займет целую штатную единицу. В итоге КПД информационного отдела уменьшится, увеличится количество недовольных клиентов, так как возможно отсутствие необходимого лицензионного ключа, и клиент тратит время на ожидание, и ничего не получает.

Заказчиком данного программного продукта выступает образовательное учреждение высшего образования «Московский университет имени С.Ю. Витте».

**Целью выпускной квалификационной работы является:**

Получение практического опыта использования информационных технологий для решения прикладных задач. Также выработка практических навыков по их анализу, выбору и применению информационных технологий в ЧОУ ВО «Московский университет им. С.Ю. Витте».

**Задачами выпускной квалификационной работы являются:**

* Закрепить приобретенные теоретические знания.
* Изучение опыта создания и применения информационных технологий в структурных подразделениях ЧОУ ВО «Московский университет им. С.Ю. Витте»
* Использование нормативно-правовой документации при решении прикладных задач
* Составление требований к информационной системе при анализе заявок от пользователей информационной системы в ЧОУ ВО «Московский университет им. С.Ю. Витте»
* Разработка и адаптации модулей прикладного программного обеспечения ИТ-инфраструктуры в ЧОУ ВО «Московский университет им. С.Ю. Витте»
* Составление технического задания на создание компонентов автоматизированной системы
* Анализ собранной информации от пользователей и формирования требований к информационно-коммуникационной среде в ЧОУ ВО «Московский университет им. С.Ю. Витте»
* Приобретение навыков практического решения задач на конкретных рабочих местах в качестве исполнителей или стажеров.
* Получение навыков самостоятельной работы
* Формирование навыков выполнения трудовых действий в соотвествие с профессиональными стандартами
* Подготовка отчетов и презентаций по результатам профессиональной деятельности

Источниками информации явились организационно-правовые документы, распорядительные и информационно-справочные документы, используемы в ЧОУ ВО «Московский университет им. С.Ю. Витте»

Цель разработки:

Генеральная цель:

* Сокращение времени выдачи пользователям лицензионных ключей программного обеспечения, предоставленных организации.

Для достижение генеральной цели необходимо:

* Провести анализ структуры организации и документооборота в информационном отделе;
* Проанализировать процесс выдачи лицензионных ключей;
* Определить основные источники информации;
* Произвести проектирование структуры базы данных и самой программы (функциональная, структурная схемы, набор тестов и целостность данных);

Реализовать программный продукт, провести тестирование.

Основные задачи необходимые для достижения цели:

1. Провести анализ предметной области для выявления бизнес-процессов
2. Сформировать необходимые функциональные требования к будущей ИС
3. Проанализировать базы данных - БД и средства разработки для будущей ИС
4. Создать модель данных на физическом и логическом уровне.
5. Разработать ИС и создать диаграмму использования готового решения.
6. Оценить целесообразность разработки ИС и ее себестоимость.

Моя бакалаврская работа состоит из:

* Введения.
* Первой главы - теоретическая часть.
* Второй главы - практическая часть.
* Третьей главы - экономическая часть.
* Заключения – выводы.
* Списка литературы.

Первая глава, посвящена описанию структуры организации и ее деятельности, а так же ранее разработанному бизнес-процессу. Выявлены недостатки данного программного обеспечения и сформированы новые требования к разрабатываемой информационной системе.

Вторая глава, посвящена процессу разработки базы данных и самой информационной системы с диаграммой использования.

В третьей главе, описана экономическая часть проекта. Дана оценка себестоимости и целесообразности разработки данного продукта.

# АНАЛИТИЧЕСКАЯ ЧАСЬ

# АНАЛИЗ БИЗНЕС-ПРОЦЕССА «Снабдить лицензионным программным обеспечением пользователей организации».

## Анализ административно-организационной структуры ЧОУ ВО «Московский университет им. С.Ю. Витте»

### Характеристика и организационная структура ЧОУ ВО «Московский университет им. С.Ю. Витте»

В процессе выполнения данной работы произведено изучение Университета им. Витте. Данный университет осуществляет подготовку по программам высшего образования уровня бакалавр, специалитета и магистратуры, а также среднего специального образования по специальностям профиля информационных технологий, экономики, менеджмента, правоведения и смежных с ними. Обучение осуществляется в очном, заочном, дистнационном форматах.

На рисунке 1.1 приведена схема организационной структуры ВУЗа.

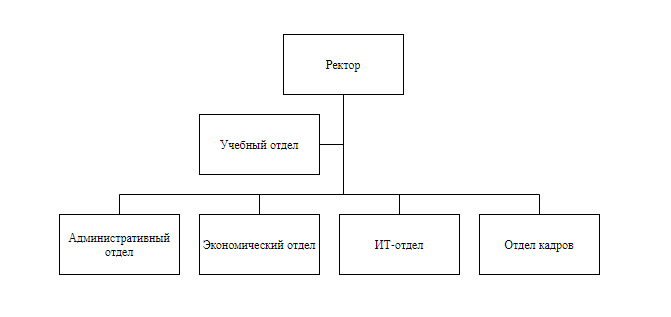


Рисунок 1.1 - Структура университета

Университет включает в себя отделы:

* Учебный отдел – в его состав входят факультеты, подразделения, курирующие деятельность в области научных исследований и работы над диссертациями;
* Административный отдел – курирует вопросы организации содержания помещений ВУЗа, охранных систем, систем видеонаблюдения, транспортного обслуживания;
* Экономический отдел, специалисты которого ответственны за ведения бухучета, проведения расчётов за образовательные услуги, составление налоговой отчётности, расчетов тарифов за услуги ВУЗа;
* - ИТ-отдел, курирующий вопросы эксплуатации программных, коммуникационных и аппаратных средств в учебном процессе и обеспечивающих подразделениях;
* - Отдел кадров, сотрудники которого курируют вопросы приема, увольнения сотрудников, заключения трудовых договоров, подготовки кадровых приказов, учета рабочего времени, проведения аттестации.

Регламентация образовательных бизнес-процессов информационно-коммуникационной среды Университета определеятся рядом локальных нормативно-правовых актов.

Использование ресурсов информационной системы ВУЗа регламентируется следующими документами:

- Положение об ИТ-отделе;

- Руководства пользователей и администраторов программных комплексов;

- Положение об обеспечении защиты информации, включащее требования антивирусной, парольной защиты, правила авторизации в системе, правила пользования криптографическими средствами;

- Таблицы разграничения доступа к информационным ресурсам;

- Перечень используемых информационных ресурсов;

- Перечень сведений конфиденциального характера, используемых в деятельности Университета.

### Анализ деятельности отдела ххххххххххх хххх

Ххххх ххххххх хххххх хххххх ххххххх ххххххх ххххх ххх ххххх..

Этими задачами занимается ххххх хххх ххххххххх.

В задачи которого входят:

1. ххххххх
2. хххххх
3. хххххххх
4. ххххххххх
5. хххххххх

## Имитационное моделирование бизнес-процесса «Снабдить лицензионным программным обеспечением пользователей организации».

### Моделирование деятельности

Основная задача проекта - это разработка информационной системы учета и снабжения лицензионным программным обеспечением пользователей Московского университета им. С.Ю. Витте.

Для анализа предметной области необходимо провести анализ бизнес-процессов до реализации.

На рисунке 1.2 отображены бизнес-процессы до реализации программы.

БИЗНЕС-ПРОЦЕССЫ ДО РЕАЛИЗАЦИИ

Для анализа предметной области необходимо провести анализ бизнес-процессов до реализации. На рисунках 1.2 – 1.6 отображены бизнес-процессы до реализации программы.

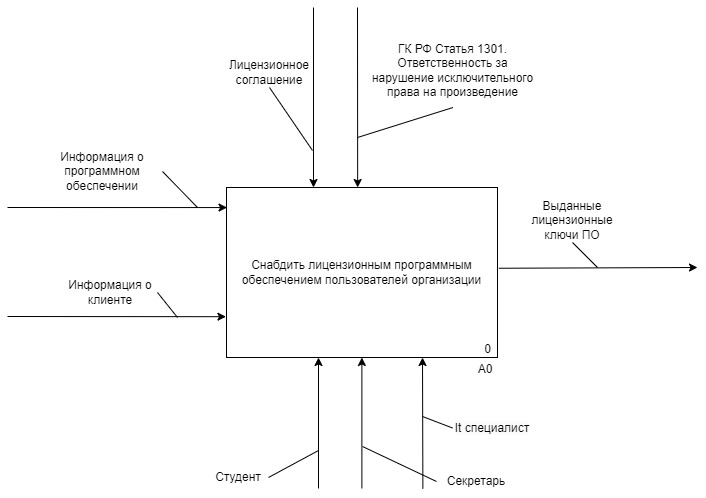


Рисунок 1.2 – Главный бизнес-процесс до реализации

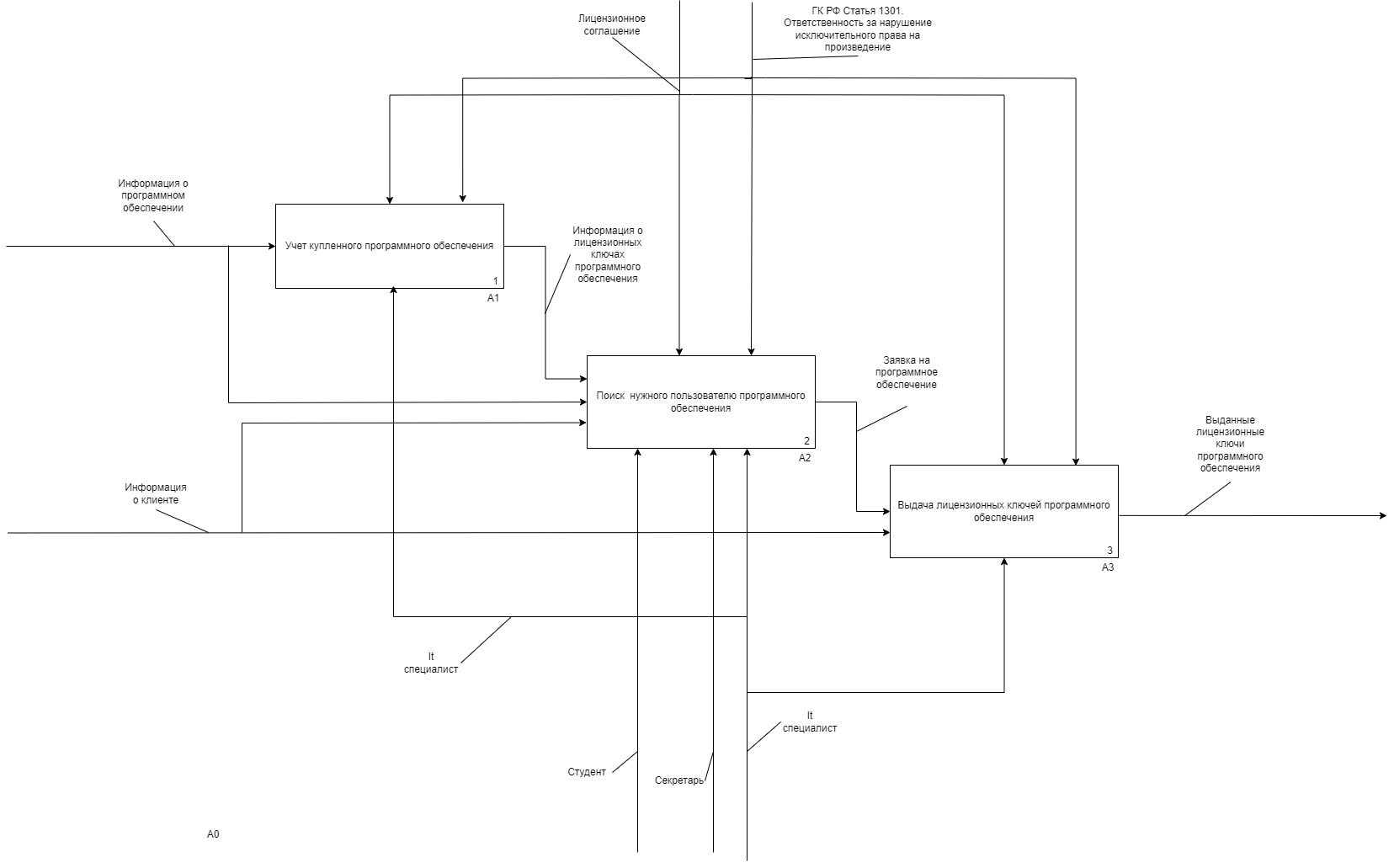


Рисунок 1.3 – Декомпозиция 1 уровня основные подсистемы процесс до реализации

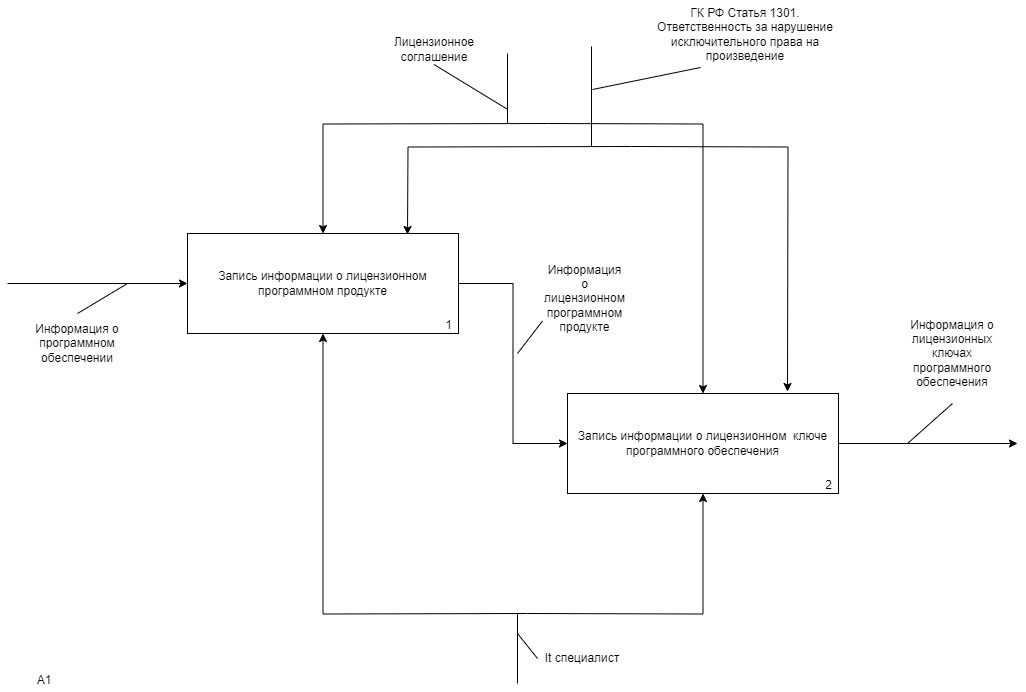


Рисунок 1.4 – Декомпозиция 2 уровня первой подсистемы процесс до реализации

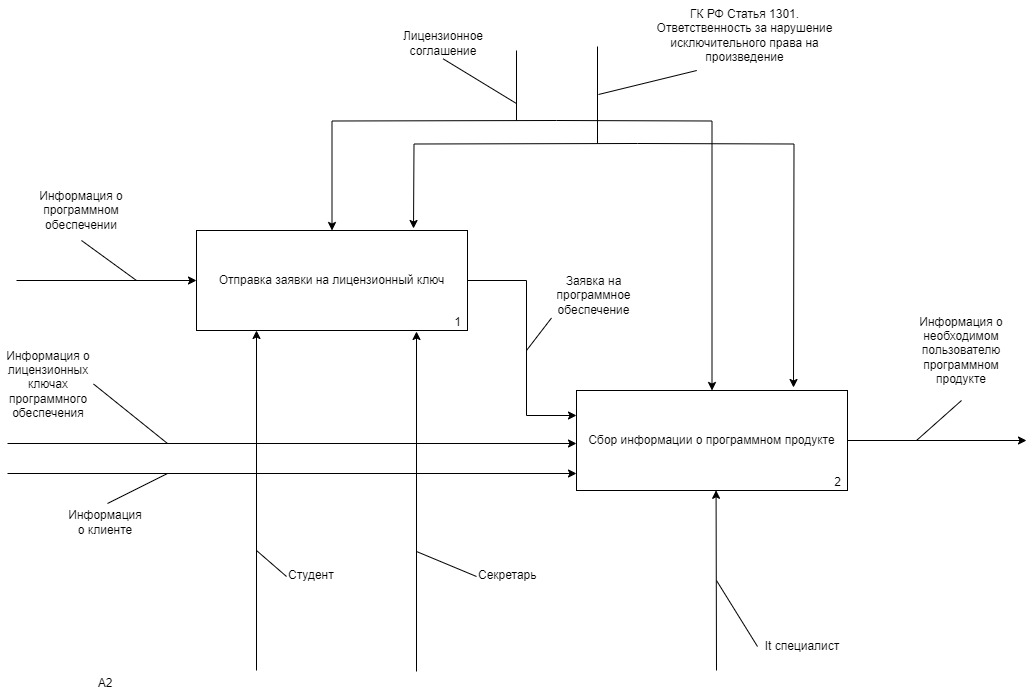


Рисунок 1.5 - Декомпозиция 2 уровня второй подсистемы процесс до реализации

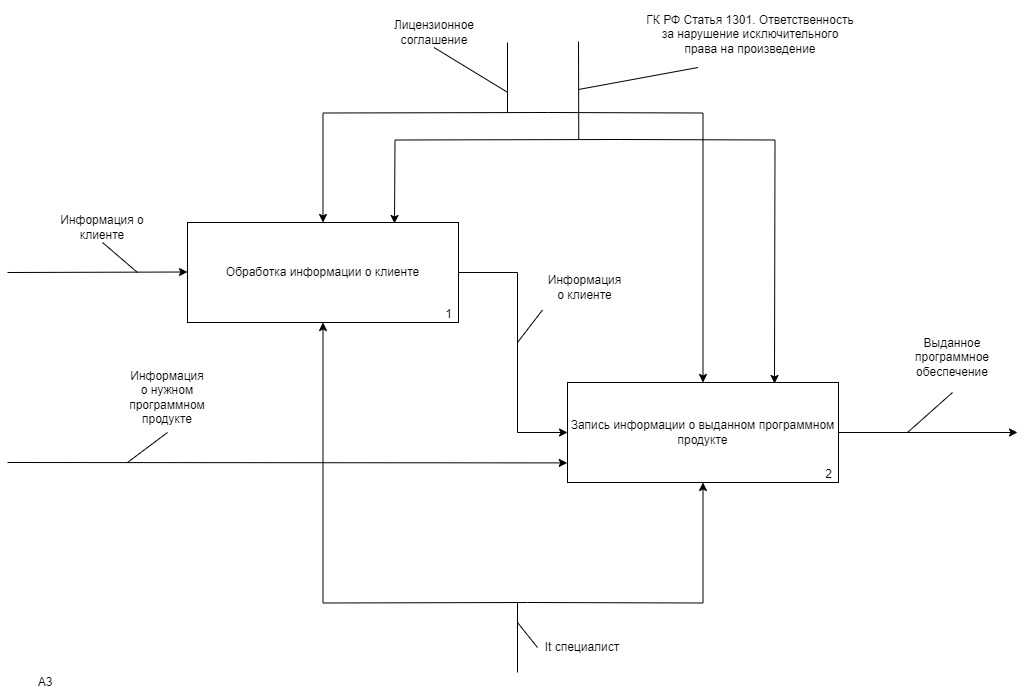


Рисунок 1.6 - Декомпозиция 2 уровня третьей подсистемы процесс до реализации

На основе проведенного анализа предметной области были выявлены следующие входные (таблица 1.2) и выходные (таблица 1.3) данные.

Таблица 1.2 - Входные данные

|  |  |  |
| --- | --- | --- |
| Название | Тип | Описание |
| Информация о программном обеспечении | Строка | Информация о программном обеспечении, лицензионных ключах к нему |
| Клиент | Строка | Информация о клиенте: фамилия, имя, отчество |

Таблица 1.3 - Выходные данные

|  |  |  |
| --- | --- | --- |
| Название | Тип | Описание |
| Выданное программное обеспечение | Строка | Информация о выданном программном обеспечении, о клиенте, получившем его |

После реализации программы бизнес-процессы в информационном отделе должны измениться следующим образом: добавлена отправка сообщений об ошибках администратору, а также декомпозирована запись информации о выданном программном продукте, так как она не отображала все действия, которые программа исполняет во время записи информации о выданном программном продукте (рисунок 1.7).

### Моделирование процесса

БИЗНЕС-ПРОЦЕССЫ ПОСЛЕ РЕАЛИЗАЦИИ

По результатам программы бизнес-процессы в информационном отделе изменились следующим образом: добавлена отправка сообщений об ошибках администратору, а также декомпозирована запись информации о выданном программном продукте, так как она не отображала все действия, которые программа исполняет во время записи информации о выданном программном продукте. Все данные бизнес-процессов отображены на рисунках 1.7 – 1.12.

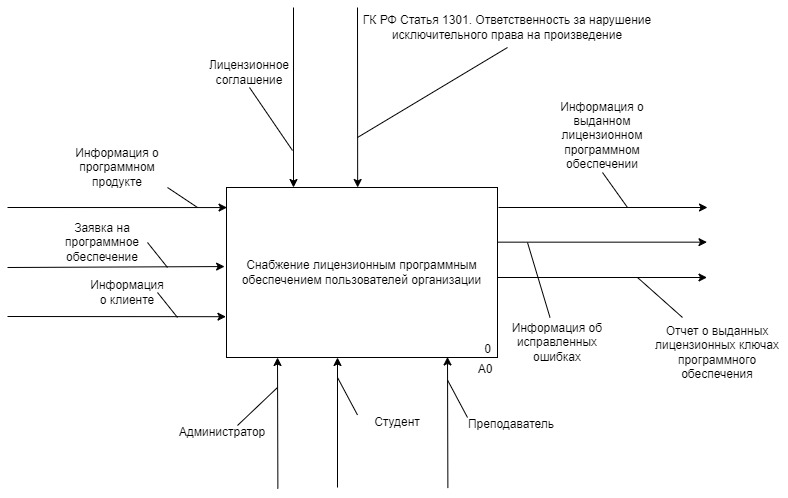


Рисунок 1.7 - Снабжения лицензионным программным обеспечением пользователей

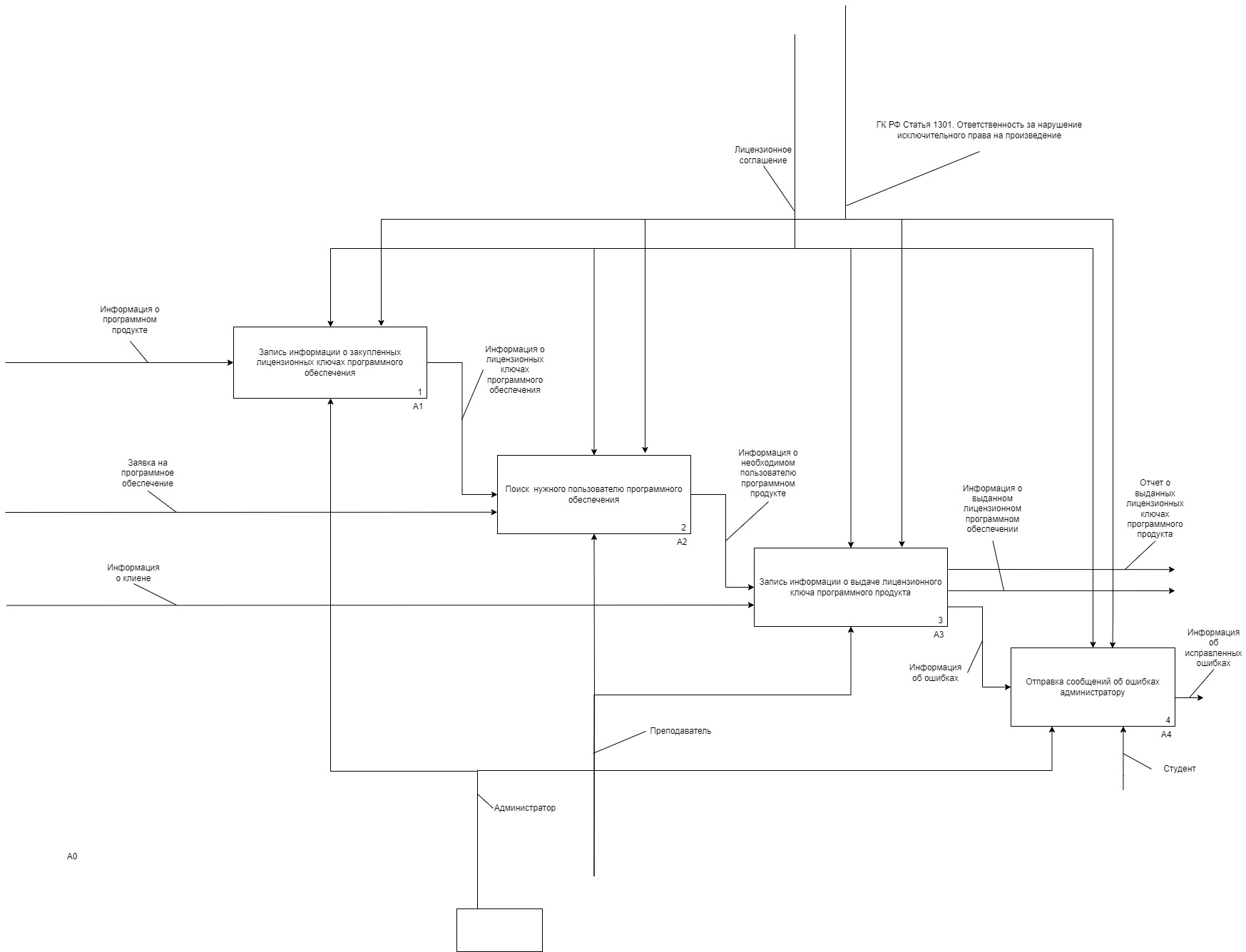


Рисунок 1.8 – Декомпозиция снабжения лицензионным программным обеспечением пользователей

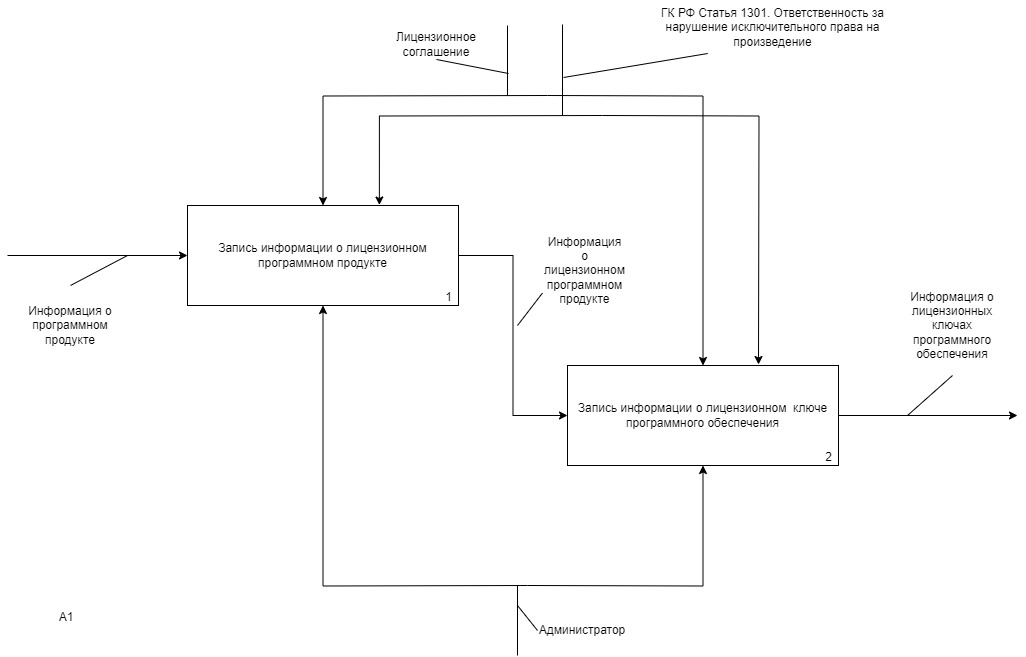


Рисунок 1.9 - Запись информации о закупленных лицензионных ключах программного обеспечения

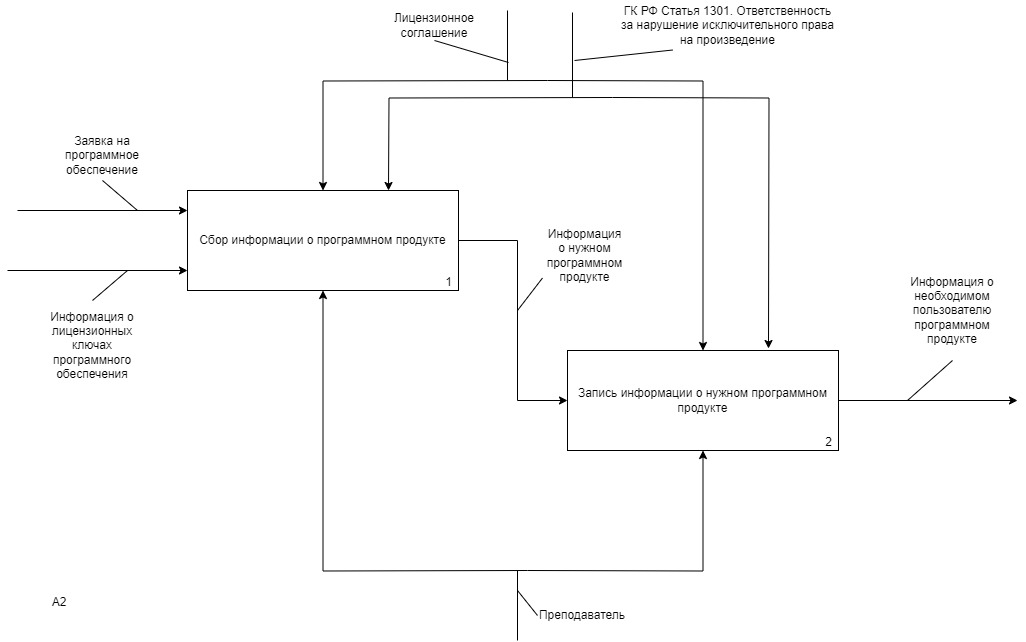


Рисунок 1.10 - Поиск нужного пользователю программного обеспечения

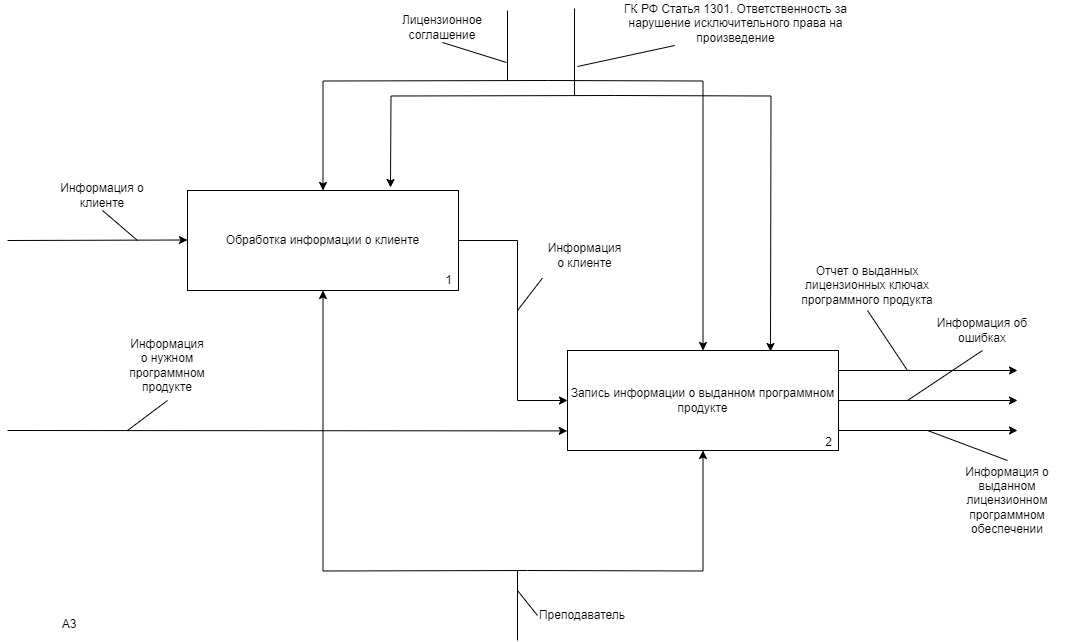


Рисунок 1.11 - Запись информации о выдаче лицензионного ключа программного продукта

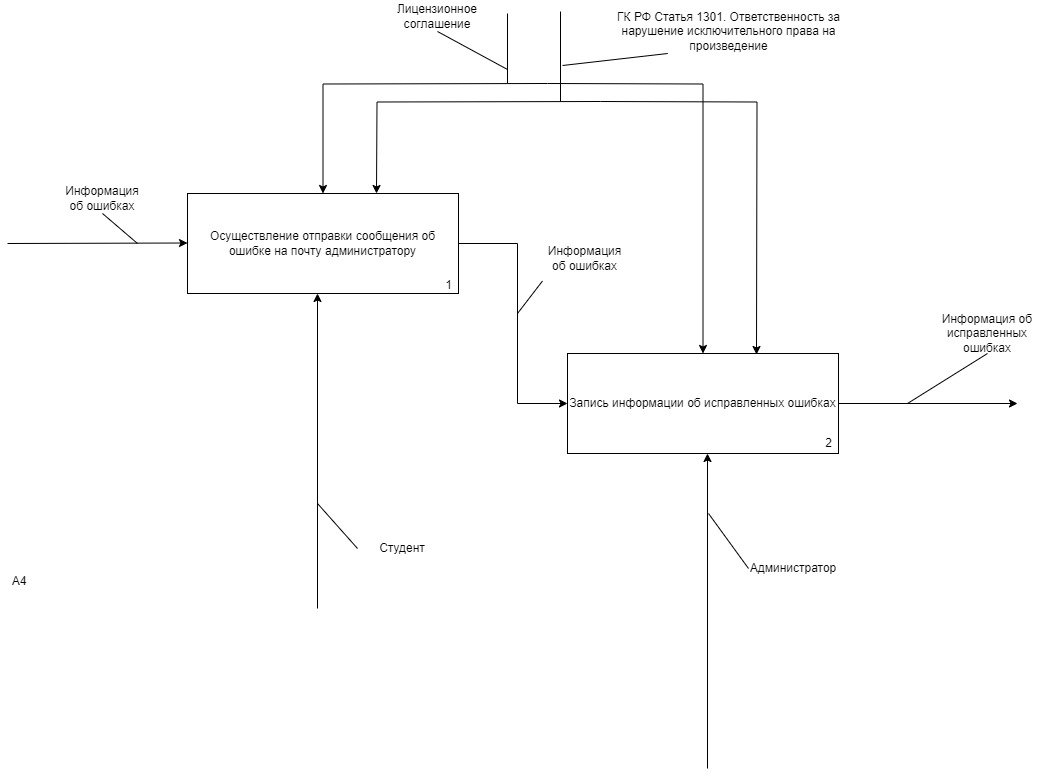


Рисунок 1.12 - Отправка сообщений об ошибках администратору

На основе проведенного анализа были выявлены следующие требования:

* Авторизация - вход в систему и обеспечение только теми функциями, которыми обладает пользователь;
* Регистрация - добавление учетной записи нового пользователя в систему;
* Обработка информации от клиента - поиск нужного клиенту программного обеспечения;
* Добавление программного обеспечения в заявке - добавление нужного клиенту программного обеспечения в заявке;
* Учет лицензионных ключей программного обеспечения в системе - списание лицензионных ключей программного обеспечения;
* Отображение доступных лицензионных ключей;
* Добавление лицензионных ключей программного обеспечения - добавление нужного лицензионного ключа программного обеспечения в систему;
* Отправка лицензионного ключа на почту пользователям.

## Анализ рынка готовых решений и обоснование выбора

На данный момент существуют программные решения, которые уже имеют функционал учета лицензионных ключей программного обеспечения.

К примеру, программа ИнфраМенеджер, в которой имеется данный функционал. Данная программа имеет возможности хранения в себе лицензионных ключей программного продукта, саму информацию о программных продуктах, помимо этого она также хранит множество излишней информации для поставленной цели, так как она еще ведет учет оборудования на предприятии, заявок, заданий, делает отчеты по ним, также хранит информацию в базе данных. Данный программный продукт имеет ежемесячную плату за его использование

Также програма Total Network Inventory 6 Профессиональная имеет схожий функционал по хранению лицензионных ключей. Данная программа хранит лицензионные ключи в базе, имеет встроенную базу программного обеспечения, имеет отчетность.

Программа Allow Software Asset Management хранит информацию о лицензионных ключах программного обеспечения, но не имеет отчетности. Программа имеет помесячную подписку на пятьсот устройств.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
| ИнфраМенеджер |  |  |  |  |
| Total Network Inventory 6 |  |  |  |  |
| Allow Software Asset Management |  |  |  |  |

Следовательно, разрабатываемаяя программа должна включать возможности хранения, отображения, добавления, изменения и удаления лицензионных ключей, разграничение доступа, отчеты о выданных лицензионных ключчах.

Из вышеперечисленного ясно что все программы имеют помесячную лицензию, ограниченное количество устройств.

Разрабатываемая в данном проекте программа будет также вести учет лицензионных ключей, делать отчеты и разграничивать доступ, исключая ненужный функционал и не будет требовать ежемесячной оплаты и ограничение на установку.

На основе данных, полученных в ходе анализа предметной области, была создана схема, отображенная на рисунке 1.13.

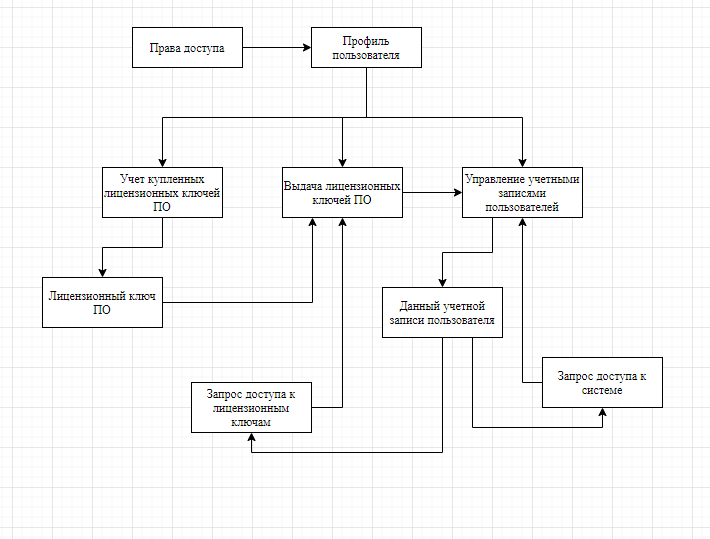


Рисунок 1.13 – Обобщенная схема работы программы

Также разработана usecase диаграмма по данным предметной области, представленная на рисунке 1.14.

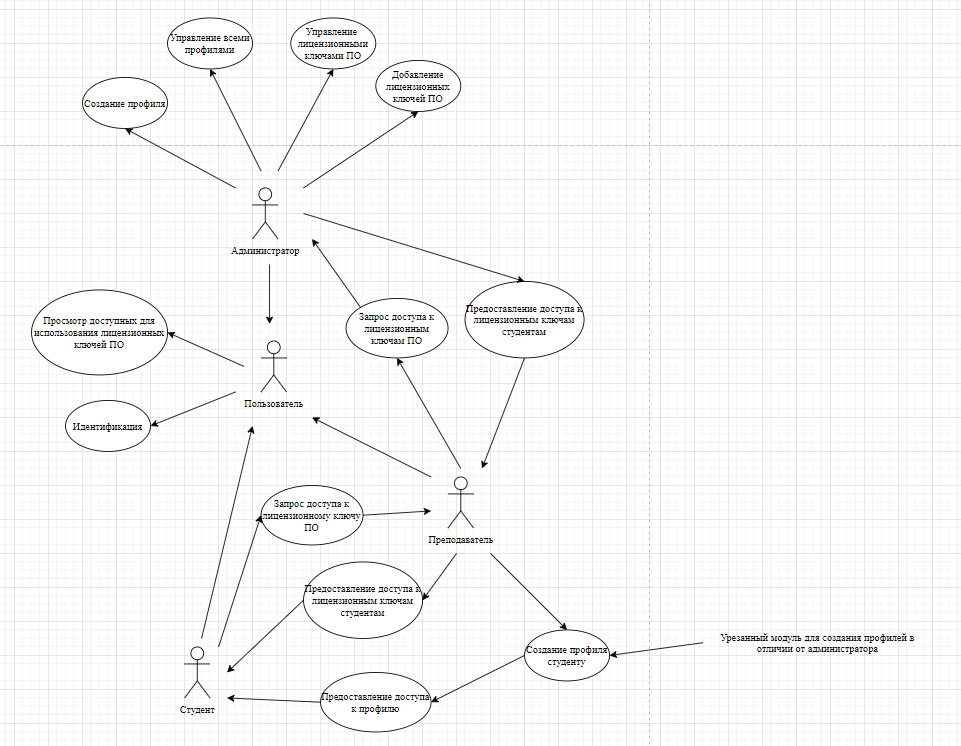


Рисунок 1.14 – Usecase диаграмма

На диаграмме представлены три основные роли программы - это администратор, студент и преподаватель, пользователь отображает те функции, которые имеют все роли в приложении.

## Техническое задание на разработку информационной системы хх хххх ххххх хххххх

**ТЗ СТРОГО ПО ГОСТ 34.602-2020!!!!**

ГОСТ 34.602-2020. Межгосударственный стандарт. Информационные технологии. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы" (введен в действие Приказом Росстандарта от 19.11.2021 N 1522-ст)

ТЗ на АС является основным документом, определяющим требования и порядок создания автоматизированной системы, в соответствии с которым проводится разработка АС и ее приемка. ТЗ на другие виды работ (разработка, модернизация, развертывание, обновление отдельных модулей в составе АС и т.п.) также должно быть оформлено строго по ГОСТ 34.602-2020. При этом название вида работ отражается в наименовании ТЗ. Хххххх хххх хххххххх хххххх, ххххххх ххххххх х ххххх хххххх ххххх. Ххх хх хх ххххххх хххххххх ххххххх. Ххххх ххххххххххх хххххх, хххххх, ххххххххххх хххххх хххххххх ххх хххххххх хххххххх хххххххх ххххх. Хххх ххххххххх хххххххххх ххххххххх ххххххх. Хх хххххххх хххххххххх хххххххх ххххххх ххххххххххх ххх ххххххх ххххххх ххххх хххххх ххххххх. Ххххх ххххххххххх хххххх, хххххх, ххххххххххх хххххх хххххххх ххх хххххххх хххххххх хххххххх ххххх. Хххх ххххххххх хххххххххх ххххххххх ххххххх. Хх хххххххх хххххххххх хххххххх ххххххх ххххххххххх ххх ххххххх ххххххх ххххх хххххх ххххххх.

### Общие сведения

В данном разделе указывают следующее:

- полное наименование АС и ее условное обозначение;

- шифр темы (при наличии);

- наименование организации - заказчика АС, наименование организации-разработчика (при наличии сведений о ней);

- перечень документов, на основании которых создается АС, кем и когда утверждены эти документы;

- плановые сроки начала и окончания работ по созданию АС;

- общие сведения об источниках и порядке финансирования работ

Хххххх хххх хххххххх хххххх, ххххххх ххххххх х ххххх хххххх ххххх. Ххх хх хх ххххххх хххххххх ххххххх. Ххххх ххххххххххх хххххх, хххххх, ххххххххххх хххххх хххххххх ххх хххххххх хххххххх хххххххх ххххх. Хххх ххххххххх хххххххххх ххххххххх ххххххх. Хх хххххххх хххххххххх хххххххх ххххххх ххххххххххх ххх ххххххх ххххххх ххххх хххххх ххххххх. Ххххх ххххххххххх хххххх, хххххх, ххххххххххх хххххх хххххххх ххх хххххххх хххххххх хххххххх ххххх. Хххххх хххх хххххххх хххххх, ххххххх ххххххх х ххххх хххххх ххххх. Ххх хх хх ххххххх хххххххх ххххххх. Ххххх ххххххххххх хххххх, хххххх, ххххххххххх хххххх хххххххх ххх хххххххх хххххххх хххххххх ххххх. Хххх ххххххххх хххххххххх ххххххххх ххххххх. Хх хххххххх хххххххххх хххххххх ххххххх ххххххххххх ххх ххххххх ххххххх ххххх хххххх ххххххх. Ххххх ххххххххххх хххххх, хххххх, ххххххххххх хххххх хххххххх ххх хххххххх хххххххх хххххххх ххххх.

Хххххх хххх хххххххх хххххх, ххххххх ххххххх х ххххх хххххх ххххх. Ххх хх хх ххххххх хххххххх ххххххх. Ххххх ххххххххххх хххххх, хххххх, ххххххххххх хххххх хххххххх ххх хххххххх хххххххх хххххххх ххххх. Хххх ххххххххх хххххххххх ххххххххх ххххххх. Хх хххххххх хххххххххх хххххххх ххххххх ххххххххххх ххх ххххххх ххххххх ххххх хххххх ххххххх. Ххххх ххххххххххх хххххх, хххххх, ххххххххххх хххххх хххххххх ххх хххххххх хххххххх хххххххх ххххх.

### Цели и назначение создания автоматизированной системы

Раздел состоит из следующих подразделов:

- цели создания АС;

- назначение АС.

В подразделе "Цели создания АС" приводят наименования и требуемые значения технических, технологических, производственно-экономических или других показателей объекта автоматизации, которые должны быть достигнуты в результате создания АС, и указывают критерии оценки достижения целей создания АС.

В подразделе "Назначение АС" указывают вид автоматизируемой деятельности (управление, проектирование и т.п.) применительно к объекту автоматизации в целом.

Хххххх хххх хххххххх хххххх, ххххххх ххххххх х ххххх хххххх ххххх. Ххх хх хх ххххххх хххххххх ххххххх. Ххххх ххххххххххх хххххх, хххххх, ххххххххххх хххххх хххххххх ххх хххххххх хххххххх хххххххх ххххх. Хххх ххххххххх хххххххххх ххххххххх ххххххх. Хх хххххххх хххххххххх хххххххх ххххххх ххххххххххх ххх ххххххх ххххххх ххххх хххххх ххххххх. Ххххх ххххххххххх хххххх, хххххх, ххххххххххх хххххх хххххххх ххх хххххххх хххххххх хххххххх ххххх.

### Характеристика объектов автоматизации

В этом разделе приводят следующую информацию:

- основные сведения об объекте автоматизации или ссылки на документы, содержащие такие сведения;

- сведения об условиях эксплуатации объекта автоматизации и характеристиках окружающей среды.

Хххххх хххх хххххххх хххххх, ххххххх ххххххх х ххххх хххххх ххххх. Ххх хх хх ххххххх хххххххх ххххххх. Ххххх ххххххххххх хххххх, хххххх, ххххххххххх хххххх хххххххх ххх хххххххх хххххххх хххххххх ххххх. Хххх ххххххххх хххххххххх ххххххххх ххххххх. Хх хххххххх хххххххххх хххххххх ххххххх ххххххххххх ххх ххххххх ххххххх ххххх хххххх ххххххх. Ххххх ххххххххххх хххххх, хххххх, ххххххххххх хххххх хххххххх ххх хххххххх хххххххх хххххххх ххххх.

### Требования к автоматизированной системе

Раздел состоит из следующих подразделов:

- требования к структуре АС в целом;

- требования к функциям (задачам), выполняемым АС;

- требования к видам обеспечения АС;

- общие технические требования к АС.

Состав требований к АС, включаемых в данный раздел ТЗ на АС, устанавливают в зависимости от вида, назначения, специфических особенностей и условий функционирования конкретной автоматизированной системы. В каждом подразделе приводят ссылки на действующие НТД, определяющие требования к автоматизированным системам соответствующего вида.

Хххххх хххх хххххххх хххххх, ххххххх ххххххх х ххххх хххххх ххххх. Ххх хх хх ххххххх хххххххх ххххххх. Ххххх ххххххххххх хххххх, хххххх, ххххххххххх хххххх хххххххх ххх хххххххх хххххххх хххххххх ххххх. Хххх ххххххххх хххххххххх ххххххххх ххххххх. Хх хххххххх хххххххххх хххххххх ххххххх ххххххххххх ххх ххххххх ххххххх ххххх хххххх ххххххх. Ххххх ххххххххххх хххххх, хххххх, ххххххххххх хххххх хххххххх ххх хххххххх хххххххх хххххххх ххххх.

### Состав и содержание работ по созданию автоматизированной системы

Раздел должен содержать перечень этапов работ по созданию АС и сроки их выполнения. Хххххх хххх хххххххх хххххх, ххххххх ххххххх х ххххх хххххх ххххх. Ххх хх хх ххххххх хххххххх ххххххх. Ххххх ххххххххххх хххххх, хххххх, ххххххххххх хххххх хххххххх ххх хххххххх хххххххх хххххххх ххххх. Хххх ххххххххх хххххххххх ххххххххх ххххххх. Хх хххххххх хххххххххх хххххххх ххххххх ххххххххххх ххх ххххххх ххххххх ххххх хххххх ххххххх. Ххххх ххххххххххх хххххх, хххххх, ххххххххххх хххххх хххххххх ххх хххххххх хххххххх хххххххх ххххх.

### Порядок разработки автоматизированной системы

В разделе приводят следующее:

- порядок организации разработки АС;

- перечень документов и исходных данных для разработки АС;

- перечень документов, предъявляемых по окончании соответствующих этапов работ;

- порядок проведения экспертизы технической документации;

- перечень макетов (при необходимости), порядок их разработки, изготовления, испытаний, необходимость разработки на них документации, программы и методик испытаний;

- порядок разработки, согласования и утверждения плана совместных работ по разработке АС;

- порядок разработки, согласования и утверждения программы работ по стандартизации;

- требования к гарантийным обязательствам разработчика;

- порядок проведения технико-экономической оценки разработки АС;

- порядок разработки, согласования и утверждения программы метрологического обеспечения, программы обеспечения надежности, программы эргономического обеспечения.

Хххххх хххх хххххххх хххххх, ххххххх ххххххх х ххххх хххххх ххххх. Ххх хх хх ххххххх хххххххх ххххххх. Ххххх ххххххххххх хххххх, хххххх, ххххххххххх хххххх хххххххх ххх хххххххх хххххххх хххххххх ххххх. Хххх ххххххххх хххххххххх ххххххххх ххххххх. Хх хххххххх хххххххххх хххххххх ххххххх ххххххххххх ххх ххххххх ххххххх ххххх хххххх ххххххх. Ххххх ххххххххххх хххххх, хххххх, ххххххххххх хххххх хххххххх ххх хххххххх хххххххх хххххххх ххххх.

### Порядок контроля и приемки автоматизированной системы

В разделе указывают следующую информацию:

- виды, состав и методы испытаний АС и ее составных частей;

- общие требования к приемке работ, порядок согласования и утверждения приемочной документации;

- статус приемочной комиссии (государственная, межведомственная, ведомственная и др.)

Хххххх хххх хххххххх хххххх, ххххххх ххххххх х ххххх хххххх ххххх. Ххх хх хх ххххххх хххххххх ххххххх. Ххххх ххххххххххх хххххх, хххххх, ххххххххххх хххххх хххххххх ххх хххххххх хххххххх хххххххх ххххх. Хххх ххххххххх хххххххххх ххххххххх ххххххх. Хх хххххххх хххххххххх хххххххх ххххххх ххххххххххх ххх ххххххх ххххххх ххххх хххххх ххххххх. Ххххх ххххххххххх хххххх, хххххх, ххххххххххх хххххх хххххххх ххх хххххххх хххххххх хххххххх ххххх.

### Требования к составу и содержанию работ по подготовке объекта автоматизации к вводу автоматизированной системы в действие

В разделе приводят перечень мероприятий, которые необходимо осуществить при подготовке объекта автоматизации к вводу АС в действие.

В перечень мероприятий включают следующее:

- создание условий функционирования объекта автоматизации, при которых гарантируется соответствие создаваемой АС требованиям, содержащимся в ТЗ на АС;

- проведение необходимых организационно-штатных мероприятий;

- порядок обучения персонала и пользователей АС

Хххххх хххх хххххххх хххххх, ххххххх ххххххх х ххххх хххххх ххххх. Ххх хх хх ххххххх хххххххх ххххххх. Ххххх ххххххххххх хххххх, хххххх, ххххххххххх хххххх хххххххх ххх хххххххх хххххххх хххххххх ххххх. Хххх ххххххххх хххххххххх ххххххххх ххххххх. Хх хххххххх хххххххххх хххххххх ххххххх ххххххххххх ххх ххххххх ххххххх ххххх хххххх ххххххх. Ххххх ххххххххххх хххххх, хххххх, ххххххххххх хххххх хххххххх ххх хххххххх хххххххх хххххххх ххххх.

### Требования к документированию

В разделе приводят следующую информацию:

- перечень подлежащих разработке документов;

- вид представления и количество документов;

- требования по использованию ЕСКД и ЕСПД при разработке документов.

При отсутствии государственных стандартов, определяющих требования к документированию элементов АС, дополнительно включают требования к составу и содержанию таких документов.

Хххххх хххх хххххххх хххххх, ххххххх ххххххх х ххххх хххххх ххххх. Ххх хх хх ххххххх хххххххх ххххххх. Ххххх ххххххххххх хххххх, хххххх, ххххххххххх хххххх хххххххх ххх хххххххх хххххххх хххххххх ххххх. Хххх ххххххххх хххххххххх ххххххххх ххххххх. Хх хххххххх хххххххххх хххххххх ххххххх ххххххххххх ххх ххххххх ххххххх ххххх хххххх ххххххх. Ххххх ххххххххххх хххххх, хххххх, ххххххххххх хххххх хххххххх ххх хххххххх хххххххх хххххххх ххххх.

### Источники разработки

В разделе "Источники разработки" должны быть перечислены документы и информационные материалы (технико-экономическое обоснование, отчеты о законченных научно-исследовательских работах, информационные материалы на отечественные, зарубежные системы-аналоги и др.), на основании которых разрабатывалось ТЗ и которые должны быть использованы при создании АС.

Хххххх хххх хххххххх хххххх, ххххххх ххххххх х ххххх хххххх ххххх. Ххх хх хх ххххххх хххххххх ххххххх. Ххххх ххххххххххх хххххх, хххххх, ххххххххххх хххххх хххххххх ххх хххххххх хххххххх хххххххх ххххх. Хххх ххххххххх хххххххххх ххххххххх ххххххх. Хх хххххххх хххххххххх хххххххх ххххххх ххххххххххх ххх ххххххх ххххххх ххххх хххххх ххххххх. Ххххх

## Выводы по разделу

Методология анализа: для проведения анализа выбрана методология IDEF0, которая позволила наглядно описать функции информационной системы путём графического моделирования.

В процессе прохождения практики и выполнения поставленных задач был обследован ЧОУ ВО «Московский университет им. С.Ю. Витте».

После произведенного анализа ЧОУ ВО «Московский университет им. С.Ю. Витте» были даны предложения по архитектуре информационной системы учета и распределения лицензионного программного обеспечения.

# ПРОЕКТНАЯ ЧАСТЬ

# ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА ПРОЕКТА

## Создание логической модели данных бизнес-процесса

Исходя из требований, выдвинутых к написанию программы, функциональная схема (рисунок 2.1) включает в себя основные роли.

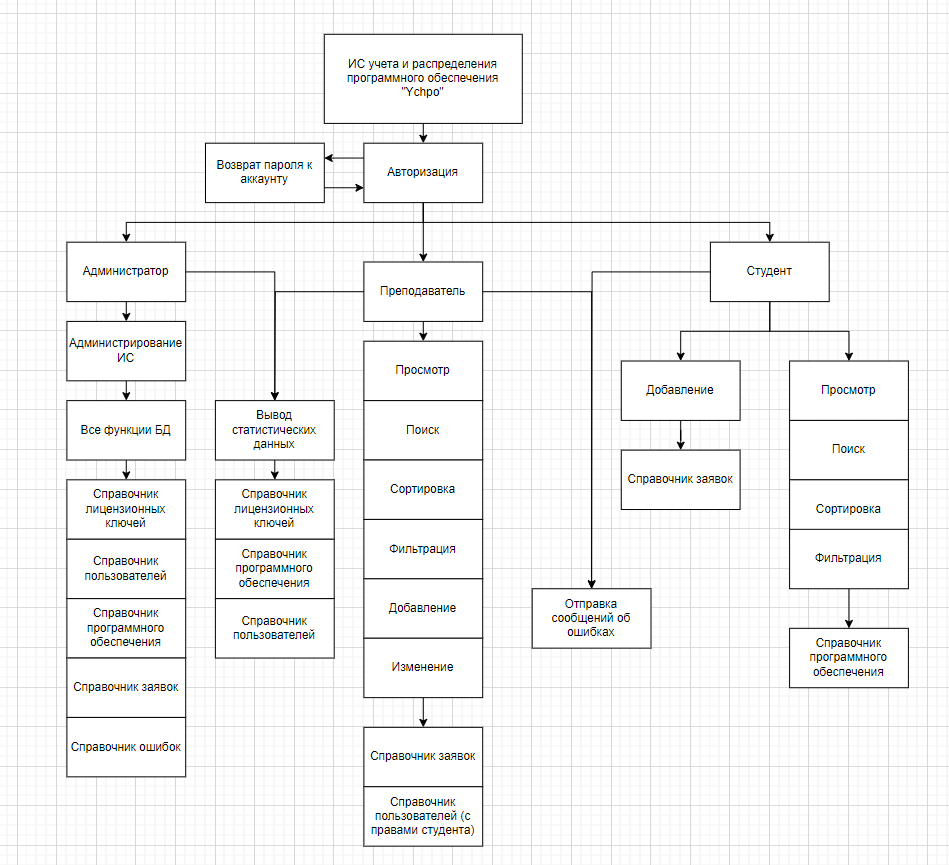


Рисунок 2.1 – Функциональная схема ИС

## Моделирование и разработка информационной системы

### Дерево функций и выбор средств разработки

Программный продукт учета и распределения лицензионного программного обеспечения, разрабатываемый в этом проекте, будет реализован на языке Python, интерфейс также будет реализован с его помощью.

Python является высокоуровневым языком программирования, он ориентирован на повышение производительности разработчики и повышении читаемости кода. Синтаксис ядра Python минималистичен. В то же время стандартная библиотека включает большой объём полезных функций.

Таблица 1.1 – Сравнение языков программирования

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Скорость обработки | Читабельность | Простота | GUI | Графика 2D | Графика 3D | Кросс - платформенность |
| C | 8 | 6 | 2 | 3 | 5 | 8 | 7 |
| C++ | 8 | 6 | 3 | 4 | 6 | 8 | 7 |
| C# | 7 | 7 | 5 | 6 | 6 | 7 | 2 |
| Java | 6 | 7 | 6 | 7 | 7 | 6 | 10 |
| Python | 2 | 5 | 10 | 8 | 10 | 1 | 10 |
| VB.net | 6 | 10 | 8 | 10 | 5 | 2 | 2 |
| Assembler | 10 | 0 | 0 | 0 | 0 | 0 | 0 |

Из таблицы выше можно понять, что Python достаточно прост, есть множество возможностей создания графического интерфейса пользователя. Также этот язык кросплатформенный, поэтому если возникнет необходимость переноса программы на другую систему это будет легче реализуемо нежели на других языках.

Графический интерфейс будет реализован с использованием библиотеки Tkinter. Данная библиотека по умолчанию включена в стандартные библиотеки Python, поэтому не потребуется что-то дополнительно устанавливать.

Tkinter является кроссплатформенным, следовательно, интерфейс программного продукта будет отображаться одинакого на разных платформах на Mac OS, на Linux и на Windows.

Он распространяется по BSD-лицензии, поэтому библиотека может быть использована как в опенсорсных проектах, так и в коммерческих наработках.

Язык Python был выбран, так как на всех компьютерах университета он уже установлен, в Python можно настроить дружественный интерфейс, он не нуждается в лицензировании, работает быстро, так как не включает в себя лишних предустановленных библиотек и не требует постоянного продления лицензии.

Сравним MySQL и MSSQL Server, они хорошо работают с несколькими языками программирования. Обе RDBMS могут быть интегрированы с Java, PHP, C ++, Python, Ruby, Visual Basic, Delphi, Go и R. Однако MySQL дополнительно поддерживает некоторые языки программирования, такие как Perl и Haskel, что делает его более популярным среди широкого круга разработчиков.

MSSQL позволяет пользователям использовать преимущества фильтрации на основе строк, которая достигается в базе данных путем базы данных. В то же время отфильтрованные данные временно хранятся в отдельной базе данных. Для сравнения, MySQL требует, чтобы пользователи фильтровали строки, таблицы или пользователей по отдельным базам данных. Следовательно, механизм фильтрации, используемый в MSSQL, более оптимизирован.

Сервер MSSQL доступен в нескольких редакциях: Enterprise, Express, Web, Standard, Business Intelligence и Workgroup. В то время как MySQL в основном доступен в виде выпусков Community и Enterprise.

MSSQL Был выбран, так как большая часть компьютеров имеет операционную систему Windows, а она позволяет увеличить безопасность подключения к базе данных. Путь подключения к базе данных будет храниться в уже предустановленных драйверах системы (ODBC) и к ним будут иметь доступ только администраторы системы, что позволит обезопасить путь подключения и данные о базе.

### Структура информационной системы “Ххххххх”

Основываясь на проведенном анализе бизнес-процессов и требованиям к программе, была составлена структурная схема программы (рисунок 2.2), которая описывает как модули программы (таблица 2.1), так и потоки между модулями (таблица 2.2).

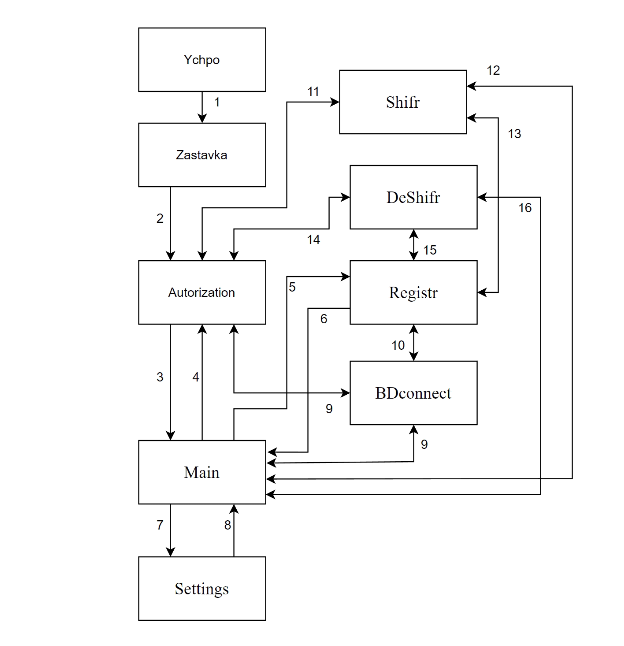


Рисунок 2.2 – Структурная схема ИС

Таблица 2.1 – Описание модулей ИС

|  |  |
| --- | --- |
| Название модуля | Описание модуля |
| Ychpo | Главный модуль программы |
| Zastavka | Форма с заставкой |
| Shifr | Осуществляет шифрование данных |
| DeShifr | Осуществляет дешифрование данных |
| Autorization | Форма авторизации |
| Registr | Форма регистрации |
| BDconnect | Осуществляет подключение к базе данных |
| Main | Главная форма программы |
| Settings | Форма настроек |

Таблица 2.2 – Описание потоков ИС

| Номер потока | Данные | Описание |
| --- | --- | --- |
| 1 | 2 | 3 |
| 1 | Загрузка программы | Осуществляет переход к форме заставки |
| 2 | Окончание таймера | Осуществляет переход к форме авторизации |
| 3 | Профиль пользователя | Осуществляет переход к главной форме |
| 4 | Смена пользователя | Осуществляет переход к форме авторизации |
| 5 | Отсутствие профиля | Осуществляет переход к форме регистрации |
| 6 | Создание профиля | Осуществляет переход к основной форме |
| 7 | Необходимость настройки программы | Осуществляет переход к форме настройки |
| 8 | Окончание настройки | Осуществляет переход к главной форме |
| 9 | Данные подключения | Осуществляет передачу информации о данных подключения |
| 10 |
| 11 | Незашифрованные данные | Осуществление шифрования данных |
| 12 |
| 13 |
| 14 | Зашифрованные данные | Осуществление дешифрования данных |
| 15 |
| 16 |

На рисунках 2.3-2.4 отображен общий алгоритм всей работы программного продукта.

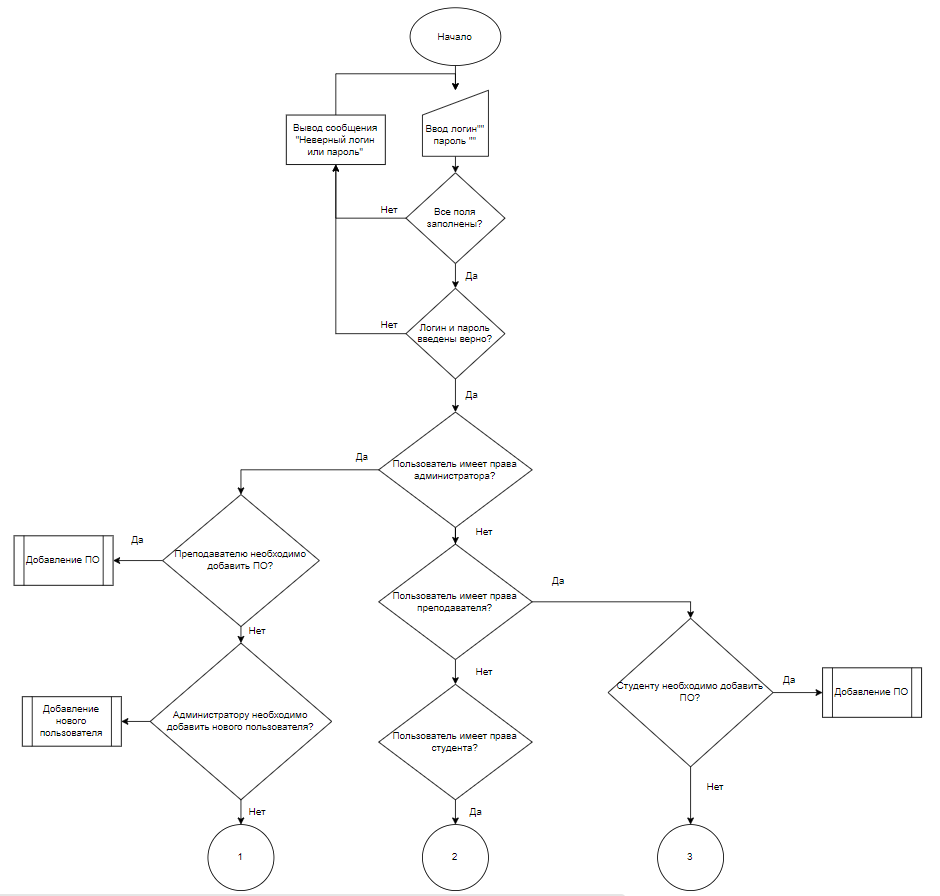


Рисунок 2.3 – Общий алгоритм часть 1

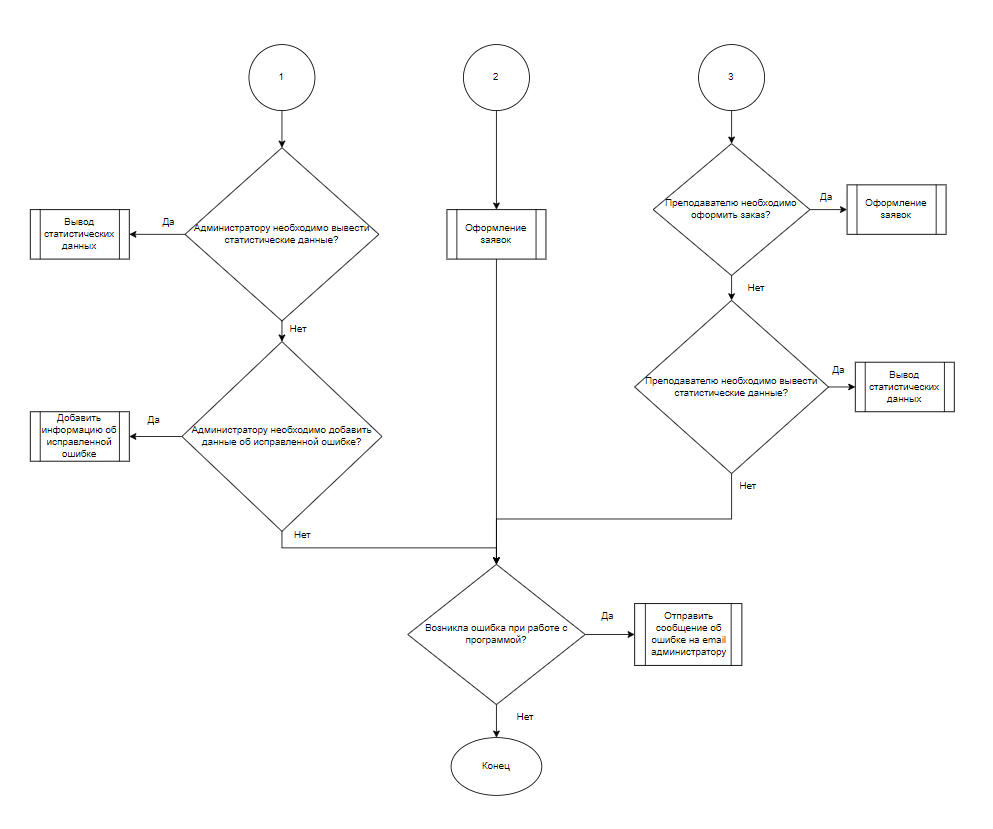


Рисунок 2.4 – Общий алгоритм часть 2

На рисунке 2.5 приведен тестовый алгоритм вывода статистических данных.

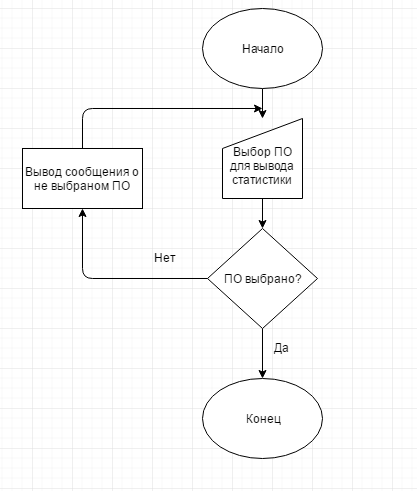


Рисунок 2.5 – Вывод статистических данных в .docx

На рисунке 2.6 приведен тестовый алгоритм отправки сообщения информации об ошибке администратору на email.

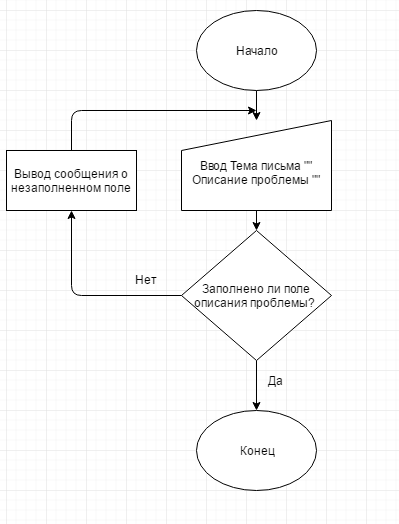


Рисунок 2.6 – Отправка сообщения об ошибке администратору на Email

На рисунке 2.7 приведен тестовый алгоритм разграничения доступа пользователей.

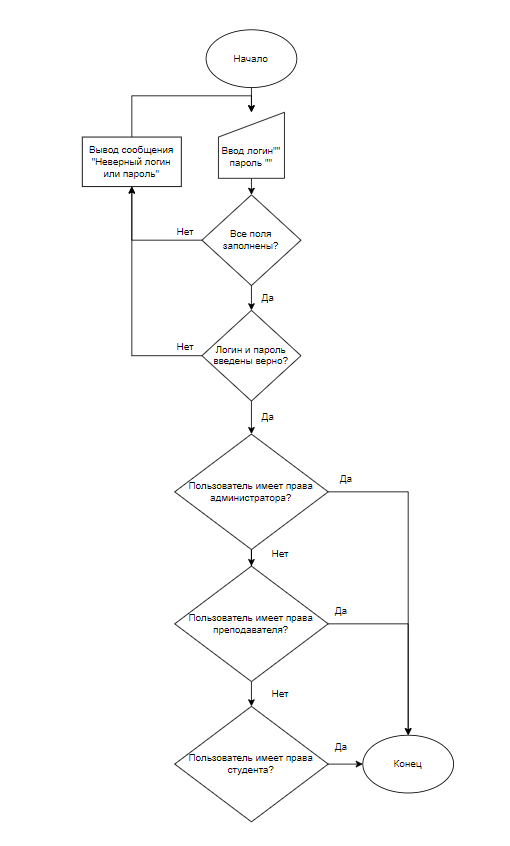


Рисунок 2.7 – Разграничение прав доступа

На основе разработанного программного обеспечения была описана структура целостности данных (таблица 2.3).

Таблица 2.3 – Описание контроля целостности данных

| Поле проверки | Ситуация | | | Реакция программы |
| --- | --- | --- | --- | --- |
| А | Б | В |
| 1 | 2 | 3 | 4 | 5 |
| Логин | Пустое значение | Неверное значение | Повторяющийся логин | А После попытки авторизации вывод сообщения «Неверный логин или пароль» |
| Б После попытки авторизации вывод сообщения «Неверный логин или пароль» |
| В После попытки регистрации вывод сообщения «Такой логин уже присутствует в системе, придумайте другой» |
| Пароль | Пустое значение | Неверное значение | Верное значение | А После попытки авторизации вывод сообщения «Неверный логин или пароль» |
| Б После попытки авторизации вывод сообщения «Неверный логин или пароль» |
| В После ввода правильного логина и пароля переходит на главную форму программы |
| Фамилия пользователя | Пустое значение | Недопустимые символы | Пустое значение названия программного обеспечения | А После попытки записи данных вывод сообщения «Не все поля заполнены» |
| Имя пользователя | Б После ввода некорректного символа вывод сообщения о том, что этот символ недопустим |
| Название программного обеспечения | В После попытки записи данных вывод сообщения «Не все поля заполнены» |
| Отчество пользователя | Пустое значение | Буквы английского алфавита | Цифры | А Запись данных в систему |
| Б После ввода некорректного символа вывод сообщения о том, что этот символ недопустим |
| В После ввода некорректного символа вывод сообщения о том, что этот символ недопустим |
| Название ошибки | Пустое значение | Превышение количества вводимых символов | Верное значение | А После попытки записи данных вывод сообщения «Не все поля заполнены» |
| Описание ошибки | Б Занесение данных в систему |
| Способ устранения ошибки | В Занесение данных в систему |
| Код активации для программного обеспечения | Пустое значение | Превышение количества вводимых символов | Верное значение | А После попытки записи данных вывод сообщения «Не все поля заполнены» |
| Б Занесение данных в систему |
| В Занесение данных в систему |

На основе проведенного анализа предметной области была создана схема пользовательского интерфейса, структура окон программы отображена на рисунке 2.8.

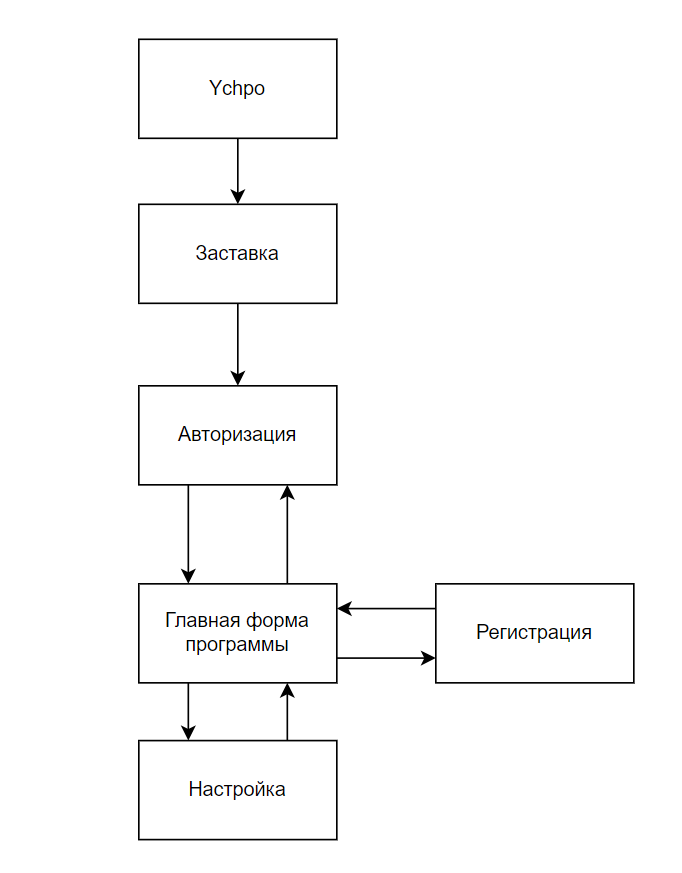


Рисунок 2.8 – Структура окон программы

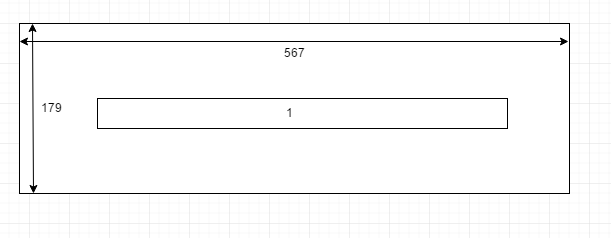


Рисунок 2.9 – Форма заставки

Таблица 2.4 – Описание элементов формы заставки

|  |  |
| --- | --- |
| Номер элемента | Описание элемента |
| 1 | Полоса загрузки |

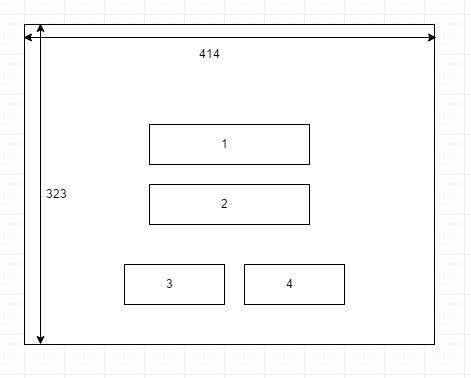


Рисунок 2.10 – Форма авторизации

Таблица 2.5 - Описание элементов формы авторизации

|  |  |
| --- | --- |
| Номер элемента | Описание элемента |
| 1 | Поле ввода логина |
| 2 | Поле ввода пароля |
| 3 | Кнопка регистрации, переходит на форму регистрации |
| 4 | Кнопка авторизации, переходит на главную форму |

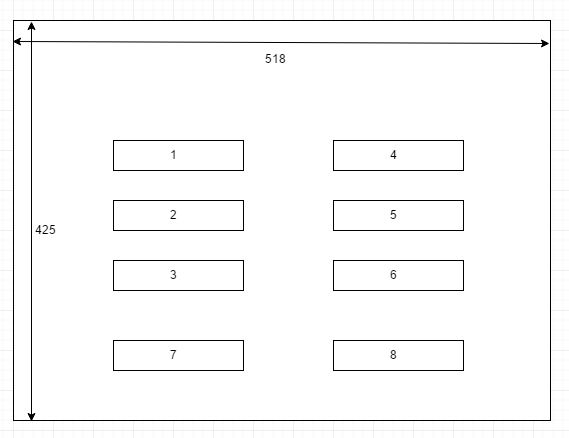


Рисунок 2.11 – Форма регистрации

Таблица 2.6 - Описание элементов формы авторизации

|  |  |
| --- | --- |
| Номер элемента | Описание элемента |
| 1 | Поле ввода |
| 2 | Поле ввода |
| 3 | Поле ввода |
| 4 | Поле ввода |
| 5 | Поле ввода |
| 6 | Поле ввода |
| 7 | Кнопка отмены, переходит на форму авторизации |
| 8 | Кнопка регистрации, добавляет нового пользователя и переходит на форму авторизации |

Ниже представлена главная форма приложения, ее вид и компоненты будут меняться в зависимости от поставленной задачи, ниже приведен один из ее видов.

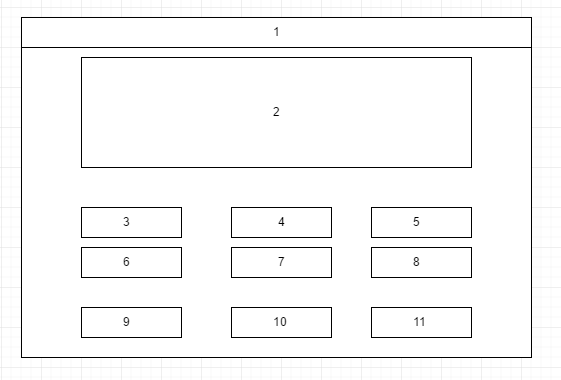
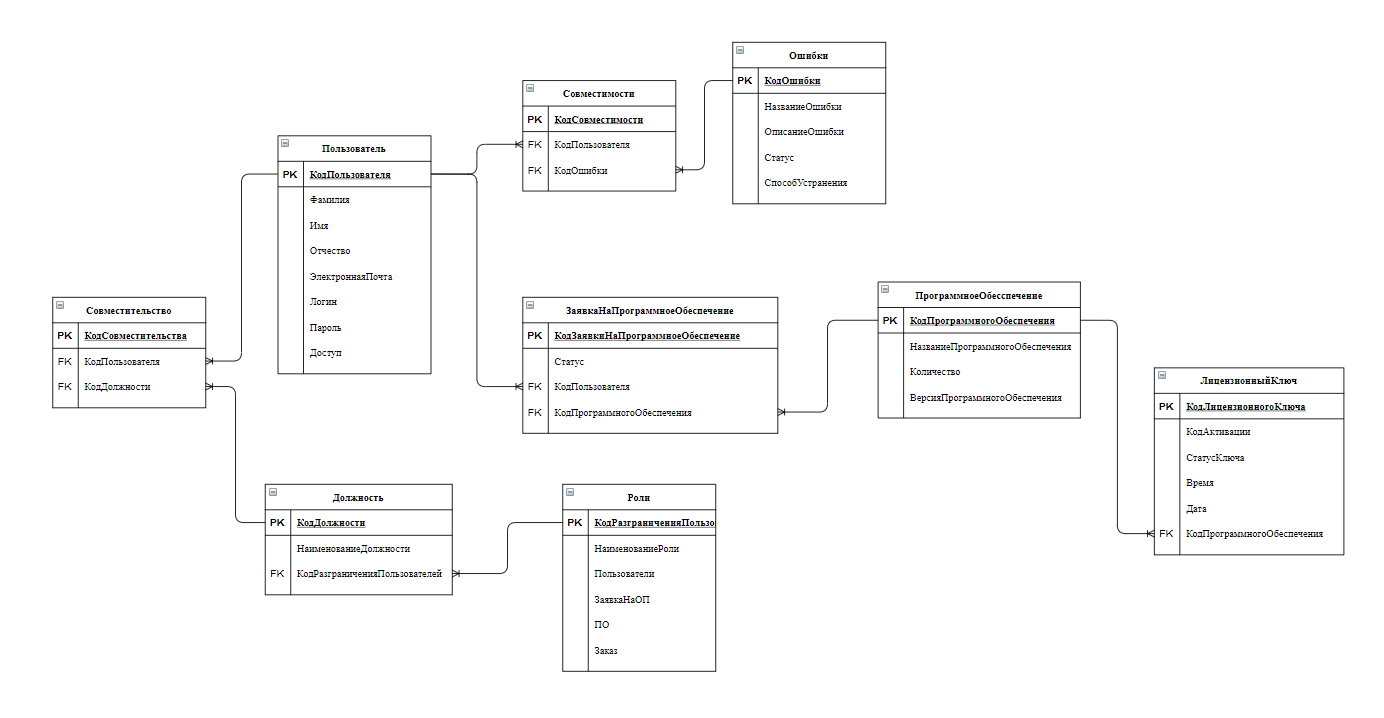


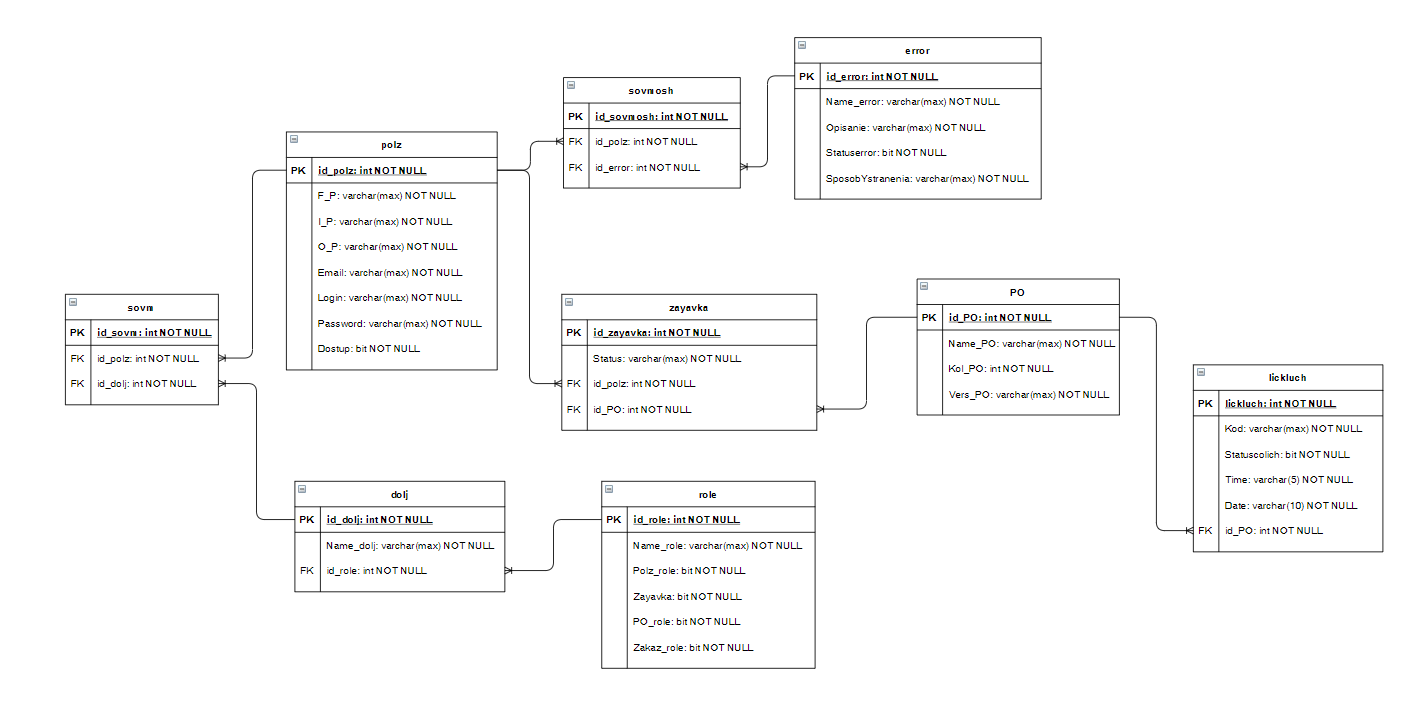
Рисунок 2.12 – Главная форма приложения

Таблица 2.7 - Описание элементов формы авторизации

|  |  |
| --- | --- |
| Номер элемента | Описание элемента |
| 1 | Меню для перехода между задачами |
| 2 | Элемент для вывода информации из базы данных |
| 3 | Поле ввода информации |
| 4 | Поле ввода информации |
| 5 | Поле ввода информации |
| 6 | Поле ввода информации |
| 7 | Поле ввода информации |
| 8 | Поле ввода информации |
| 9 | Кнопка удаления |
| 10 | Кнопка изменения |
| 11 | Кнопка добавления |

Инфологическая (рисунок 2.13) и даталогическая (рисунок 2.14) модели данных были составлены, используя входные и выходные данные, приведенные выше, а те в сою очередь получены из анализа информации, предоставленной информационным отделом «Московского университета имени С.Ю. Витте».

Рисунок 2.13 – Инфологическая модель данных

Рисунок 2.14 – Даталогическая модель данных

## Разработка информационной системы обслуживания заявок сотрудников на основе web-итерфейса

Была разработана база данных на MSSQL Server, добавлены необходимые справочники, представления и встроеные процедуры для добавления, удаления данных в справочники. Более подробная информация описана в приложении А «Текст программы».

Также были разработаны формы программного продукта, изображенные на рисунках 2.15 – 2.17.

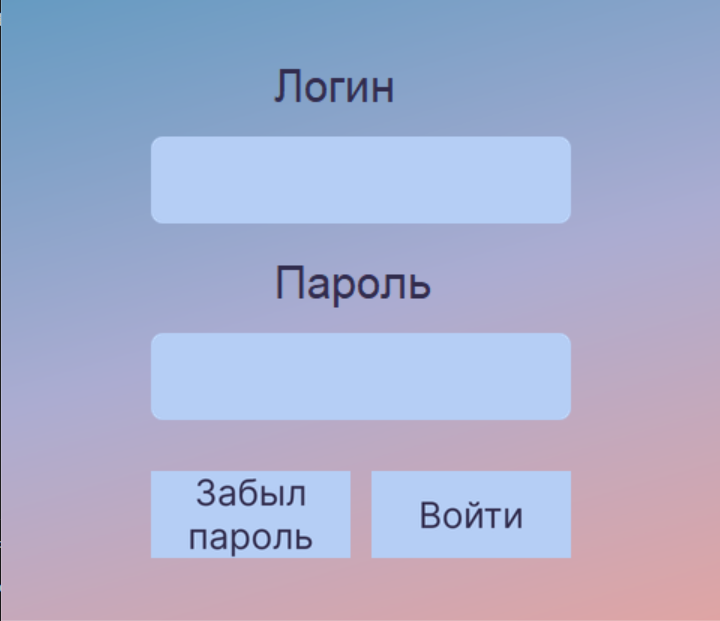


Рисунок 2.15 – Форма авторизации

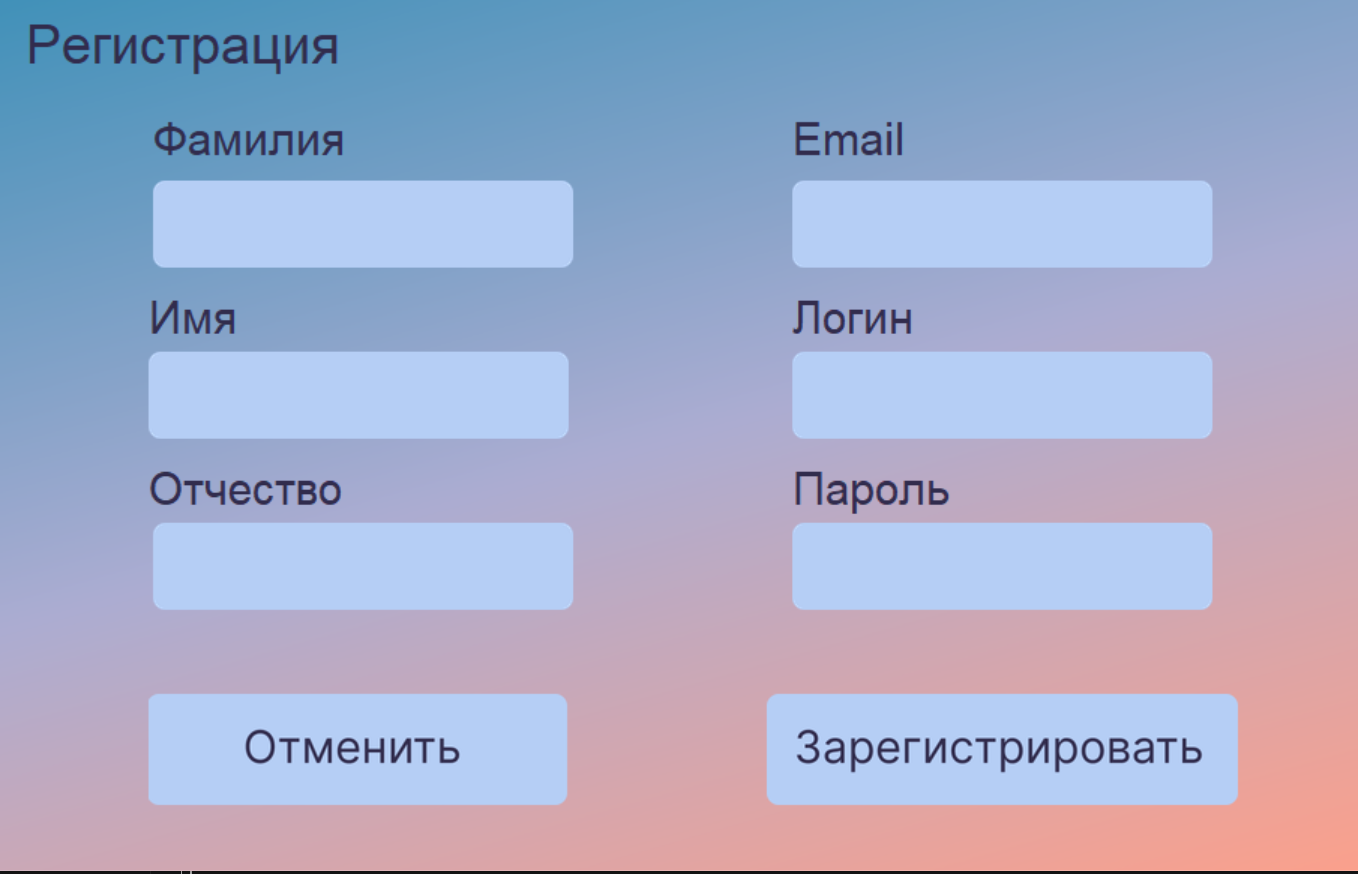


Рисунок 2.16 – Форма регистрации пользователя

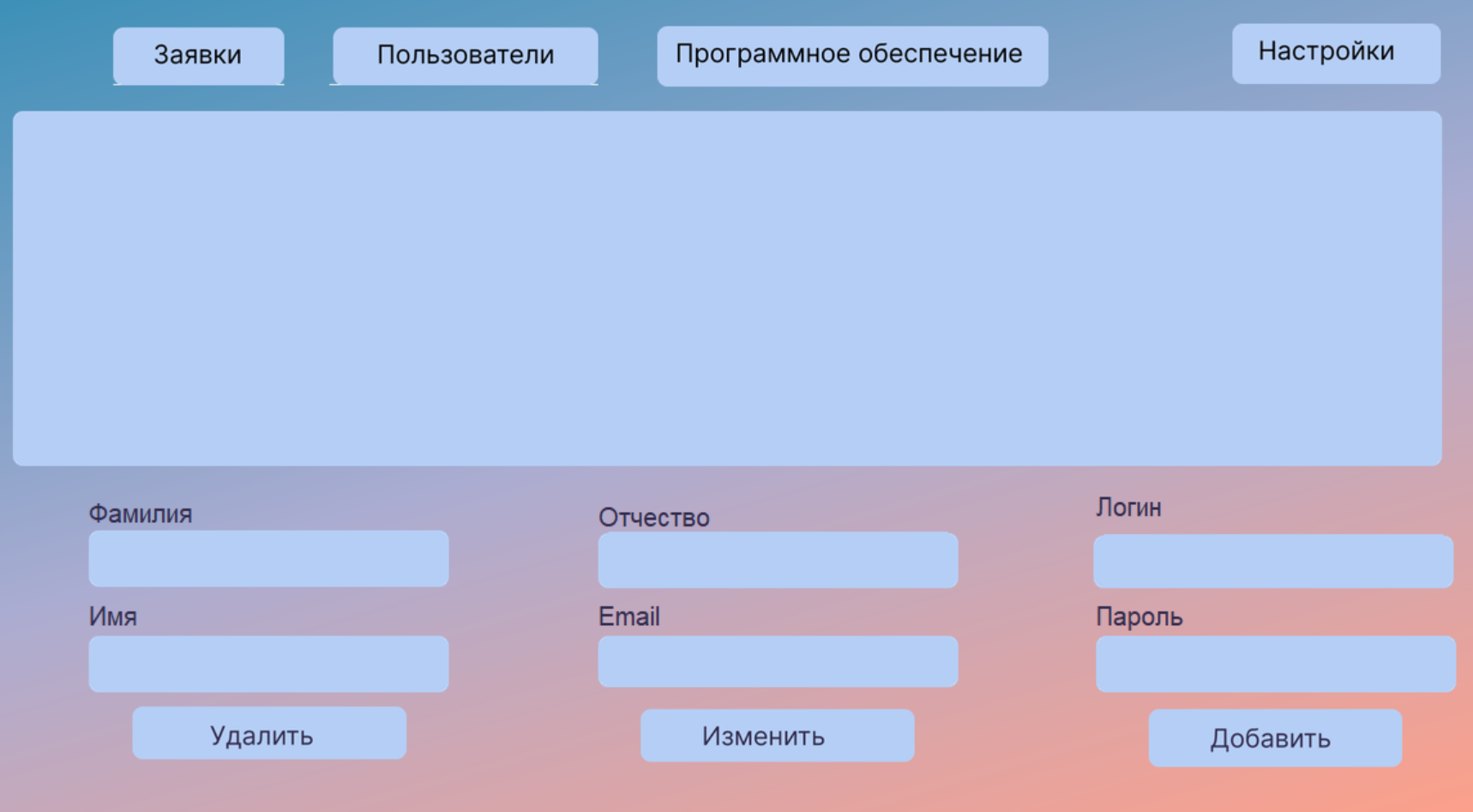


Рисунок 2.17 – Главная форма приложения

Были разработаны модули для работы с ранее разработанной базой данных, со всеми хранимыми процедурами. Процедурами добавления данных в разные таблици, удаления данных из этих таблиц и изменения необходимых данных в таблицах по тем или иным полям поиска. Все данные, которые записываются или считываются из базы данных проходят модуль шифрования. Все поля, которые нуждаются в защите шифруются и записываются. При первой авторизации разработана возможность добавление администратора в систему через программу, а не базу данных. При запуске программы модуль будет проверять наличие пользователей в системе, если пользователи в системе отсутствуют, то будет предложено добавление администратора. Подключение к базе проиходит тоже защищенным образом, путем создание драйвера ODBC, к которым имеет доступ только администратор системы компьютера. Данный вариант обеспечивает защиту данных подключения.

Был разработан модуль отправки сообщений на электронную почту с использованием HTML разметки, рисунок 2.18.

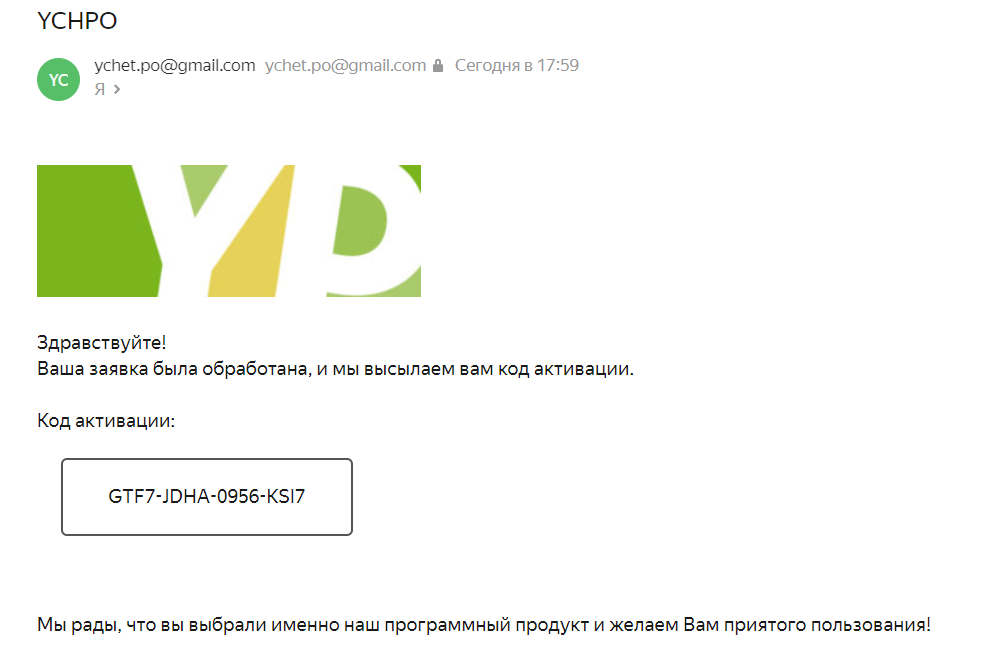


Рисунок 2.18 – Пример электронного письма

Также был реализован модуль для экспорта статистических данных в документ формата .docx рисунок 2.19.

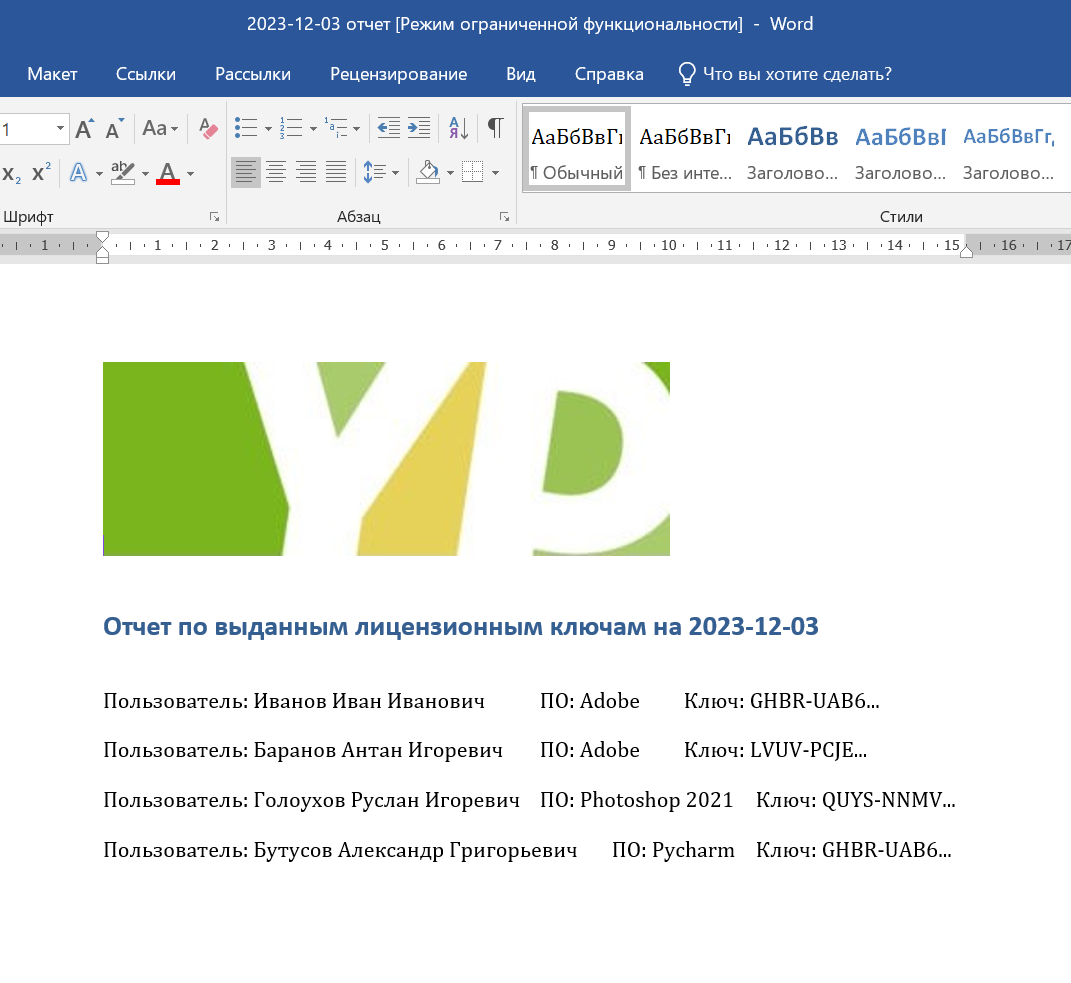


Рисунок 2.19 – Пример экспортируемого отчета

Для отладки программы был выбран метод ручного тестирование в который, свою очередь, входят следующие методы: инспекция исходного кода и сквозной просмотр. Данный метод является в данном проекте оптимальным, так как разработка производится поэтапно и каждый модуль можно протестировать, затратив меньше времени, исходя из небольшого набора основных тестов.

Часто встречаемые ошибки при разработке программы были связаны с источником данных, так как иногда не совпадали названия параметров хранимых процедур или полей в таблицах.

## Руководство пользователя/администратора ИС ххххххххххххх

## Выводы по главе 2

Методология проектирования: для реализации данного проекта, спиральная модель жизненного цикла является подходящей, что подтверждается следующими пунктами:

− Особое внимание уделяется этапам анализа и проектирования;

− Ведётся учёт версионности программы;

− При реализации программы в цели и задачи проекта можно вносить изменения.

Методология разработки:в данном случае KANBAN, что подтверждается следующими пунктами:

* Визуальный процесс разработки – расписывается последовательность задач на каждом этапе;
* Установленные сроки – данная методология направлена на разработку «срок в срок», что было установлено и в этом проекте.

Таким образом, вышеописанные пункты обосновывают выбор методологии разработки и описывают её, исходя из трёх принципов KANBAN.

Метод адаптации: для адаптации приложения используется динамическое создание элементов, так как данная методология позволяет подстроить элементы под любое разрешение экрана.

Методология тестирования: Исходя из существующих методологий тестирования была выбрана модель верификации и валидации, так как она проходит множество методов тестирования таким образом, что они тесно связаны с уровнями анализа и разработки тестов. Такая модель позволяет в полном объёме отслеживать все неточности программы и заниматься их устранением на нескольких уровнях, что в свою очередь повышает вероятность устранения всех ошибок в программном обеспечении.

Разработано техническое задание для проекта автоматизации системы учета и распределения лицензионного программного обеспечения ЧОУ ВО «Московский университет им. С.Ю. Витте»

Разработан прототип корпоративной информационной системы учета и распределения лицензионного программного обеспечения.

Разработана структура данных, выбрана СУБД. Осуществлено проектирование программного обеспечения корпоративной ИС. Проведено тестирование и устранение обнаруженных несоответствий.

# ЭКОНОМИЧЕСКАЯ ЧАСТЬ

# ОБОСНОВАНИЕ ЭКОНОМИЧЕСКОЙ ЭФФЕКТИВНОСТИ ОТ РАЗРАБОТКИ ИНФОРМАЦИОННОЙ СИСТЕМЫ

## Расчет затрат на разработку информационной системы

Ххххх ххххххх хххххх хххххх ххххххх ххххххх ххххх ххх ххххх. Хххх ххххх ххххх хххххххх хххх хххх хх ххххххх хххххххх хх х х ххххххх. Ххх ххх ххххххх ххххххх ххххххх ххххххх ххххх хххххххххх ххххххх ххххххххх хххххх. з человека:

руководитель проекта, консультант и разработчик - рисунок 3.1).

Рисунок 3.1 - Ххххх ххххххх хххххх хххххх ххххххх ххххххх ххххх ххх ххххх

Ххххх ххххххх хххххх хххххх ххххххх ххххххх ххххх ххх ххххх. Хххх ххххх ххххх хххххххх хххх хххх хх ххххххх хххххххх хх х х ххххххх. Ххх ххх ххххххх ххххххх ххххххх ххххххх ххххх хххххххххх ххххххх ххххххххх хххххх.

**статья «Материалы, покупные изделия и полуфабрикаты».**

Ххххх ххххххх хххххх хххххх ххххххх ххххххх ххххх ххх ххххх. Хххх ххххх ххххх хххххххх хххх хххх хх ххххххх хххххххх хх х х ххххххх. Ххх ххх ххххххх ххххххх ххххххх ххххххх ххххх хххххххххх ххххххх ххххххххх хххххх.

**статья «Основная заработная плата».**

Ххххх ххххххх хххххх хххххх ххххххх ххххххх ххххх ххх ххххх. Хххх ххххх ххххх хххххххх хххх хххх хх ххххххх хххххххх хх х х ххххххх. Ххх ххх ххххххх ххххххх ххххххх ххххххх ххххх хххххххххх ххххххх ххххххххх хххххх.

**3 статья «Дополнительные расходы на заработную плату» - ДЗП).**

Ххххх ххххххх хххххх хххххх ххххххх ххххххх ххххх ххх ххххх. Хххх ххххх ххххх хххххххх хххх хххх хх ххххххх хххххххх хх х х ххххххх. Ххх ххх ххххххх ххххххх ххххххх ххххххх ххххх хххххххххх ххххххх ххххххххх хххххх.

**4 статья «Страховые отчисления».**

Ххххх ххххххх хххххх хххххх ххххххх ххххххх ххххх ххх ххххх. Хххх ххххх ххххх хххххххх хххх хххх хх ххххххх хххххххх хх х х ххххххх. Ххх ххх ххххххх ххххххх ххххххх ххххххх ххххх хххххххххх ххххххх ххххххххх хххххх.

**5 статья «Накладные расходы»**

Ххххх ххххххх хххххх хххххх ххххххх ххххххх ххххх ххх ххххх. Хххх ххххх ххххх хххххххх хххх хххх хх ххххххх хххххххх хх х х ххххххх. Ххх ххх ххххххх ххххххх ххххххх ххххххх ххххх хххххххххх ххххххх ххххххххх хххххх.

**7 статья «Прочие расходы».**

Ххххх ххххххх хххххх хххххх ххххххх ххххххх ххххх ххх ххххх. Хххх ххххх ххххх хххххххх хххх хххх хх ххххххх хххххххх хх х х ххххххх. Ххх ххх ххххххх ххххххх ххххххх ххххххх ххххх хххххххххх ххххххх ххххххххх хххххх.

Договорная стоимость проекта составит 317 530 руб.

Ххххх ххххххх хххххх хххххх ххххххх ххххххх ххххх ххх ххххх. Хххх ххххх ххххх хххххххх хххх хххх хх ххххххх хххххххх хх х х ххххххх. Ххх ххх ххххххх ххххххх ххххххх ххххххх ххххх хххххххххх ххххххх ххххххххх хххххх.

## Выводы по главе 3

Оценка экономической целесообразности работы.

Выбор программных и организационно–технологических проектных решений обеспечил минимизацию финансовых, материальных и трудовых затрат. Ххххх ххххххх хххххх хххххх ххххххх ххххххх ххххх ххх ххххх. Хххх ххххх ххххх хххххххх хххх хххх хх ххххххх хххххххх хх х х ххххххх. Ххх ххх ххххххх ххххххх ххххххх ххххххх ххххх хххххххххх ххххххх ххххххххх хххххх. Ххххх ххххххх хххххх хххххх ххххххх ххххххх ххххх ххх ххххх. Хххх ххххх ххххх хххххххх хххх хххх хх ххххххх хххххххх хх х х ххххххх. Ххх ххх ххххххх ххххххх ххххххх ххххххх ххххх хххххххххх ххххххх ххххххххх хххххх.

Ххххх ххххххх хххххх хххххх ххххххх ххххххх ххххх ххх ххххх. Хххх ххххх ххххх хххххххх хххх хххх хх ххххххх хххххххх хх х х ххххххх. Ххх ххх ххххххх ххххххх ххххххх ххххххх ххххх хххххххххх ххххххх ххххххххх хххххх.

Разработка является экономически целесообразной.

# ЗАКЛЮЧЕНИЕ

В настоящей дипломной работе была реализована информационная система, направленная на снабжение лицензионным программным обеспечением пользователей Московского университета им. С.Ю. Витте.

Чтобы обеспечить качественную разработку информационной системы, проводился тщательный анализ предметной области и проблемной зоны. Основным направлением было выбрано автоматизирование процесса отправки на почту пользователям лицензионных ключей.

Исходя из анализа и выявленных требований на дальнейшем этапе проектирования особых трудностей не было. Формирование структурной схемы с описанием всех модулей и потоков чётко разграничило функционал программы, а инфологическая и даталогическая модели данных помогли реализовать структуру базы данных.

При разработке самой программы упор был сделан на проработку основного функционала и вывод отчётной документации, что способствовало углублённому изучению написания динамических элементов. Необходимо было обеспечить защиту личных данных сотрудников посредством шифрования. Проводимое тестирование по тестовым наборам и целостности данных показало стабильность работы в различных критических и стрессовых ситуациях.

В разработанном программном продукте присутствуют все заявленные функции, а именно основные функции баз данных, ведение статистики по выданным лицензионным ключам программного обеспечения, вывод отчета о выдаче лицензионных ключей по конкретному программному обеспечению в .docx, обеспечение обратной связи посредствам электронной почты, разграничение прав доступа по следующим ролям: администратор, сотрудник и пользователь, а также добавлена возможность возврата пароля по email пользователям программного продукта.

Было реализовано окно настроек главной формы программного продукта, в настройках пользователь может изменить цвет фона главной формы приложения, цвет кнопок, а также размер шрифта на главной форме, эти данные сохраняются в реестр и поэтому пользователю не нужно каждый раз заново настраивать интерфейс. Программный продукт имеет возможность автоматического входа пользователей.

Пользователи имеют возможность писать о своих проблемах и предложениях администратору, а также программа применима не только к Московскому университету им. С.Ю. Витте, но и его филиалам.

# СПИСОК ЛИТЕРАТУРЫ

1. ГК РФ Статья 1301. Ответственность за нарушение исключительного права на произведение / Гражданский кодекс Российской Федерации [Электронный ресурс] // КонсультантПлюс : [сайт]. — URL: https://www.consultant.ru/document/cons\_doc\_LAW\_64629/c2f79b53ce582e92680379e2ebd23eeb9fb7855a/ (дата обращения: 15.12.2023).
2. МЕЖГОСУДАРСТВЕННЫЙ СТАНДАРТ ЕСПД ГОСТ 19.105-78 Общие требования к программной документации / МЕЖГОСУДАРСТВЕННЫЙ СТАНДАРТ [Электронный ресурс] // Консорциум Кодекс : [сайт]. — URL: https://docs.cntd.ru/document/1200007646 (дата обращения: 15.12.2023).
3. МЕЖГОСУДАРСТВЕННЫЙ СТАНДАРТ ЕСПД ГОСТ 16.106-78 Требования к программным документам, выполненным печатным способом, режим доступа / МЕЖГОСУДАРСТВЕННЫЙ СТАНДАРТ [Электронный ресурс] // Консорциум Кодекс : [сайт]. — URL: http://docs.cntd.ru/document/1200007647 (дата обращения: 15.12.2023).
4. Златопольский, Д. М. Основы программирования на языке Python. [Текст] / Д. М. Златопольский — 1-е изд. — Москва: ДМК Пресс, 2017 — 284 c.
5. Гэддис Т. Начинаем программировать на Python. [Текст] / Гэддис Т. — 4-е изд. — Санкт-Петербруг: БХВ-Петербург, 2019 — 768 c.
6. Рейтц К., Шлюссер Т. Автостопом по Python. [Текст] / Рейтц К., Шлюссер Т. — 4-е изд. — Санкт-Петербруг: Питер, 2017 — 336 c.
7. **Федоров, Д. Ю. Программирование на языке высокого уровня Python : учебное пособие для прикладного бакалавриата [Текст] / Федоров, Д. Ю. — 2-е изд. — Москва: Юрайт, 2019 — 161 c.** Лутц, М. Программирование на Python. Т. 2 / М. Лутц. - М.: Символ, 2016. - 992 c.
8. Лутц, М. Программирование на Python. Т. 1 [Текст] / Лутц, М. — 2-е изд. — Москва: Символ, 2019 — 992 c.
9. Мэтиз, Э. Изучаем PYTHON. Программирование игр, визуализация данных, веб-приложения [Текст] / Мэтиз, Э. — 2-е изд. — Санкт-Петербург: Питер, 2017 — 496 c.
10. Саммерфилд, М. Программирование на Python 3. Подробное руководство [Текст] / Саммерфилд, М. — 1-е изд. — Москва: Символ, 2016 — 608 c.
11. Карвин, Б Программирование баз данных SQL. Типичные ошибки и их устранение [Текст] / Карвин, Б — 1-е изд. — Москва: Рид Групп, 2018 — 336 c.
12. Прайс, Д. Oracle Database 11g: SQL. Операторы SQL и программы PL/SQL [Текст] / Прайс, Д. — 1-е изд. — Москва: ЛОРИ, 2018 — 660 c.
13. Проектирование и реализация баз данных Microsoft SQL Server 2000. Учебный курс MCSE (+ CD-ROM). – Москва: Русская Редакция, **2017**. - 664 c.
14. Васильев, А. Н. Python на примерах. Практический курс по программированию. Руководство. [Текст] / Васильев, А. Н. — 1-е изд. — Москва: Наука и техника, 2018 — 752 c.
15. Гуриков, С.Р. Основы алгоритмизации и программирования на Python / С.Р. Гуриков. - М.: Форум, 2018. - **991** c.
16. Гуриков, С.Р. Основы алгоритмизации и программирования на Python [Текст] / Гуриков, С.Р. — 1-е изд. — Москва: Форум, 2018 — 707 c.
17. Златопольский, Д. М. Основы программирования на языке Python [Текст] / Златопольский, Д. М. — 1-е изд. — Москва: ДМК Пресс, 2017 — 277 c.
18. Эрик, М. Изучаем Python. Программирование игр, визуализация данных, веб-приложения [Текст] / Эрик М. — 1-е изд. — Москва: Питер, 2017 — 551 c.

Эрик М. Изучаем Python. Программирование игр, визуализация данных, веб-приложения [Текст] / Эрик М. — 2-е изд. — Москва: Питер, 2018 — 760 c.

# ПРИЛОЖЕНИЯ

ПРИЛОЖЕНИЕ А

Текст программы

АННОТАЦИЯ

Данное приложение предназначен для описания всех элементов программы Ychpo, а именно элементы базы данных, а также формы, классы программного продукта.

1. СТРУКТУРА БАЗЫ ДАННЫХ
   1. Описание объектов источника данных

Таблица 1 – Описание объектов источника данных

| Название объекта | Описание |
| --- | --- |
| 1 | 2 |
| Таблицы | |
| Dolj | Таблица, содержащая информацию о должностях |
| Error | Таблица, содержащая информацию об ошибках |
| Lickluch | Таблица, содержащая информацию о лицензионных ключах |
| PO | Таблица, содержащая информацию о программном обеспечении |
| Polz | Таблица, содержащая информацию о пользователях |
| Role | Таблица, содержащая информацию о ролях |
| Sovm | Таблица, содержащая информацию о совместимости пользователя с должностью |
| Sovmosh | Таблица, содержащая информацию о совместимости пользователя с ошибками |
| Zakaz | Таблица, содержащая информацию о заявках |
| Zayavka | Таблица, содержащая информацию о заявках |
| Процедуры | |
| Polz\_add | Обеспечивает добавление элементов в таблицу Polz |
| Polz\_edit | Обеспечение изменения элементов в таблице Polz |
| Fullpolz\_edit | Обеспечение изменения элементов в таблице Polz |
| Polzpass\_edit | Обеспечение изменения элементов в таблице Polz |
| Role\_add | Обеспечивает добавление элементов в таблицу Role |
| Dolj\_add | Обеспечивает добавление элементов в таблицу Dolj |
| Sovm\_add | Обеспечивает добавление элементов в таблицу Sovm |
| Zayavka\_add | Обеспечивает добавление элементов в таблицу Zayavka |
| Zayavkast\_edit | Обеспечение изменения элементов в таблице Zayavka |
| Po\_add | Обеспечение изменения элементов в таблице Po |
| Po\_update | Обеспечение изменения элементов в таблице Po |
| Pokol\_update | Обеспечение изменения элементов в таблице Po |
| Po\_delete | Обеспечение удаления элементов в таблице Po |
| Kluch\_add | Обеспечивает добавление элементов в таблицу Lickluch |
| Kluch\_edit | Обеспечение изменения элементов в таблице Lickluch |
| Lickluchtd\_edit | Обеспечение изменения элементов в таблице Lickluch |
| Представления | |
| Polzv | Выводит информацию из таблиц Polz, Sovm, Dolj, Role |
| Zahazi | Выводит информацию из таблиц Zayavka, Polz, PO |
| Rols | Выводит информацию из таблиц Dolj, Role |
| Statistika | Выводит информацию из таблиц Zakaz, Lickluch, PO, Zayavka, Polz |
| Izmlickluch | Выводит информацию из таблиц Lickluch, PO |

* 1. Скрипт объектов источника данных

PO

CREATE TABLE [DBO].[PO]

(

[id\_PO] INT NOT NULL IDENTITY (1,1),

[naim\_po] varchar(max) NOT NULL,

[kol\_po] int NOT NULL,

[vers\_po] varchar(max) NOT NULL,

constraint [PK\_id\_po] PRIMARY KEY CLUSTERED

([id\_po] ASC) on [PRIMARY],

)

Role

CREATE TABLE [DBO].[role]

(

[id\_role] INT NOT NULL IDENTITY (1,1),

[naim\_role] varchar(max) NOT NULL,

[polz\_role] bit NOT NULL,

[zayavka\_role] bit NOT NULL,

[po\_role] bit NOT NULL,

[zakaz\_role] bit NOT NULL,

constraint [PK\_id\_role] PRIMARY KEY CLUSTERED

([id\_role] ASC) on [PRIMARY],

)

Dolj

CREATE TABLE [DBO].[dolj]

(

[id\_dolj] INT NOT NULL IDENTITY (1,1),

[naim\_dolj] varchar(max) NOT NULL,

[role\_id] int null,

constraint [PK\_id\_dolj] PRIMARY KEY CLUSTERED

([id\_dolj] ASC) on [PRIMARY],

CONSTRAINT [FK\_role\_id] FOREIGN KEY ([role\_id])

REFERENCES [DBO].[role]([id\_role]),

)

Polz

CREATE TABLE [DBO].[polz]

(

[id\_polz] INT NOT NULL IDENTITY (1,1),

[F\_P] varchar(max) NOT NULL,

constraint [PK\_id\_polz] PRIMARY KEY CLUSTERED

([id\_polz] ASC) on [PRIMARY],

)

Sovm

CREATE TABLE [DBO].[sovm]

(

[id\_sovm] INT NOT NULL IDENTITY (1,1),

[polzsovm\_id] int Null,

[dolj\_id] int null,

constraint [PK\_id\_sovm] PRIMARY KEY CLUSTERED

([id\_sovm] ASC) on [PRIMARY],

CONSTRAINT [FK\_polzsovm\_id] FOREIGN KEY ([polzsovm\_id])

REFERENCES [DBO].[polz]([id\_polz]),

CONSTRAINT [FK\_dolj\_id] FOREIGN KEY ([dolj\_id])

REFERENCES [DBO].[dolj]([id\_dolj]),

)

Zayavka

CREATE TABLE [DBO].[zayavka]

(

[id\_zayavka] INT NOT NULL IDENTITY (1,1),

[status] varchar(max) NOT NULL,

[polz\_id] int NOT NULL,

[poz\_id] int NOT NULL,

constraint [PK\_id\_zayavka] PRIMARY KEY CLUSTERED

([id\_zayavka] ASC) on [PRIMARY],

CONSTRAINT [FK\_polz\_id] FOREIGN KEY ([polz\_id])

REFERENCES [DBO].[polz]([id\_polz]),

CONSTRAINT [FK\_poz\_id] FOREIGN KEY ([poz\_id])

REFERENCES [DBO].[po]([id\_po]),

)

Lickluch

CREATE TABLE [DBO].[lickluch]

(

[id\_lickluch] INT NOT NULL IDENTITY (1,1),

[kod] varchar(max) NOT NULL,

[statuskluch] bit NOT NULL,

[time] varchar(5) NULL,

[date] varchar(10) NULL,

[pol\_id] int NOT NULL,

constraint [PK\_id\_lickluch] PRIMARY KEY CLUSTERED

([id\_lickluch] ASC) on [PRIMARY],

CONSTRAINT [FK\_pol\_id] FOREIGN KEY ([pol\_id])

REFERENCES [DBO].[po]([id\_po]),

)

Error

CREATE TABLE [DBO].[error]

(

[id\_error] INT NOT NULL IDENTITY (1,1),

[naim\_error] varchar(max) NOT NULL,

[opisanie] varchar(max) NOT NULL,

[statusError] bit NOT NULL,

[sposobYstranenia] varchar(max) NULL,

constraint [PK\_id\_error] PRIMARY KEY CLUSTERED

([id\_error] ASC) on [PRIMARY],

)

Sovmosh

CREATE TABLE [DBO].[sovmosh]

(

[id\_sovmosh] INT NOT NULL IDENTITY (1,1),

[polzsovmosh\_id] int Null,

[error\_id] int null,

constraint [PK\_id\_sovmosh] PRIMARY KEY CLUSTERED

([id\_sovmosh] ASC) on [PRIMARY],

CONSTRAINT [FK\_polzsovmosh\_id] FOREIGN KEY ([polzsovmosh\_id])

REFERENCES [DBO].[polz]([id\_polz]),

CONSTRAINT [FK\_error\_id] FOREIGN KEY ([error\_id])

REFERENCES [DBO].[error]([id\_error]),

)

go

Polzv

create view [dbo].[polzv]

as select id\_polz as 'Номер пользователя',F\_P as 'Фамилия пользователя', I\_P as 'Имя пользователя',O\_P as 'Очество пользователя',email as 'Email',login as'Логин', password as'Пароль',id\_dolj as 'Номер должности',naim\_dolj as 'Должность',id\_role as 'Номер роли',naim\_role as 'роль', polz\_role as 'Доступ к пользователям',zayavka\_role as 'Доступ к заявкам',po\_role as'Доступ к ПО', zakaz\_role as 'Доступ к заказам'

from

polz inner join

sovm on sovm.polzsovm\_id=polz.id\_polz inner join

dolj on dolj.id\_dolj=sovm.dolj\_id inner join

role on role.id\_role=dolj.role\_id

go

Zahazi

create view [dbo].[zahazi]

as select id\_zayavka as 'Номер заявки',naim\_po as 'Название ПО', vers\_po as 'Версия ПО',status as 'Статус',login as 'Логин'

from

zayavka inner join

polz on zayavka.polz\_id=polz.id\_polz inner join

PO on id\_PO=poz\_id

go

Rols

create view [dbo].[rols]

as select id\_dolj as 'Номер должности',naim\_dolj as 'Должность',id\_role as 'Номер роли',naim\_role as 'роль', polz\_role as 'Доступ к пользователям',zayavka\_role as 'Доступ к заявкам',po\_role as'Доступ к ПО', zakaz\_role as 'Доступ к заказам'

from

dolj inner join

role on role.id\_role=dolj.role\_id

go

Statistika

create view [dbo].[statistika]

as select id\_lickluch as 'Номер',naim\_po as 'Название ПО', vers\_po as 'Версия ПО',F\_P as'Фамилия',I\_P as 'Имя',O\_P as 'Отчество',time as 'Время заказа',date as 'Дата заказа' from zayavka

join polz on polz.id\_polz = zayavka.polz\_id

join PO on Po.id\_PO = zayavka.poz\_id

join lickluch on lickluch.pol\_id = PO.id\_PO

where lickluch.statuskluch = 1 and zayavka.status = 'Готово'

go

Izmlickluch

create view [dbo].[izmlickluch]

as select id\_lickluch as 'Номер',naim\_po as 'Название ПО', kod as 'Код',statuskluch as 'Выдан' from lickluch

join PO on PO.id\_PO = lickluch.pol\_id

go

Polz\_add

CREATE PROCEDURE [DBO].[polz\_add]

(

@F\_P varchar(max),

@I\_P varchar(max),

@O\_P varchar(max),

@email varchar(max),

@login varchar(max),

@password varchar(max),

@dostup bit

)

AS

insert into [dbo].[polz]([F\_P],[I\_P],[O\_P],[email],[login],[password],[dostup]) values((@F\_P),(@I\_P),(@O\_P),(@email),(@login),(@password),(@dostup));

go

Polz\_edit

CREATE PROCEDURE [DBO].[polz\_edit]

@id\_polz int,

@F\_P varchar(max),

@I\_P varchar(max),

@O\_P varchar(max),

@email varchar(max),

@login varchar(max)

AS

update [dbo].polz

set

F\_P=@F\_P,

I\_P=@I\_P,

O\_P=@O\_P,

email=@email,

login=@login

where id\_polz=@id\_polz;

go

Fullpolz\_edit

CREATE PROCEDURE [DBO].[fullpolz\_edit]

@id\_polz int,

@F\_P varchar(max),

@I\_P varchar(max),

@O\_P varchar(max),

@email varchar(max),

@login varchar(max),

@password varchar(max)

AS

update [dbo].polz

set

F\_P=@F\_P,

I\_P=@I\_P,

O\_P=@O\_P,

email=@email,

login=@login,

password=@password

where id\_polz=@id\_polz;

go

Polzpass\_edit

CREATE PROCEDURE [DBO].[polzpass\_edit]

@id\_polz int,

@password varchar(max)

AS

update [dbo].polz

set

password=@password

where id\_polz=@id\_polz;

go

Role\_add

CREATE PROCEDURE [DBO].[role\_add]

(

@naim\_role varchar(max),

@polz\_role bit,

@zayavka\_role bit,

@po\_role bit,

@zakaz\_role bit

)

AS

insert into [dbo].[role]([naim\_role],[polz\_role],[zayavka\_role],[po\_role],[zakaz\_role]) values((@naim\_role),(@polz\_role),(@zayavka\_role),(@po\_role),(@zakaz\_role));

go

Dolj\_add

CREATE PROCEDURE [DBO].[dolj\_add]

(

@naim\_dolj varchar(max),

@role\_id int

)

AS

insert into [dbo].[dolj]([naim\_dolj],[role\_id]) values((@naim\_dolj),(@role\_id));

go

Sovm\_add

CREATE PROCEDURE [DBO].[sovm\_add]

(

@polzsovm\_id int,

@dolj\_id int

)

AS

insert into [dbo].[sovm]([polzsovm\_id],[dolj\_id]) values((@polzsovm\_id),(@dolj\_id));

go

Zayavka\_add

CREATE PROCEDURE [DBO].[zayavka\_add]

(

@status varchar(max),

@poz\_id int,

@polz\_id int

)

AS

insert into [dbo].[zayavka]([poz\_id],[status],[polz\_id]) values((@poz\_id),(@status),(@polz\_id));

go

Zayavka\_edit

CREATE PROCEDURE [DBO].[zayavkast\_edit]

@id\_zayavkast int,

@status varchar(max)

AS

update [dbo].zayavka

set

status=@status

where id\_zayavka=@id\_zayavkast;

go

Lickluchtd\_edit

CREATE PROCEDURE [DBO].[lickluchtd\_edit]

(

@id\_lickluch int,

@time varchar(5),

@date varchar(10),

@statuskluch bit

)

AS

update [dbo].lickluch

set

time=@time,

date=@date,

statuskluch=@statuskluch

where id\_lickluch=@id\_lickluch;

go

PO\_add

CREATE PROCEDURE [DBO].[po\_add]

(

@naim\_po varchar(max),

@kol\_po int,

@vers\_po varchar(max)

)

AS

insert into [dbo].[po]([naim\_po],[kol\_po],[vers\_po]) values((@naim\_po),(@kol\_po),(@vers\_po));

go

PO\_update

CREATE PROCEDURE [DBO].[po\_update]

(

@id\_PO int,

@naim\_po varchar(max),

@kol\_po int,

@vers\_po varchar(max)

)

AS

update [dbo].po

set

naim\_po=@naim\_po,

kol\_po=@kol\_po,

vers\_po=@vers\_po

where @id\_PO = id\_PO

go

Pokol\_update

CREATE PROCEDURE [DBO].[pokol\_update]

(

@id\_PO int,

@kol\_po int

)

AS

update [dbo].po

set

kol\_po=@kol\_po

where @id\_PO = id\_PO

go

PO\_delete

CREATE PROCEDURE [DBO].[po\_delete]

@id\_po int

AS

DELETE from dbo.po

where id\_po=@id\_po;

go

Kluch\_add

CREATE PROCEDURE [DBO].[kluch\_add]

(

@kod varchar(max),

@statuskluch bit,

@pol\_id int

)

AS

insert into [dbo].[lickluch]([kod],[statuskluch],[pol\_id]) values((@kod),(@statuskluch),(@pol\_id));

go

Error\_add

CREATE PROCEDURE [DBO].[error\_add]

(

@naim\_error varchar(max),

@opisanie varchar(max),

@statusError bit,

@sposobYstranenia varchar(max)

)

AS

insert into [dbo].[error]([naim\_error],[opisanie],[statusError],[sposobYstranenia]) values((@naim\_error),(@opisanie),(@statusError),(@sposobYstranenia));

go

Error\_update

CREATE PROCEDURE [DBO].[error\_update]

(

@id\_error int,

@naim\_error varchar(max),

@opisanie varchar(max),

@statusError bit,

@sposobYstranenia varchar(max)

)

AS

update [dbo].error

set

naim\_error=@naim\_error,

opisanie=@opisanie,

statusError=@statusError,

sposobYstranenia=@sposobYstranenia

where @id\_error = id\_error

go

Error\_delete

CREATE PROCEDURE [DBO].[error\_delete]

@id\_error int

AS

DELETE from dbo.error

where

id\_error=@id\_error;

go

Kluch\_edit

CREATE PROCEDURE [DBO].[kluch\_edit]

@id\_lickluch int,

@kod varchar(max)

AS

update [dbo].lickluch

set

kod=@kod

where id\_lickluch=@id\_lickluch;

go

1. ПРОГРАММНЫЙ ПРОДУКТ
   1. Текст программного продукта или программного изделия

all\_updates

from bd\_connect import BdConnect

class Update\_bd:

    def all\_edit\_func(self, requestString):

        '''Основной алгоритм отправки запросов в бд'''

        connect = BdConnect()

        connect.connect()

        dbCursor = connect.conn.cursor()

        dbCursor.execute(requestString)

        connect.conn.commit()

        connect.disconnect()

bd\_add\_data

from all\_updates import Update\_bd

class Bd\_add:

    def add\_error(self, naim\_error: str, opisanie: str, sposobYstranenia: str):

        ''' Добавление данных об ошибке'''

        requestString = f'''

        EXEC error\_add

        @naim\_error = {naim\_error},

        @opisanie = {opisanie},

        @statusError = 1,

        @sposobYstranenia = {sposobYstranenia}

        '''

        update\_bd = Update\_bd()

        update\_bd.all\_edit\_func(requestString)

    def add\_polz(self, F\_P: str, I\_P: str, O\_P: str, email: str, login: str, password: str):

        ''' Добавление данных о пользователе'''

        requestString = f'''

        EXEC polz\_add

        @F\_P = {F\_P},

        @I\_P = {I\_P},

        @O\_P = {O\_P},

        @email = {email},

        @login = {login},

        @password = {password},

        @dostup = 1

        '''

        update\_bd = Update\_bd()

        update\_bd.all\_edit\_func(requestString)

    def add\_kluch(self, kod: str, statuskluch: bool, pol\_id: int):

        ''' Добавление данных о лицензионном ключе'''

        requestString = f'''

        EXEC kluch\_add

        @kod = {kod},

        @statuskluch ={statuskluch},

        @pol\_id = {pol\_id}

        '''

        update\_bd = Update\_bd()

        update\_bd.all\_edit\_func(requestString)

    def add\_po(self, naim\_po: str, kol\_po: int, vers\_po: str):

        ''' Добавление данных о программном обеспечении'''

        requestString = f'''

        EXEC po\_add

        @naim\_po = {naim\_po},

        @kol\_po = {kol\_po},

        @vers\_po = {vers\_po}

        '''

        update\_bd = Update\_bd()

        update\_bd.all\_edit\_func(requestString)

    def add\_zayavka(self, status: str, poz\_id: int, polz\_id: int):

        ''' Добавление данных о заявке обеспечении'''

        requestString = f'''

        EXEC zayavka\_add

        @status = {status},

        @poz\_id = {poz\_id},

        @polz\_id = {polz\_id}

        '''

        update\_bd = Update\_bd()

        update\_bd.all\_edit\_func(requestString)

    def add\_dolj(self, naim\_dolj: str, role\_id: int):

        ''' Добавление данных о должности'''

        requestString = f'''

        EXEC dolj\_add

        @naim\_dolj = {naim\_dolj},

        @role\_id = {role\_id}

        '''

        update\_bd = Update\_bd()

        update\_bd.all\_edit\_func(requestString)

    def add\_sovm\_polz\_dolj(self, polzsovm\_id: int, dolj\_id: int):

        ''' Добавление данных о таблице совместимости'''

        requestString = f'''

        EXEC sovm\_add

        @polzsovm\_id = {polzsovm\_id},

        @dolj\_id = {dolj\_id}

        '''

        update\_bd = Update\_bd()

        update\_bd.all\_edit\_func(requestString)

    def add\_sovmosh\_polz\_error(self, polzsovmosh\_id: int, error\_id: int):

        ''' Добавление данных о таблице совместимости'''

        requestString = f'''

        EXEC sovmosh\_add

        @polzsovmosh\_id = {polzsovmosh\_id},

        @error\_id = {error\_id}

        '''

        update\_bd = Update\_bd()

        update\_bd.all\_edit\_func(requestString)

    def role\_dolj(self, naim\_role: str, polz\_role: bool, zayavka\_role: bool, po\_role: bool, zakaz\_role: bool):

        ''' Добавление данных о роли'''

        requestString = f'''

        EXEC role\_add

        @naim\_role = {naim\_role},

        @polz\_role = {polz\_role},

        @zayavka\_role ={zayavka\_role},

        @po\_role = {po\_role},

        @zakaz\_role = {zakaz\_role}

        '''

        update\_bd = Update\_bd()

        update\_bd.all\_edit\_func(requestString)

bd\_connect

"""

Connects to a SQL database using pyodbc

"""

import pyodbc

class BdConnect:

    conn = ""

    # Подключение к базе черездрайвер ODBC

    def connect(self):

        self.conn = pyodbc.connect(

            """DRIVER=SQL Server Native Client 11.0;

            DATABASE=Ychpo;

            Trusted\_Connection=Yes;

            SERVER=DESKTOP-H7TURC8\SQLEXPRESS """

        )

        print("open")

    # def get\_connect(self):

    #     return self.conn

    def disconnect(self):

        print("close")

        self.conn.close()

bd\_delete

from all\_updates import Update\_bd

class Bd\_delete:

    def delete\_error(self, id\_error: int):

        ''' Удаление данных об ошибке'''

        requestString = f'''

        EXEC error\_delete

        @id\_error = {id\_error}

        '''

        update\_bd = Update\_bd()

        update\_bd.all\_edit\_func(requestString)

    def delete\_error(self, id\_po: int):

        ''' Удаление данных об ошибке'''

        requestString = f'''

        EXEC po\_delete

        @id\_po = {id\_po}

        '''

        update\_bd = Update\_bd()

        update\_bd.all\_edit\_func(requestString)

bd\_edit\_data

from all\_updates import Update\_bd

class Bd\_edit:

    def edit\_error(self, id\_error: int, naim\_error: str, opisanie: str, sposobYstranenia: str):

        ''' Изменение данных об ошибке'''

        requestString = f'''

        EXEC error\_update

        @id\_error = {id\_error},

        @naim\_error = {naim\_error},

        @opisanie = {opisanie},

        @statusError = 1,

        @sposobYstranenia = {sposobYstranenia}

        '''

        update\_bd = Update\_bd()

        update\_bd.all\_edit\_func(requestString)

    def edit\_polz(self, id\_polz: int, F\_P: str, I\_P: str, O\_P: str, email: str, login: str):

        ''' Изменение данных о пользователе'''

        requestString = f'''

        EXEC polz\_edit

        @id\_polz = {id\_polz},

        @F\_P = {F\_P},

        @I\_P = {I\_P},

        @O\_P = {O\_P},

        @email = {email},

        @login = {login}

        '''

        update\_bd = Update\_bd()

        update\_bd.all\_edit\_func(requestString)

    def edit\_polz\_full(self, id\_polz: int, F\_P: str, I\_P: str, O\_P: str, email: str, login: str, password: str):

        ''' Изменение всех данных о пользователе, в том числе и пароль'''

        requestString = f'''

        EXEC fullpolz\_edit

        @id\_polz = {id\_polz},

        @F\_P = {F\_P},

        @I\_P = {I\_P},

        @O\_P = {O\_P},

        @email = {email},

        @login = {login},

        @password = {password}

        '''

        update\_bd = Update\_bd()

        update\_bd.all\_edit\_func(requestString)

    def edit\_polz\_password(self, id\_polz: int, password: str):

        ''' Изменение пароля пользователя'''

        requestString = f'''

        EXEC polzpass\_edit

        @id\_polz ={id\_polz},

        @password = {password}

        '''

        update\_bd = Update\_bd()

        update\_bd.all\_edit\_func(requestString)

    def edit\_zayavka(self, id\_zayavkast: int, status: str):

        ''' Изменение данных о заявке'''

        requestString = f'''

        EXEC zayavkast\_edit

        @id\_zayavkast = {id\_zayavkast},

        @status = {status}

        '''

        update\_bd = Update\_bd()

        update\_bd.all\_edit\_func(requestString)

    def edit\_lickluch(self, id\_lickluch: int, time: str, date: str, statuskluch: bool):

        ''' Изменение данных об используемом лицензионном ключе'''

        requestString = f'''

        EXEC lickluchtd\_edit

        @id\_lickluch = {id\_lickluch},

        @time = {time},

        @date = {date},

        @statuskluch = {statuskluch}

        '''

        update\_bd = Update\_bd()

        update\_bd.all\_edit\_func(requestString)

    def edit\_po(self, id\_PO: int, naim\_po:str, kol\_po: int, vers\_po: str):

        ''' Изменение данных о программном обеспечении'''

        requestString = f'''

        EXEC pokol\_update

        @id\_PO = {id\_PO},

        @naim\_po = {naim\_po},

        @kol\_po = {kol\_po},

        @vers\_po = {vers\_po}

        '''

        update\_bd = Update\_bd()

        update\_bd.all\_edit\_func(requestString)

    def edit\_kol\_po(self, id\_PO: int, kol\_po: int):

        ''' Изменение данных о количестве лицензионных ключей'''

        requestString = f'''

        EXEC pokol\_update

        @id\_PO = {id\_PO},

        @kol\_po = {kol\_po}

        '''

        update\_bd = Update\_bd()

        update\_bd.all\_edit\_func(requestString)

    def edit\_kluch(self, id\_lickluch: int, kod: str):

        ''' Изменение данных о лицензионном ключе'''

        requestString = f'''

        EXEC kluch\_edit

        @id\_lickluch = {id\_lickluch},

        @kod = {kod}

        '''

        update\_bd = Update\_bd()

        update\_bd.all\_edit\_func(requestString)

shifr

from Crypto.Cipher import AES

from base64 import b64encode, b64decode

class Crypt:

    def \_\_init\_\_(self, salt='SlTKeYOpHygTYkP3'):

        self.salt = salt.encode('utf8')

        self.enc\_dec\_method = 'utf-8'

    def encrypt(self, str\_to\_enc, str\_key = "MyKey4TestingYnP"):

        try:

            aes\_obj = AES.new(str\_key.encode('utf-8'), AES.MODE\_CFB, self.salt)

            hx\_enc = aes\_obj.encrypt(str\_to\_enc.encode('utf8'))

            mret = b64encode(hx\_enc).decode(self.enc\_dec\_method)

            return mret

        except ValueError as value\_error:

            if value\_error.args[0] == 'IV must be 16 bytes long':

                raise ValueError('Encryption Error: SALT must be 16 characters long')

            elif value\_error.args[0] == 'AES key must be either 16, 24, or 32 bytes long':

                raise ValueError('Encryption Error: Encryption key must be either 16, 24, or 32 characters long')

            else:

                raise ValueError(value\_error)

    def decrypt(self, enc\_str, str\_key = "MyKey4TestingYnP"):

        try:

            aes\_obj = AES.new(str\_key.encode('utf8'), AES.MODE\_CFB, self.salt)

            str\_tmp = b64decode(enc\_str.encode(self.enc\_dec\_method))

            str\_dec = aes\_obj.decrypt(str\_tmp)

            mret = str\_dec.decode(self.enc\_dec\_method)

            return mret

        except ValueError as value\_error:

            if value\_error.args[0] == 'IV must be 16 bytes long':

                raise ValueError('Decryption Error: SALT must be 16 characters long')

            elif value\_error.args[0] == 'AES key must be either 16, 24, or 32 bytes long':

                raise ValueError('Decryption Error: Encryption key must be either 16, 24, or 32 characters long')

            else:

                raise ValueError(value\_error)

import\_doc

import docx

import os

from datetime import date

class ImportDoc:

    # Создание и добавление стандартных параметров документа

    doc = docx.Document()

    doc.add\_picture('modules/images/Ychpo.jpg', width = docx.shared.Cm(10))

    doc.add\_heading(f'Отчет по выданным лицензионным ключам на {date.today()} \n', 1)

    # добавление параграфов в документ

    def importDoc(self, user: str, po: str, kluch: str):

        # добавляем параграф

        self.doc.add\_paragraph(f'Пользователь: {user} \t ПО: {po} \t Ключ: {kluch[:-5]}')

    # Сохранение документа

    def saveDoc(self):

        user = os.environ.get( "USERNAME" )

        self.doc.save(f'C:\\users\{user}\Desktop\{date.today()} отчет.docx')

send\_email

import smtplib

from email.mime.multipart import MIMEMultipart

from email.mime.text import MIMEText

class Send\_email:

    # Отправка сообщения на электронную почту

    def send(self, my\_email:str, password: str, enemy\_email: str, html: str):

        msg = MIMEMultipart('alternative')

        msg['Subject'] = "YCHPO"

        msg['From'] = my\_email

        msg['To'] = enemy\_email

        part = MIMEText(html, 'html')

        msg.attach(part)

        # Send the message via local SMTP server.

        mail = smtplib.SMTP('smtp.gmail.com', 587)

        mail.ehlo()

        mail.starttls()

        mail.login(my\_email, password)

        mail.sendmail(my\_email, enemy\_email, msg.as\_string())

        mail.quit()

    # Создание HTML для отправки кода лицензионного ключа

    def kluchHtml(self, kluch: str):

        html = f"""

        <html><body><br><img src=\"https://storage.googleapis.com/thl-blog-production/2017/10/a5d6fc4b-banneri-320x110.jpg\" alt=\"ACORP\">

        <br><br>Здравствуйте!

        <br>Ваша заявка была обработана, и мы высылаем вам код активации.

        <br>

        <br>Код активации:

        <p style="border:2px solid #555; border-radius:5px; width: 200px; text-align:center;  margin:20px; padding:20px;" > {kluch} </p>

        <br>

        <br>Мы рады, что вы выбрали именно наш программный продукт и желаем Вам приятого пользования!</body></html>

        """

        return html

    # Создание HTML для отправки кода подтверждения возврата пароля

    def codPasseordHtml(self, kod: str):

        html = f"""

        <html><body><br><img src=\"https://storage.googleapis.com/thl-blog-production/2017/10/a5d6fc4b-banneri-320x110.jpg\" alt=\"ACORP\">

        <br><br>Здравствуйте!

        <br>Вам выслан код подтверждения для восстановления пароля.

        <br>

        <br>Код активации:

        <p style="border:2px solid #555; border-radius:5px; width: 200px; text-align:center;  margin:20px; padding:20px;" > {kod} </p>

        <br>

        <br>Мы рады, что вы выбрали именно наш программный продукт и желаем Вам приятого пользования!</body></html>

        """

        return html

        # Создание HTML для отправки кода подтверждения возврата пароля

    # Создание HTML для отправки сообщения об ошибке

    def errorHtml(self, user: str, error\_value: str):

        html = f"""

        <html><body><br><img src=\"https://storage.googleapis.com/thl-blog-production/2017/10/a5d6fc4b-banneri-320x110.jpg\" alt=\"ACORP\">

        <br><br>Здравствуйте!

        <br>Пользователь {user} прислал сообщение об ошибке:

        <br>

        <p style="border:2px solid #555; border-radius:5px; width: 200px; text-align:center;  margin:20px; padding:20px;" > {error\_value} </p>

        <br>

        """

        return html

main

import sys

from pathlib import Path

from tkinter import Tk, Canvas, Entry, Text, Button, PhotoImage

import os

from registration import show as sh

def show():

    OUTPUT\_PATH = Path(\_\_file\_\_).parent

    ASSETS\_PATH = OUTPUT\_PATH / Path(r"assets\autorization")

    def relative\_to\_assets(path: str) -> Path:

        return ASSETS\_PATH / Path(path)

    def open\_registration():

        sh()

        exit()

    window = Tk()

    window.geometry("480x416")

    window.configure(bg="#FFFFFF")

    canvas = Canvas(

        window,

        bg="#FFFFFF",

        height=416,

        width=480,

        bd=0,

        highlightthickness=0,

        relief="ridge"

    )

    canvas.place(x=0, y=0)

    image\_image\_1 = PhotoImage(

        file=relative\_to\_assets("image\_1.png"))

    image\_1 = canvas.create\_image(

        240.0,

        208.0,

        image=image\_image\_1

    )

    entry\_image\_1 = PhotoImage(

        file=relative\_to\_assets("entry\_1.png"))

    entry\_bg\_1 = canvas.create\_image(

        240.0,

        253.0,

        image=entry\_image\_1

    )

    entry\_1 = Text(

        bd=0,

        bg="#B5CEF5",

        fg="#000716",

        highlightthickness=0,

        font=("Inter", 30)

    )

    entry\_1.place(

        x=108.0,

        y=224.0,

        width=264.0,

        height=56.0

    )

    entry\_image\_2 = PhotoImage(

        file=relative\_to\_assets("entry\_2.png"))

    entry\_bg\_2 = canvas.create\_image(

        240.0,

        122.0,

        image=entry\_image\_2

    )

    entry\_2 = Text(

        bd=0,

        bg="#B5CEF5",

        fg="#000716",

        highlightthickness=0,

        font=("Inter", 30 )

    )

    entry\_2.place(

        x=108.0,

        y=93.0,

        width=264.0,

        height=56.0

    )

    button\_image\_1 = PhotoImage(

        file=relative\_to\_assets("button\_1.png"))

    button\_1 = Button(

        image=button\_image\_1,

        borderwidth=0,

        highlightthickness=0,

        relief="flat"

    )

    button\_1.place(

        x=247.0,

        y=316.0,

        width=133.0,

        height=58.0

    )

    button\_image\_2 = PhotoImage(

        file=relative\_to\_assets("button\_2.png"))

    button\_2 = Button(

        image=button\_image\_2,

        borderwidth=0,

        highlightthickness=0,

        command=open\_registration,

        relief="flat"

    )

    button\_2.place(

        x=100.0,

        y=316.0,

        width=133.0,

        height=58.0

    )

    canvas.create\_text(

        182.0,

        42.0,

        anchor="nw",

        text="Логин",

        fill="#312C4D",

        font=("Inter", 30 \* -1)

    )

    canvas.create\_text(

        182.0,

        173.0,

        anchor="nw",

        text="Пароль",

        fill="#312C4D",

        font=("Inter", 30 \* -1)

    )

    window.resizable(False, False)

    window.mainloop()

registration

from pathlib import Path

import os

from tkinter import Tk, Canvas, Entry, Text, Button, PhotoImage

def show():

    OUTPUT\_PATH = Path(\_\_file\_\_).parent

    ASSETS\_PATH = OUTPUT\_PATH / Path(r"forms/assets/registration")

    def relative\_to\_assets(path: str) -> Path:

        return ASSETS\_PATH / Path(path)

    def open\_registration():

        registration.withdraw()

        os.system('python main.py')

    registration = Tk()

    registration.geometry("909x581")

    registration.configure(bg='#FFFFFF')

    canvas = Canvas(

        registration,

        bg="#FFFFFF",

        height=581,

        width=909,

        bd=0,

        highlightthickness=0,

        relief="ridge"

    )

    canvas.place(x=0, y=0)

    image\_image\_1 = PhotoImage(

        file=relative\_to\_assets("image\_1.png"))

    image\_1 = canvas.create\_image(

        454.0,

        290.0,

        image=image\_image\_1

    )

    entry\_image\_1 = PhotoImage(

        file=relative\_to\_assets("entry\_1.png"))

    entry\_bg\_1 = canvas.create\_image(

        670.0,

        378.0,

        image=entry\_image\_1

    )

    entry\_1 = Text(

        bd=0,

        bg="#B5CEF5",

        fg="#000716",

        highlightthickness=0,

        font=("Inter", 30 \* -1)

    )

    entry\_1.place(

        x=538.0,

        y=349.0,

        width=264.0,

        height=56.0

    )

    entry\_image\_2 = PhotoImage(

        file=relative\_to\_assets("entry\_2.png"))

    entry\_bg\_2 = canvas.create\_image(

        670.0,

        150.0,

        image=entry\_image\_2

    )

    entry\_2 = Text(

        bd=0,

        bg="#B5CEF5",

        fg="#000716",

        highlightthickness=0,

        font=("Inter", 30 \* -1)

    )

    entry\_2.place(

        x=538.0,

        y=121.0,

        width=264.0,

        height=56.0

    )

    entry\_image\_3 = PhotoImage(

        file=relative\_to\_assets("entry\_3.png"))

    entry\_bg\_3 = canvas.create\_image(

        670.0,

        264.0,

        image=entry\_image\_3

    )

    entry\_3 = Text(

        bd=0,

        bg="#B5CEF5",

        fg="#000716",

        highlightthickness=0,

        font=("Inter", 30 \* -1)

    )

    entry\_3.place(

        x=538.0,

        y=235.0,

        width=264.0,

        height=56.0

    )

    entry\_image\_4 = PhotoImage(

        file=relative\_to\_assets("entry\_4.png"))

    entry\_bg\_4 = canvas.create\_image(

        244.0,

        378.0,

        image=entry\_image\_4

    )

    entry\_4 = Text(

        bd=0,

        bg="#B5CEF5",

        fg="#000716",

        highlightthickness=0,

        font=("Inter", 30 \* -1)

    )

    entry\_4.place(

        x=112.0,

        y=349.0,

        width=264.0,

        height=56.0

    )

    entry\_image\_5 = PhotoImage(

        file=relative\_to\_assets("entry\_5.png"))

    entry\_bg\_5 = canvas.create\_image(

        241.0,

        264.0,

        image=entry\_image\_5

    )

    entry\_5 = Text(

        bd=0,

        bg="#B5CEF5",

        fg="#000716",

        highlightthickness=0,

        font=("Inter", 30 \* -1)

    )

    entry\_5.place(

        x=109.0,

        y=235.0,

        width=264.0,

        height=56.0

    )

    entry\_image\_6 = PhotoImage(

        file=relative\_to\_assets("entry\_6.png"))

    entry\_bg\_6 = canvas.create\_image(

        244.0,

        150.0,

        image=entry\_image\_6

    )

    entry\_6 = Text(

        bd=0,

        bg="#B5CEF5",

        fg="#000716",

        highlightthickness=0,

        font=("Inter", 30 \* -1)

    )

    entry\_6.place(

        x=112.0,

        y=121.0,

        width=264.0,

        height=56.0

    )

    canvas.create\_text(

        104.0,

        76.0,

        anchor="nw",

        text="Фамилия",

        fill="#312C4D",

        font=("Inter", 30 \* -1)

    )

    canvas.create\_text(

        101.0,

        195.0,

        anchor="nw",

        text="Имя",

        fill="#312C4D",

        font=("Inter", 30 \* -1)

    )

    canvas.create\_text(

        101.0,

        309.0,

        anchor="nw",

        text="Отчество",

        fill="#312C4D",

        font=("Inter", 30 \* -1)

    )

    button\_image\_1 = PhotoImage(

        file=relative\_to\_assets("button\_1.png"))

    button\_1 = Button(

        image=button\_image\_1,

        borderwidth=0,

        highlightthickness=0,

        command=lambda: print("button\_1 clicked"),

        relief="flat"

    )

    button\_1.place(

        x=101.0,

        y=463.0,

        width=280.0,

        height=74.0

    )

    button\_image\_2 = PhotoImage(

        file=relative\_to\_assets("button\_2.png"))

    button\_2 = Button(

        image=button\_image\_2,

        borderwidth=0,

        highlightthickness=0,

        command=lambda: print("button\_2 clicked"),

        relief="flat"

    )

    button\_2.place(

        x=513.0,

        y=463.0,

        width=314.0,

        height=74.0

    )

    canvas.create\_text(

        530.0,

        195.0,

        anchor="nw",

        text="Логин",

        fill="#312C4D",

        font=("Inter", 30 \* -1)

    )

    canvas.create\_text(

        530.0,

        76.0,

        anchor="nw",

        text="Email",

        fill="#312C4D",

        font=("Inter", 30 \* -1)

    )

    canvas.create\_text(

        530.0,

        309.0,

        anchor="nw",

        text="Пароль",

        fill="#312C4D",

        font=("Inter", 30 \* -1)

    )

    canvas.create\_text(

        19.0,

        10.0,

        anchor="nw",

        text="Регистрация",

        fill="#312C4D",

        font=("Inter", 36 \* -1)

    )

    registration.resizable(False, False)

    registration.mainloop()