1. **Организационные вопросы оформления на предприятии, инструктаж по охране труда и технике безопасности, распределение по рабочим местам.**

Инструктаж по охране труда и технике безопасности на предприятии УЭТК СГУ на работе за компьютером проводится с целью обеспечения безопасности и здоровья работников при работе на компьютере, а так же с целью предотвращения производственных травм и заболеваний, связанных с работой за компьютером.

Инструктаж помогает работникам понимать как правильно использовать компьютерное оборудование и какие меры необходимо принимать для предотвращения возможных опасностей, связанных с работой за компьютером. В ходе инструктажа рассматриваются основные правила и требования, касающихся безопасности и здоровья при работе за компьютером, а так же обучаются методам работы за компьютером, которые позволяют уменьшить риск производственных травм и заболеваний.

Инструктаж по охране труда и технике безопасности на работе за ПК:

* Рабочее место. Правильная организация рабочего места играет важную роль в предотвращении заболеваний связанных с работой за компьютером. Необходимо сидеть прямо, регулировать высоту стула, рукоплечевой сустав должен быть параллелен столу.
* Монитор. Расстояние между глазами и монитором должно быть не менее 50-70 см. Яркость и контрастность монитора должны быть настроены на уровне, приятном для глаз.
* Освещение. Хорошее освещение помогает предотвратить усталость глаз и головные боли. Настольная лампа поможет поддерживать оптимальное освещение на рабочем месте.
* Перерывы. Регулярные перерывы помогают предотвратить усталость и повысить эффективность работы. Рекомендуется делать перерывы каждый час, на 10-15 минут.
* Упражнения для глаз и шеи. Регулярные упражнения для глаз и шеи помогают предотвратить боли и напряжение в мышцах.
* Клавиатура и мышь. Правильная позиция рук и запястий при работе с клавиатурой и мышью помогает предотвратить травмы. Руки и запястья должны быть расположены на уровне локтей.
* Защита глаз. Если вы работаете за компьютером в течение длительного времени, наденьте защитные очки, чтобы предотвратить усталость глаз и сухость.
* Безопасность. Никогда нельзя отключать заземление на компьютере. Это может привести к поражению электрическим током.

Это основные правила, которые необходимо соблюдать при работе за компьютером в университетском экономико-технологическом колледже Сочинского государственного университета

1. **Ознакомление со структурой и характером деятельности предприятия.**

Университетский экономико-технологический колледж Сочинского государственного университета является структурным подразделением университета, который специализируется на подготовке специалистов в области экономики и технологий. Колледж имеет несколько факультетов, включая факультеты экономики, менеджмента, информатики и технологий.

Колледж предлагает своим студентам обширную базу для изучения, включая современные компьютерные классы, специализированные лаборатории, библиотеку и многие другие ресурсы.

В рамках своей деятельности колледж проводит различные мероприятия, направленные на подготовку и развитие студентов, включая семинары, тренинги, конференции и практики в ведущих компаниях региона.

Кроме того, колледж активно сотрудничает с работодателями и предоставляет своим студентам возможности для стажировки и трудоустройства после окончания обучения. Все это позволяет колледжу успешно выпускать высококвалифицированных специалистов, которые востребованы на рынке труда.

В колледже функционирует более 50 лабораторий и учебных кабинетов, оснащенных современным оборудованием: лаборатория информационно-коммуникационных систем, делопроизводства и оргтехники, служба приема и размещения гостей, товароведения продовольственных товаров и продукции общественного питания, учебная бухгалтерия, учебный банк, технологии разработки баз данных, системного программирования, учебный кулинарный цех, учебный кондитерский цех, другие лаборатории. Учебные кабинеты оснащены интерактивными досками, проекторами, комплектами наглядных пособий.

Важным шагом в последовательном повышении качества образования является открытие и успешная аккредитация специализированного центра WorldSkills по компонентам «Поварское дело» (далее – СЦК). Аккредитация Союзом WorldSkills СЦК подтверждает наличие в экономико-технологическом колледже СГУ современного оборудования и технологий, отвечающих требованиях WorldSkills, а так же наличие экспертов для осуществления обучения и оценки соответствующих квалификаций по стандартам WorldSkills. В своей структуре колледж имеет три отделения: отделение туризма и сервиса, экономика и права, техническое.

1. **Сбор материалов для составления технического задания по теме дипломной работы**
   1. **Анализ деятельности работы архива в рамках архивации документов УЭТК СГУ**

Архив УЭТК СГУ выполняет функцию хранения и обеспечения доступа к документам, необходимым для деятельности колледжа. Для архива установлены определенные правила и процедуры, регламентирующие его работу.

Одним из главных направлений работы архива является сбор и хранение документов, относящихся к истории колледжа, его учебной, научной и общественной деятельности. В архиве также хранятся документы, связанные с трудовой и финансовой деятельностью колледжа.

Для обеспечения более удобного доступа к документам архив производит их каталогизацию и индексацию. При этом учитываются не только формальные атрибуты документов (название, автор, дата и т.д.), но и их содержание.

Для работы архива используются различные технические средства, включая специальное программное обеспечение для автоматизации процессов хранения и обработки документов. Однако, в настоящее время система учета архивных фондов и документов не является полностью автоматизированной и требует дополнительной работы по совершенствованию.

Также важным аспектом работы архива является его охрана и безопасность. Для этого в архиве установлены системы контроля доступа, а также правила по обращению с документами, в том числе их хранения и уничтожения.

В целом, архив УЭТК СГУ функционирует достаточно эффективно, однако есть возможности для совершенствования и оптимизации процессов работы, в том числе через внедрение новых технических решений.

**3.2** **Анализ программных средств, необходимых для разработки приложения для автоматизации процессов хранения в архиве УЭТК СГУ.**

Для разработки приложения для автоматизации процессов хранения в архиве УЭТК СГУ необходимо провести анализ программных средств, которые позволят реализовать задачи проекта.

Одним из основных требований является возможность хранения и управления базами данных, содержащими информацию об архивных документах. Для этих целей можно использовать СУБД (систему управления базами данных), такие как MySQL, PostgreSQL, MS SQL Server и др. Они обладают широким функционалом, поддерживают стандарт SQL и предоставляют средства для резервного копирования и восстановления данных, а также защиты базы данных от несанкционированного доступа.

Для разработки приложения для автоматизации процессов хранения в архиве могут использоваться различные языки программирования, такие как Java, Python, C#, Ruby и другие. В зависимости от выбора языка программирования, можно использовать средства разработки, такие как Visual Studio, PyCharm и др. Они предоставляют инструменты для разработки, отладки и тестирования приложений.

Также важно обратить внимание на использование средств версионного контроля, например, Git, SVN, Mercurial. Это позволит отслеживать изменения в исходном коде приложения, управлять версиями приложения и быстро возвращаться к предыдущим версиям.

Важным аспектом является также выбор средств для тестирования приложения, например, JUnit для Java, PyTest для Python, NUnit для C# и другие. Это позволяет проверять корректность работы приложения, выявлять ошибки и улучшать качество кода.

* 1. **Описание этапов разработки подсистемы.**

Этапы разработки подсистемы приложения для автоматизации процессов хранения в архиве УЭТК СГУ могут быть описаны следующим образом:

Сбор и анализ требований пользователя. Этот этап включает в себя определение функциональных и нефункциональных требований, а также определение пользовательского интерфейса и взаимодействия системы с пользователем.

Проектирование системы. На этом этапе определяются архитектура системы, базы данных, выбор технологий и инструментов для разработки, а также диаграммы классов, последовательности и прочие схемы.

Разработка функционала системы. На этом этапе происходит написание кода для реализации функционала, который был определен на предыдущих этапах. Важно придерживаться принципов модульности и переиспользуемости кода.

Тестирование и отладка. Этот этап включает в себя тестирование системы на соответствие требованиям, проверку ее работоспособности, выявление и устранение ошибок и дефектов.

Внедрение и сопровождение. На этом этапе система запускается в рабочую среду и происходит ее внедрение. Также важно предусмотреть механизмы сопровождения системы, включая регулярное обновление и исправление ошибок.

Обучение пользователей. Этот этап включает обучение пользователей работе с системой и предоставление документации, необходимой для использования системы.

Каждый этап должен быть детально продуман и тщательно выполнен, чтобы в итоге получить качественную и надежную систему автоматизации процессов хранения в архиве УЭТК СГУ.

Поскольку приложение будет включать в себя авторизацию и показ данных необходимо создать базу данных для хранения информации об авторизации пользователей, а также всю информацию о мероприятиях. Для этого была разработана ER-диаграмма для будущего приложения. На рисунке 2 представлена ER-диаграмма базы данных.

3.3.1. Макет программного модуля

В результате проведенного анализа деятельности УЭТК СГУ было принято решение о проектировании и разработке программного обеспечения, для автоматизации процессов хранения в архиве в данном учебном учреждении.

Сначала нужно выбрать и описать тип архитектуры программного обеспечения.

Были разработаны следующие схемы в представленном порядке:

- Функциональная схема;

- Диаграмма прецедентов;

- Схема переходов

- WireFrame

- ER-диаграмма базы данных

3.3.2. Функциональная схема

На функциональной схеме показано, какие функции может выполнять программное обеспечение, и какие роли могут быть у пол

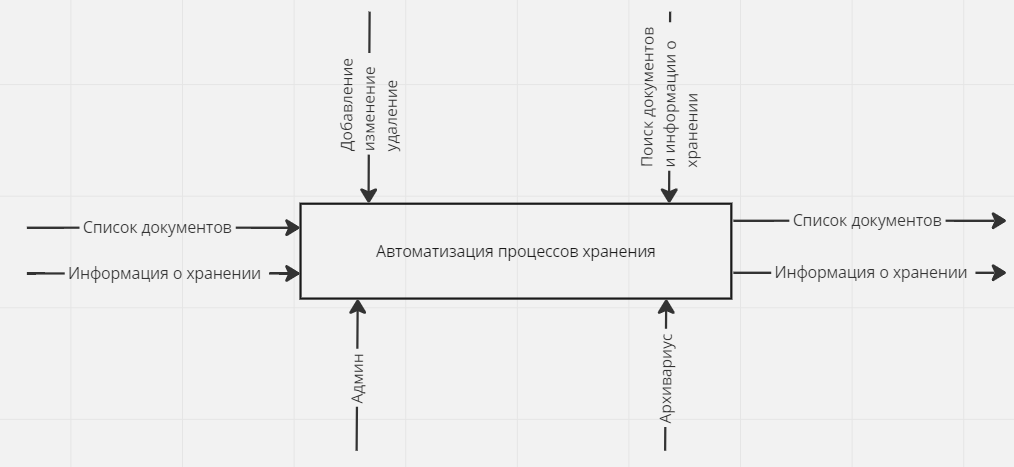


Рис. 1 Функциональная схема модуля автоматизации процессов хранения

Разработка функциональной схемы приложения (информационной системы) происходит согласно руководящему документу по стандартизации РД 50-34.698-90 «Автоматизированные системы требования к содержанию документов». В требованиях данного документа описаны нормы и правила при разработке функциональных схем.

В базе данных программного обеспечения должны храниться списки документов, хранящихся в архиве колледжа.

Роли, которые могут быть выданы пользователю, представлены: Администратором и Архивариусом.

Администратор может видеть все данные, хранящиеся в базе данных.

Архивариус может видеть только те данные, которые выводятся в приложении.

Информация о списках документов и данных о их хранении являются как входящей, так и выходящей в программном модуле. Внутри системы должны выполняться такие функции как: добавление новых документов и информации о их хранении, изменение существующей информации о документах, поиск документов.

3.3.3. Диаграмма прецедентов

Далее была создана диаграмма прецедентов, в которой представлены все доступные функции для каждой роли.

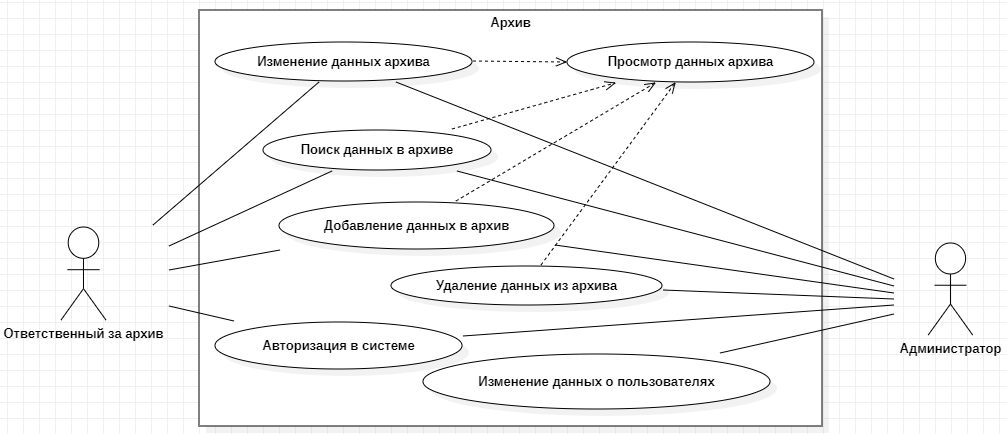


Рис. 2 Диаграмма прецедентов

Прецеденты (use case) для каждого субъекта представлены в виде структурных элементов (предоставленных графически в виде овалов) с названием прецедента для каждого субъекта.

В разрабатываемой системе каждый из субъектов имеет следующие прецеденты:

- субъект «Ответственный за архив» - прецеденты «Добавление данных в архив», «Изменение данных архива», «Поиск данных в архиве», «Авторизация в системе», «Просмотр данных архива»

- субъект «Администратор» - прецеденты «Добавление данных в архив», «Изменение данных архива», «Поиск данных в архиве», «Авторизация в системе» «Просмотр данных архива», «Удаление данных архива», «Изменение данных о пользователях»

Доступ к базе данных есть только у администратора. Для безопасности данных добавление нового пользователя происходит исключительно вводом данных в базу данных через SQL Server Management Studio.

3.3.4. WireFrame

Перед работой над приложением так же необходимо понять, как должен выглядеть программный модуль, какие окна в нём должны быть и какие элементы. Для этого был создан макет программного модуля

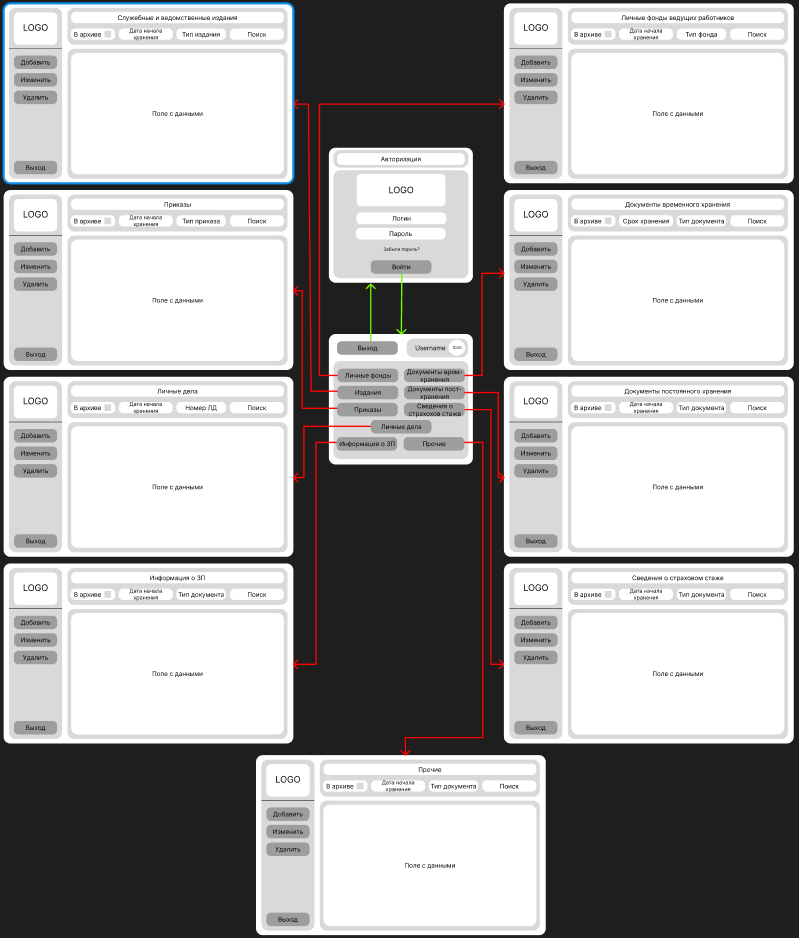


Рис. 4 Wireframe – макет приложения

На рисунке 4 показаны окна, которые должны быть реализованы в приложении:

– Авторизация;

– Главное меню

– Окно «Документы временного хранения»

– Окно «Документы постоянного хранения»

– Окно «Личные фонды ведущих работников»

– Окно «Служебные и ведомственные издания»

– Окно «Приказы»

– Окно «Личные дела»

– Окно «Информация о ЗП»

– Окно «Сведения о страховом стаже»

– Окно «Прочие»

3.3.5. Макет базы данных

Далее был сделан макет базы данных

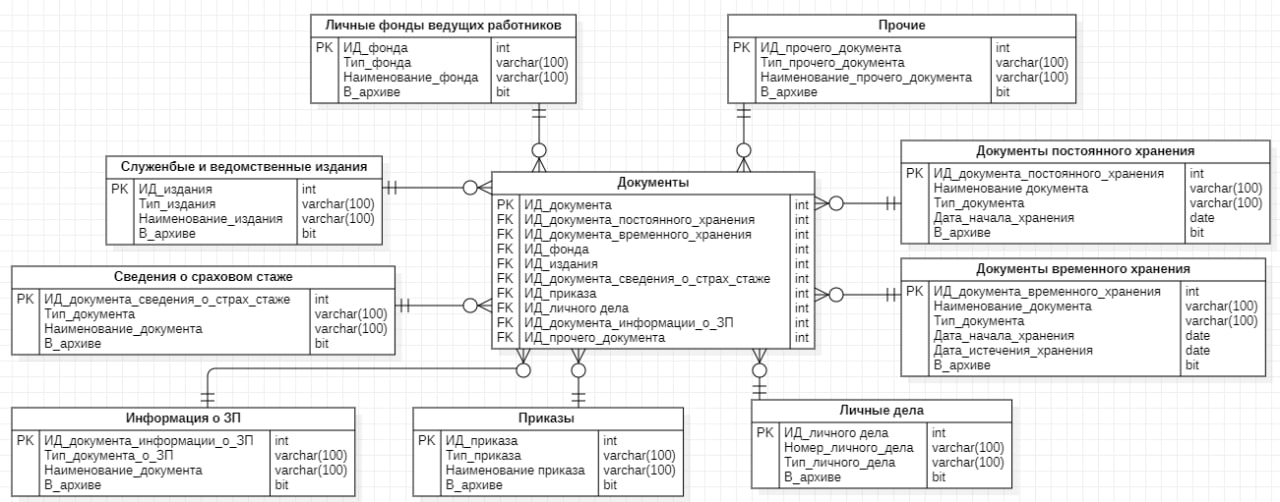


Рис. 5 Макет базы данных

3.3.6. Словарь данных

Для макета базы данных был так же создан словарь данных:

Таблица 1 Словарь данных

|  |  |  |  |
| --- | --- | --- | --- |
| Словарь данных | | | |
| Ключ | Название | Тип данных | Описание |
| Документы | | | |
| PK | ИД\_документа | int | ИД документа |
| FK | ИД\_документа\_постоянного\_хранения | int | ИД документа постоянного хранения |
| FK | ИД\_документа\_временного\_хранения | int | ИД документа временного хранения |
| FK | ИД\_фонда | int | ИД фонда |
| FK | ИД\_издания | int | ИД издания |
| FK | ИД\_документа\_сведения\_о\_страх\_стаже | int | ИД документа сведения о страх стаже |
| FK | ИД\_приказа | int | ИД приказа |
| FK | ИД\_личного\_дела | int | ИД личного дела |
| FK | ИД\_документа\_информации\_о\_ЗП | int | ИД документа информации о ЗП |
| FK | ИД\_прочего\_документа | int | ИД прочего документа |
| Документы постоянного хранения | | | |
| PK | ИД\_документа\_постоянного\_хранения | int | ИД документа постоянного хранения |
|  | Наименование\_документа | varchar | Наименование документа |
|  | Тип\_документа | varchar | Тип документа |
|  | Дата\_начала\_хранения | date | Дата начала хранения |
|  | В\_архиве | bit | В архиве |
| Документы временного хранения | | | |
| PK | ИД\_документа\_временного\_хранения | int | ИД документа временного хранения |
|  | Наименование\_документа | varchar | Наименование документа |
|  | Тип\_документа | varchar | Тип документа |
|  | Дата\_начала\_хранения | date | Дата начала хранения |
|  | Дата\_окончания\_хранения | date | Дата окончания хранения |
|  | В\_архиве | bit | В архиве |
| Личные дела | | | |
| PK | ИД\_личного\_дела | int | ИД личного дела |

Продолжение таблицы 1

|  |  |  |  |
| --- | --- | --- | --- |
|  | Номер\_личного\_дела | varchar | Номер личного дела |
|  | Тип\_личного\_дела | varchar | Тип личного дела |
|  | В\_архиве | bit | В архиве |
| Приказы | | | |
| PK | ИД\_приказа | int | ИД приказа |
|  | Тип\_приказа | varchar | Тип приказа |
|  | Наименование\_приказа | varchar | Наименование приказа |
|  | В\_архиве | bit | В архиве |
| Информация о ЗП | | | |
| PK | ИД\_документа\_информации\_о\_ЗП | int | ИД документа информации о ЗП |
|  | Тип\_документа\_о\_ЗП | varchar | Тип документа о ЗП |
|  | Наименование\_документа | varchar | Наименование документа |
|  | В\_архиве | bit | В архиве |
| Сведения о страховом стаже | | | |
| PK | ИД\_документа\_сведения\_о\_страх\_стаже | int | ИД документа сведения о страх стаже |
|  | Тип\_документа | varchar | Тип документа |
|  | Наименование\_документа | varchar | Наименование документа |
|  | В\_архиве | bit | В архиве |
| Служебные и ведомтсвенные издания | | | |
| PK | ИД\_издания | int | ИД издания |
|  | Тип\_издания | varchar | Тип издания |
|  | Наименование\_издания | varchar | Наименование издания |
|  | В\_архиве | bit | В архиве |
| Личные фонды ведущих работников | | | |
| PK | ИД\_фонда | int | ИД фонда |
|  | Тип\_фонда | varchar | Тип фонда |
|  | Наименование\_фонда | varchar | Наименование фонда |
|  | В\_архиве | bit | В архиве |
| Прочие | | | |
| PK | ИД\_прочего\_документа | int | ИД прочего документа |
|  | Тип\_прочего\_документа | varchar | Тип прочего документа |
|  | Наименование\_прочего\_документа | varchar | Наименование прочего документа |
|  | В\_архиве | bit | В архиве |

3.3.7. Разработка программного модуля

Работа над программным модулем началась с окна авторизации. Сначала были расставлены все элементы окна.

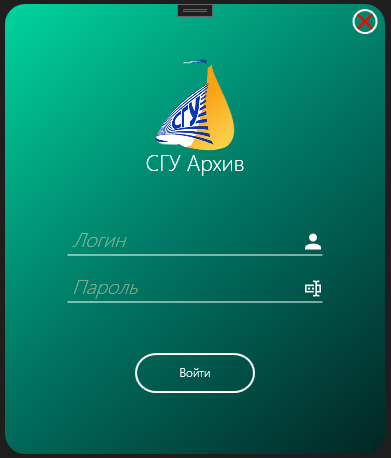


Рис. 6 Окно «Авторизация»

Далее была написан функционал окна.

Листинг 1 Код окна «Авторизация»

|  |
| --- |
| Database.ArchiveEntities dbcon;  public MainWindow()  {  InitializeComponent();  dbcon = new Database.ArchiveEntities();  } |

Продолжение листинга 1

|  |
| --- |
| private void Open\_Click(object sender, RoutedEventArgs e)  {  if ((Login.Text == "") || (Password.Password == ""))  {  MessageBox.Show("Поля не должны быть пустыми!", "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);  return;  }  else  {  string sql = "select Логин,Пароль " + "from Авторизация" + " where Логин like '" + Login.Text + "' and Пароль like '" + Password.Password + "'";  SqlConnection connection = new SqlConnection(@"data source=DESKTOP-E87MVPS;initial catalog=Archive;integrated security=True;MultipleActiveResultSets=True;App=EntityFramework");  connection.Open();  DataTable dt = new DataTable();  SqlCommand command = new SqlCommand(sql, connection);  dt.Load(command.ExecuteReader());  connection.Close();  if (dt.Rows.Count > 0)  {  Windows.AuthorizationAndMenu.Menu con = new Windows.AuthorizationAndMenu.Menu();  con.Show();  this.Close();  }  else  {  MessageBox.Show("Вы неправильно ввели логин или пароль!", "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);  return;  }  }  }  private void Close\_Click(object sender, RoutedEventArgs e)  {  this.Close();  }  private void Window\_MouseDown(object sender, MouseButtonEventArgs e)  {  if (e.LeftButton == MouseButtonState.Pressed)  {  DragMove();  }  } |

Далее было сделано окно «Главное меню»

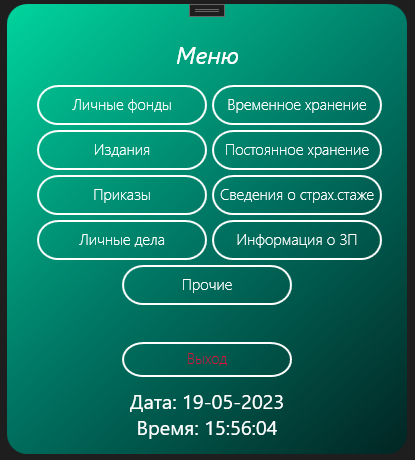


Рис. 7 Окно «Главное меню»

После окончания работы над внешним видом окна, был написан функционал окна.

Листинг 2 Код окна «Главное меню»

|  |
| --- |
| private void Window\_MouseDown(object sender, MouseButtonEventArgs e)  {  if (e.LeftButton == MouseButtonState.Pressed)  {  DragMove();  }  } |

Продолжение листинга 2

|  |
| --- |
| private void Window\_Loaded(object sender, RoutedEventArgs e)  {  var timer = new System.Windows.Threading.DispatcherTimer();  timer.Interval = new TimeSpan(0, 0, 1);  timer.IsEnabled = true;  timer.Tick += (o, t) => { Date.Content = DateTime.Now.ToString("Дата: dd-MM-yyyy"); };  timer.Tick += (o, t) => { Time.Content = DateTime.Now.ToString("Время: HH:mm:ss"); };  timer.Start();  }  private void Temporary\_storage\_Click(object sender, RoutedEventArgs e)  {  TemporaryStorage.TemporaryStorageW con = new TemporaryStorage.TemporaryStorageW();  con.Show();  }  private void Permanent\_storage\_Click(object sender, RoutedEventArgs e)  {  PermanentStorage.PermanentStorageW con = new PermanentStorage.PermanentStorageW();  con.Show();  }  private void Editions\_Click(object sender, RoutedEventArgs e)  {  Editions.EditionsW con = new Editions.EditionsW();  con.Show();  }  private void Persona\_funds\_Click(object sender, RoutedEventArgs e)  {  PersonalFunds.PersonalFundsW con = new PersonalFunds.PersonalFundsW();  con.Show();  }  private void Orders\_Click(object sender, RoutedEventArgs e)  {  Orders.OrdersW con = new Orders.OrdersW();  con.Show();  }  private void Others\_Click(object sender, RoutedEventArgs e)  {  Others.OthersW con = new Others.OthersW();  con.Show();  }  private void Wages\_Click(object sender, RoutedEventArgs e)  {  Wages.WagesW con = new Wages.WagesW();  con.Show();  } |

Продолжение листинга 2

|  |
| --- |
| private void Personal\_affairs\_Click(object sender, RoutedEventArgs e)  {  PersonalAffairs.PersonalAffairsW con = new PersonalAffairs.PersonalAffairsW();  con.Show();  }  private void Insurance\_experience\_Click(object sender, RoutedEventArgs e)  {  InsuranceExperience.InsuranceExperienceW con = new InsuranceExperience.InsuranceExperienceW();  con.Show();  }  private void Close\_Click(object sender, RoutedEventArgs e)  {  MainWindow con = new MainWindow();  con.Show();  this.Close();  } |

Далее были сделаны окна «Документы временного хранения», «Документы постоянного хранения», «Личные фонды ведущих работников», «Служебные и ведомственные издания», «Приказы», «Личные дела», «Информация о ЗП», «Сведения о страховом стаже», «Прочие»

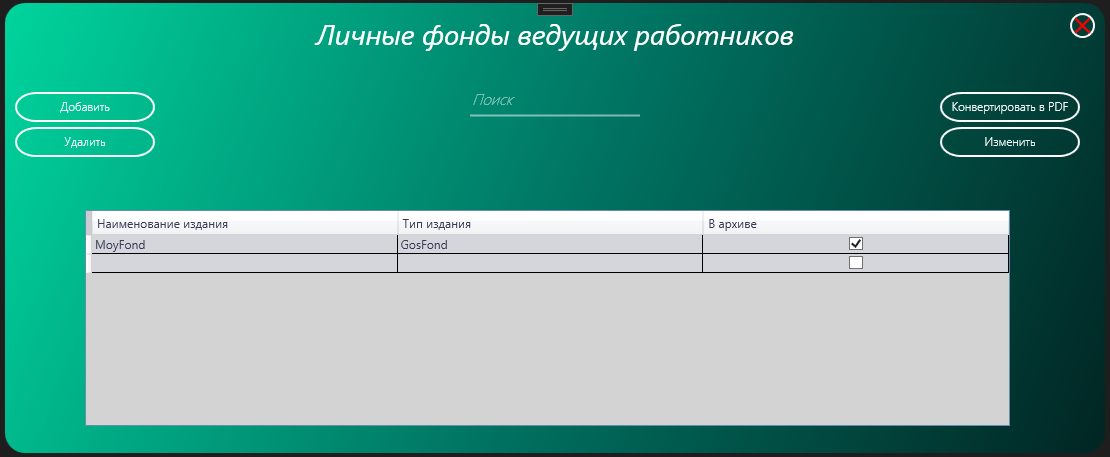


Рис. 8 Окно «Личные фонды ведущих работников»

После окончания работы над внешним видом окна, был написан функционал окна.

Листинг 3 Код окна «Личные фонды ведущих работников»

|  |
| --- |
| public PersonalFundsW()  {  InitializeComponent();  dbcon = new Database.ArchiveEntities();  DTTFunds.ItemsSource = dbcon.Личные\_фонды\_ведущих\_работников.ToList();  DTGFunds.ItemsSource = dbcon.Личные\_фонды\_ведущих\_работников.ToList();  }  private void TextBoxSearch\_KeyUp(object sender, KeyEventArgs e)  {  var filtered = dbcon.Личные\_фонды\_ведущих\_работников.Where(k => k.Наименование\_фонда.ToString().Contains(TextBoxSearch.Text)  || k.Тип\_фонда.ToString().Contains(TextBoxSearch.Text)  || k.В\_архиве.ToString().Contains(TextBoxSearch.Text)).ToList();  DTGFunds.ItemsSource = filtered;  }  private void Window\_MouseDown(object sender, MouseButtonEventArgs e)  {  if (e.LeftButton == MouseButtonState.Pressed)  {  DragMove();  }  }  private void AddDocument\_Click(object sender, RoutedEventArgs e)  {  AddPersonalFundsW con = new AddPersonalFundsW();  con.Show();  }  private void Delete\_Click(object sender, RoutedEventArgs e)  {  var a = DTGFunds.SelectedItem as Личные\_фонды\_ведущих\_работников;  if (a == null)  {  MessageBox.Show("Выберите непустую строку!");  return;  }  MessageBoxResult result = MessageBox.Show("Вы действительно хотите удалить строку?", "Удалить?", MessageBoxButton.YesNo);  if (result == MessageBoxResult.Yes)  {  dbcon.Личные\_фонды\_ведущих\_работников.Remove(a);  dbcon.SaveChanges();  DTTFunds.ItemsSource = dbcon.Личные\_фонды\_ведущих\_работников.ToList();  DTGFunds.ItemsSource = dbcon.Личные\_фонды\_ведущих\_работников.ToList();  }  } |

Продолжение листинга 3

|  |
| --- |
| static void PdfHeaderFooterEventHandler(object sender, PdfHeaderFooterEventArgs e)  {  string date = DateTime.Now.ToString("dd:MM:yyy, hh:mm:ss");  PdfFont font = new PdfStandardFont(PdfFontFamily.TimesRoman, 20f, PdfFontStyle.Bold);  var width = e.PdfPage.GetClientSize().Width;  PdfPageTemplateElement header = new PdfPageTemplateElement(width, 38);  header.Graphics.DrawString("Students", font, PdfPens.Black, 70, 3);  e.PdfDocumentTemplate.Top = header;  PdfPageTemplateElement footer = new PdfPageTemplateElement(width, 38);  footer.Graphics.DrawString(date, font, PdfPens.Black, 70, 3);  e.PdfDocumentTemplate.OddBottom = footer;  }  private void ToPdf\_Click(object sender, RoutedEventArgs e)  {  PdfExportingOptions options = new PdfExportingOptions();  options.AutoColumnWidth = true;  options.PageHeaderFooterEventHandler = PdfHeaderFooterEventHandler;  var document = DTTFunds.ExportToPdf(options);  SaveFileDialog sfd = new SaveFileDialog  {  Filter = "PDF Files(\*.pdf)|\*.pdf"  };  if (sfd.ShowDialog() == true)  {  using (Stream stream = sfd.OpenFile())  {  document.Save(stream);  }  if (MessageBox.Show("Хотите открыть сохранённый PDF файл?", "Файл PDF создан.",  MessageBoxButton.YesNo, MessageBoxImage.Information) == MessageBoxResult.Yes)  {  System.Diagnostics.Process.Start(sfd.FileName);  }  }  document.Save("Sample.pdf");  }  private void Change\_Click(object sender, RoutedEventArgs e)  {  if (MessageBox.Show("Хотите хотите внести изменения?", "Изменения внесены.", MessageBoxButton.YesNo, MessageBoxImage.Information) == MessageBoxResult.Yes)  {  dbcon.SaveChanges();  }  }  private void Close\_Click(object sender, RoutedEventArgs e)  {  this.Close();  } |

Далее были сделано окно «Служебные и ведомственные издания»

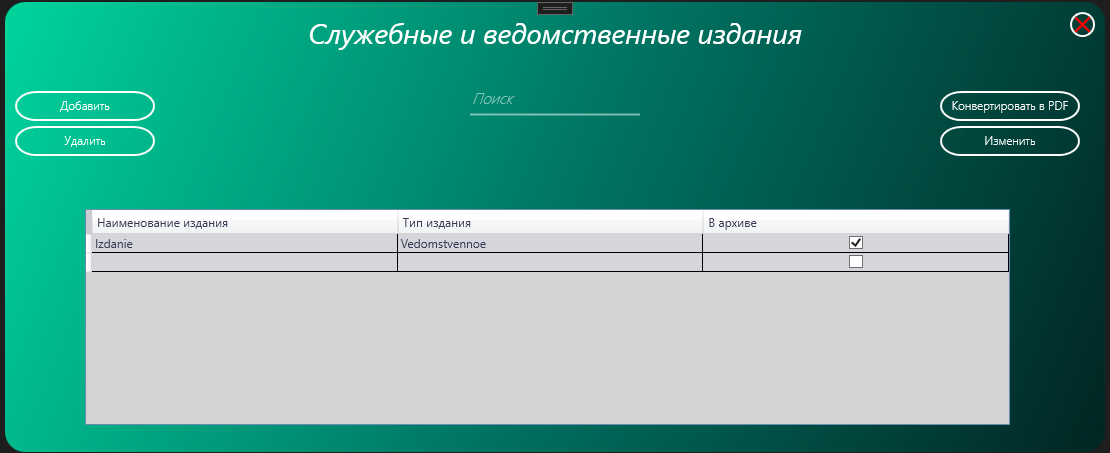


Рис. 9 Окно «Служебные и ведомственные издания»

После окончания работы над внешним видом окна, был написан функционал окна.

Листинг 4 Код окна «Служебные и ведомственные издания»

|  |
| --- |
| public EditionsW()  {  InitializeComponent();  dbcon = new Database.ArchiveEntities();  DTTEditions.ItemsSource = dbcon.Служебные\_и\_ведомственные\_издания.ToList();  DTGEditions.ItemsSource = dbcon.Служебные\_и\_ведомственные\_издания.ToList();  }  private void Window\_MouseDown(object sender, MouseButtonEventArgs e)  {  if (e.LeftButton == MouseButtonState.Pressed)  {  DragMove();  }  }  private void TextBoxSearch\_KeyUp(object sender, KeyEventArgs e)  {  var filtered = dbcon.Служебные\_и\_ведомственные\_издания.Where(k => k.Наименование\_издания.ToString().Contains(TextBoxSearch.Text)  || k.Тип\_издания.ToString().Contains(TextBoxSearch.Text)  || k.В\_архиве.ToString().Contains(TextBoxSearch.Text)).ToList();  DTGEditions.ItemsSource = filtered;  } |

Продолжение листинга 4

|  |
| --- |
| private void Delete\_Click(object sender, RoutedEventArgs e)  {  var a = DTGEditions.SelectedItem as Служебные\_и\_ведомственные\_издания;  if (a == null)  {  MessageBox.Show("Выберите непустую строку!");  return;  }  MessageBoxResult result = MessageBox.Show("Вы действительно хотите удалить строку?", "Удалить?", MessageBoxButton.YesNo);  if (result == MessageBoxResult.Yes)  {  dbcon.Служебные\_и\_ведомственные\_издания.Remove(a);  dbcon.SaveChanges();  DTTEditions.ItemsSource = dbcon.Служебные\_и\_ведомственные\_издания.ToList();  DTGEditions.ItemsSource = dbcon.Служебные\_и\_ведомственные\_издания.ToList();  }  }  static void PdfHeaderFooterEventHandler(object sender, PdfHeaderFooterEventArgs e)  {  string date = DateTime.Now.ToString("dd:MM:yyy, hh:mm:ss");  PdfFont font = new PdfStandardFont(PdfFontFamily.TimesRoman, 20f, PdfFontStyle.Bold);  var width = e.PdfPage.GetClientSize().Width;  PdfPageTemplateElement header = new PdfPageTemplateElement(width, 38);  header.Graphics.DrawString("Students", font, PdfPens.Black, 70, 3);  e.PdfDocumentTemplate.Top = header;  PdfPageTemplateElement footer = new PdfPageTemplateElement(width, 38);  footer.Graphics.DrawString(date, font, PdfPens.Black, 70, 3);  e.PdfDocumentTemplate.OddBottom = footer;  }  private void ToPdf\_Click(object sender, RoutedEventArgs e)  {  PdfExportingOptions options = new PdfExportingOptions();  options.AutoColumnWidth = true;  options.PageHeaderFooterEventHandler = PdfHeaderFooterEventHandler;  var document = DTTEditions.ExportToPdf(options);  SaveFileDialog sfd = new SaveFileDialog  {  Filter = "PDF Files(\*.pdf)|\*.pdf"  }; |

|  |
| --- |
| if (sfd.ShowDialog() == true)  {  using (Stream stream = sfd.OpenFile())  {  document.Save(stream);  }  if (MessageBox.Show("Хотите открыть сохранённый PDF файл?", "Файл PDF создан.",  MessageBoxButton.YesNo, MessageBoxImage.Information) == MessageBoxResult.Yes)  {  System.Diagnostics.Process.Start(sfd.FileName);  }  }  document.Save("Sample.pdf");  }  private void Change\_Click(object sender, RoutedEventArgs e)  {  if (MessageBox.Show("Хотите хотите внести изменения?", "Изменения внесены.", MessageBoxButton.YesNo, MessageBoxImage.Information) == MessageBoxResult.Yes)  {  dbcon.SaveChanges();  }  }  private void Close\_Click(object sender, RoutedEventArgs e)  {  this.Close();  } |

Продолжение листинга 4

Далее были сделано окно «Приказы»

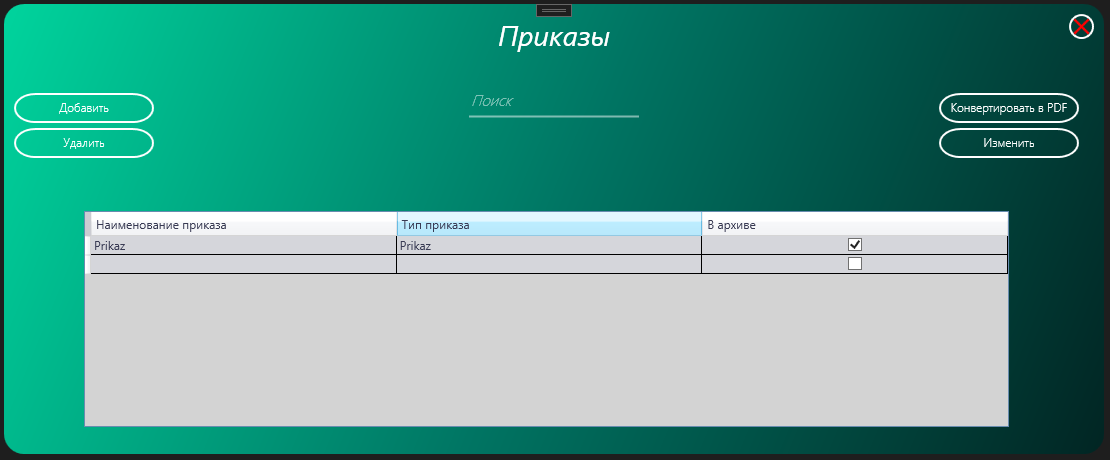


Рис. 10 Окно «Приказы»

После окончания работы над внешним видом окна, был написан функционал окна.

|  |
| --- |
| public OrdersW()  {  InitializeComponent();  dbcon = new Database.ArchiveEntities();  DTTOrders.ItemsSource = dbcon.Приказы.ToList();  DTGOrders.ItemsSource = dbcon.Приказы.ToList();  }  private void Window\_MouseDown(object sender, MouseButtonEventArgs e)  {  if (e.LeftButton == MouseButtonState.Pressed)  {  DragMove();  }  }  private void TextBoxSearch\_KeyUp(object sender, KeyEventArgs e)  {  var filtered = dbcon.Приказы.Where(k => k.Наименование\_приказа.ToString().Contains(TextBoxSearch.Text)  || k.Тип\_приказа.ToString().Contains(TextBoxSearch.Text)  || k.В\_архиве.ToString().Contains(TextBoxSearch.Text)).ToList();  DTGOrders.ItemsSource = filtered;  }  private void AddDocument\_Click(object sender, RoutedEventArgs e)  {  AddOrdersW con = new AddOrdersW();  con.Show();  }  private void Delete\_Click(object sender, RoutedEventArgs e)  {  var a = DTGOrders.SelectedItem as Приказы;  if (a == null)  {  MessageBox.Show("Выберите непустую строку!");  return;  }  MessageBoxResult result = MessageBox.Show("Вы действительно хотите удалить строку?", "Удалить?", MessageBoxButton.YesNo);  if (result == MessageBoxResult.Yes)  {  dbcon.Приказы.Remove(a);  dbcon.SaveChanges();  DTTOrders.ItemsSource = dbcon.Документы\_постоянного\_хранения.ToList();  DTGOrders.ItemsSource = dbcon.Документы\_постоянного\_хранения.ToList();  }  } |

Листинг 5 Код окна «Приказы»

|  |
| --- |
| static void PdfHeaderFooterEventHandler(object sender, PdfHeaderFooterEventArgs e)  {  string date = DateTime.Now.ToString("dd:MM:yyy, hh:mm:ss");  PdfFont font = new PdfStandardFont(PdfFontFamily.TimesRoman, 20f, PdfFontStyle.Bold);  var width = e.PdfPage.GetClientSize().Width;  PdfPageTemplateElement header = new PdfPageTemplateElement(width, 38);  header.Graphics.DrawString("Students", font, PdfPens.Black, 70, 3);  e.PdfDocumentTemplate.Top = header;  PdfPageTemplateElement footer = new PdfPageTemplateElement(width, 38);  footer.Graphics.DrawString(date, font, PdfPens.Black, 70, 3);  e.PdfDocumentTemplate.OddBottom = footer;  }  private void ToPdf\_Click(object sender, RoutedEventArgs e)  {  PdfExportingOptions options = new PdfExportingOptions();  options.AutoColumnWidth = true;  options.PageHeaderFooterEventHandler = PdfHeaderFooterEventHandler;  var document = DTTOrders.ExportToPdf(options);  SaveFileDialog sfd = new SaveFileDialog  {  Filter = "PDF Files(\*.pdf)|\*.pdf"  };  if (sfd.ShowDialog() == true)  {  using (Stream stream = sfd.OpenFile())  {  document.Save(stream);  }  if (MessageBox.Show("Хотите открыть сохранённый PDF файл?", "Файл PDF создан.",  MessageBoxButton.YesNo, MessageBoxImage.Information) == MessageBoxResult.Yes)  {  System.Diagnostics.Process.Start(sfd.FileName);  }  }  document.Save("Sample.pdf");  }  private void Change\_Click(object sender, RoutedEventArgs e)  {  if (MessageBox.Show("Хотите хотите внести изменения?", "Изменения внесены.", MessageBoxButton.YesNo, MessageBoxImage.Information) == MessageBoxResult.Yes)  {  dbcon.SaveChanges();  }  }  private void Close\_Click(object sender, RoutedEventArgs e)  {  this.Close();  } |

Продолжение листинга 5

Далее были сделано окно «Личные дела»

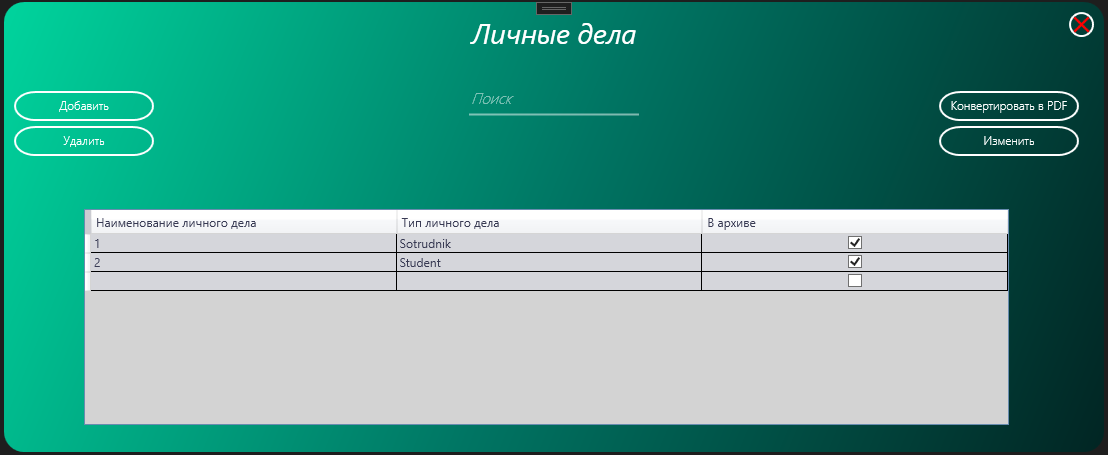


Рис. 11 Окно «Приказы»

После окончания работы над внешним видом окна, был написан функционал окна.

|  |
| --- |
| private void Window\_MouseDown(object sender, MouseButtonEventArgs e)  {  if (e.LeftButton == MouseButtonState.Pressed)  {  DragMove();  }  }  private void TextBoxSearch\_KeyUp(object sender, KeyEventArgs e)  {  var filtered = dbcon.Личные\_дела.Where(k => k.Номер\_личного\_дела.ToString().Contains(TextBoxSearch.Text)  || k.Тип\_личного\_дела.ToString().Contains(TextBoxSearch.Text)  || k.В\_архиве.ToString().Contains(TextBoxSearch.Text)).ToList();  DTGPersonalAffairs.ItemsSource = filtered;  }  private void AddDocument\_Click(object sender, RoutedEventArgs e)  {  AddPersonalAffairsW con = new AddPersonalAffairsW();  con.Show();  }  private void Delete\_Click(object sender, RoutedEventArgs e)  {  var a = DTGPersonalAffairs.SelectedItem as Личные\_дела;  if (a == null)  {  MessageBox.Show("Выберите непустую строку!");  return; } |

Листинг 6 Код окна «Личные дела»

|  |
| --- |
| MessageBoxResult result = MessageBox.Show("Вы действительно хотите удалить строку?", "Удалить?", MessageBoxButton.YesNo);  if (result == MessageBoxResult.Yes)  {  dbcon.Личные\_дела.Remove(a);  dbcon.SaveChanges();  DTTPersonalAffairs.ItemsSource = dbcon.Личные\_дела.ToList();  DTGPersonalAffairs.ItemsSource = dbcon.Личные\_дела.ToList();  }  }  static void PdfHeaderFooterEventHandler(object sender, PdfHeaderFooterEventArgs e)  {  string date = DateTime.Now.ToString("dd:MM:yyy, hh:mm:ss");  PdfFont font = new PdfStandardFont(PdfFontFamily.TimesRoman, 20f, PdfFontStyle.Bold);  var width = e.PdfPage.GetClientSize().Width;  PdfPageTemplateElement header = new PdfPageTemplateElement(width, 38);  header.Graphics.DrawString("Students", font, PdfPens.Black, 70, 3);  e.PdfDocumentTemplate.Top = header;  PdfPageTemplateElement footer = new PdfPageTemplateElement(width, 38);  footer.Graphics.DrawString(date, font, PdfPens.Black, 70, 3);  e.PdfDocumentTemplate.OddBottom = footer;  }  private void ToPdf\_Click(object sender, RoutedEventArgs e)  {  PdfExportingOptions options = new PdfExportingOptions();  options.AutoColumnWidth = true;  options.PageHeaderFooterEventHandler = PdfHeaderFooterEventHandler;  var document = DTTPersonalAffairs.ExportToPdf(options);  SaveFileDialog sfd = new SaveFileDialog  {  Filter = "PDF Files(\*.pdf)|\*.pdf"  };  if (sfd.ShowDialog() == true)  {  using (Stream stream = sfd.OpenFile())  {  document.Save(stream);  }  if (MessageBox.Show("Хотите открыть сохранённый PDF файл?", "Файл PDF создан.",  MessageBoxButton.YesNo, MessageBoxImage.Information) == MessageBoxResult.Yes)  {  System.Diagnostics.Process.Start(sfd.FileName);  }  }  document.Save("Sample.pdf");  } |

Продолжение листинга 6

|  |
| --- |
| private void Change\_Click(object sender, RoutedEventArgs e)  {  if (MessageBox.Show("Хотите хотите внести изменения?", "Изменения внесены.", MessageBoxButton.YesNo, MessageBoxImage.Information) == MessageBoxResult.Yes)  {  dbcon.SaveChanges();  }  }  private void Close\_Click(object sender, RoutedEventArgs e)  {  this.Close();  } |

Продолжение листинга 6

Далее были сделано окно «Документы временного хранения»

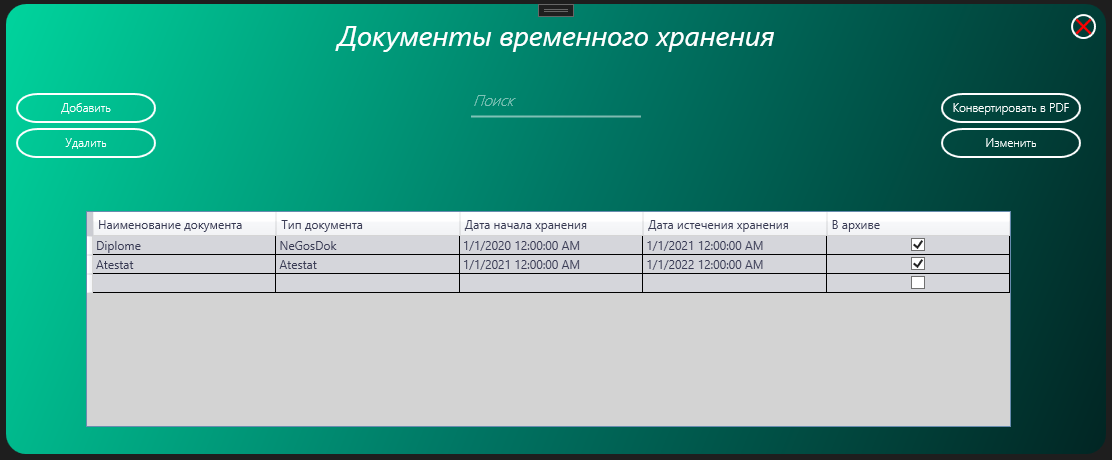


Рис 12. Окно «Документы временного хранения»

После окончания работы над внешним видом окна, был написан функционал окна.

Листинг 7 Код окна «Документы временного хранения»

|  |
| --- |
| private void Window\_MouseDown(object sender, MouseButtonEventArgs e)  {  if (e.LeftButton == MouseButtonState.Pressed)  {  DragMove();  } |

|  |
| --- |
| private void TextBoxSearch\_KeyUp(object sender, KeyEventArgs e)  {  var filtered = dbcon.Документы\_временного\_хранения.Where(k => k.Наименование\_документа.ToString().Contains(TextBoxSearch.Text)  || k.Тип\_документа.ToString().Contains(TextBoxSearch.Text)  || k.Дата\_начала\_хранения.ToString().Contains(TextBoxSearch.Text)  || k.Дата\_окончания\_хранения.ToString().Contains(TextBoxSearch.Text)  || k.В\_архиве.ToString().Contains(TextBoxSearch.Text)).ToList();  DTGTemporaryStorage.ItemsSource = filtered;  }  private void AddDocument\_Click(object sender, RoutedEventArgs e)  {  AddTemporaryStorageW con = new AddTemporaryStorageW();  con.Show();  }  private void Delete\_Click(object sender, RoutedEventArgs e)  {  var a = DTGTemporaryStorage.SelectedItem as Документы\_временного\_хранения;  if (a == null)  {  MessageBox.Show("Выберите непустую строку!");  return;  }  MessageBoxResult result = MessageBox.Show("Вы действительно хотите удалить строку?", "Удалить?", MessageBoxButton.YesNo);  if (result == MessageBoxResult.Yes)  {  dbcon.Документы\_временного\_хранения.Remove(a);  dbcon.SaveChanges();  DTTemporaryStorage.ItemsSource = dbcon.Документы\_временного\_хранения.ToList();  DTGTemporaryStorage.ItemsSource = dbcon.Документы\_временного\_хранения.ToList();  }  }  static void PdfHeaderFooterEventHandler(object sender, PdfHeaderFooterEventArgs e)  {  string date = DateTime.Now.ToString("dd:MM:yyy, hh:mm:ss");  PdfFont font = new PdfStandardFont(PdfFontFamily.TimesRoman, 20f, PdfFontStyle.Bold);  var width = e.PdfPage.GetClientSize().Width;  PdfPageTemplateElement header = new PdfPageTemplateElement(width, 38);  header.Graphics.DrawString("Students", font, PdfPens.Black, 70, 3);  e.PdfDocumentTemplate.Top = header;  PdfPageTemplateElement footer = new PdfPageTemplateElement(width, 38);  footer.Graphics.DrawString(date, font, PdfPens.Black, 70, 3);  e.PdfDocumentTemplate.OddBottom = footer;  } |

Продолжение листинга 7

Продолжение листинга 7

|  |
| --- |
| private void ToPdf\_Click(object sender, RoutedEventArgs e)  {  PdfExportingOptions options = new PdfExportingOptions();  options.AutoColumnWidth = true;  options.PageHeaderFooterEventHandler = PdfHeaderFooterEventHandler;  var document = DTTemporaryStorage.ExportToPdf(options);  SaveFileDialog sfd = new SaveFileDialog  {  Filter = "PDF Files(\*.pdf)|\*.pdf"  };  if (sfd.ShowDialog() == true)  {  using (Stream stream = sfd.OpenFile())  {  document.Save(stream);  }  if (MessageBox.Show("Хотите открыть сохранённый PDF файл?", "Файл PDF создан.",  MessageBoxButton.YesNo, MessageBoxImage.Information) == MessageBoxResult.Yes)  {  System.Diagnostics.Process.Start(sfd.FileName);  }  }  document.Save("Sample.pdf");  }  private void Change\_Click(object sender, RoutedEventArgs e)  {  if (MessageBox.Show("Хотите хотите внести изменения?", "Изменения внесены.", MessageBoxButton.YesNo, MessageBoxImage.Information) == MessageBoxResult.Yes)  {  dbcon.SaveChanges();  }  }  private void Close\_Click(object sender, RoutedEventArgs e)  {  this.Close();  } |

Далее были сделано окно «Документы постоянного хранения»

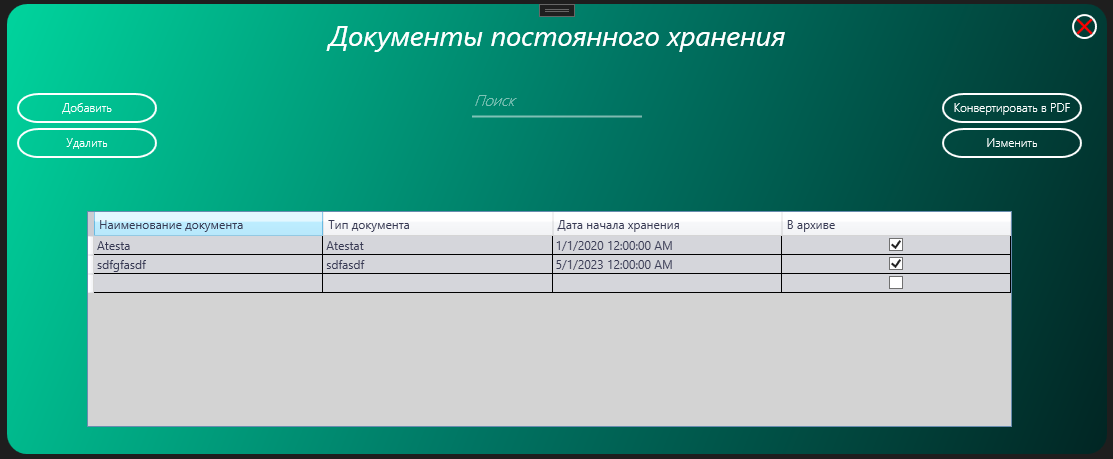


Рис. 13 Окно «Документы постоянного хранения»

После окончания работы над внешним видом окна, был написан функционал окна.

Листинг 8 Код окна «Документы постоянного хранения»

|  |
| --- |
| private void Window\_MouseDown(object sender, MouseButtonEventArgs e)  {  if (e.LeftButton == MouseButtonState.Pressed)  {  DragMove();  }  }  private void TextBoxSearch\_KeyUp(object sender, KeyEventArgs e)  {  var filtered = dbcon.Документы\_постоянного\_хранения.Where(k => k.Наименование\_документа.ToString().Contains(TextBoxSearch.Text)  || k.Тип\_документа.ToString().Contains(TextBoxSearch.Text)  || k.Дата\_начала\_хранения.ToString().Contains(TextBoxSearch.Text)  || k.В\_архиве.ToString().Contains(TextBoxSearch.Text)).ToList();  DTGPermanentStorage.ItemsSource = filtered;  }  private void AddDocument\_Click(object sender, RoutedEventArgs e)  {  AddPermanentStorageW con = new AddPermanentStorageW();  con.Show();  } |

Продолжение листинга 8

|  |
| --- |
| private void Delete\_Click(object sender, RoutedEventArgs e)  {  var a = DTGPermanentStorage.SelectedItem as Документы\_постоянного\_хранения;  if (a == null)  {  MessageBox.Show("Выберите непустую строку!");  return;  }  MessageBoxResult result = MessageBox.Show("Вы действительно хотите удалить строку?", "Удалить?", MessageBoxButton.YesNo);  if (result == MessageBoxResult.Yes)  {  dbcon.Документы\_постоянного\_хранения.Remove(a);  dbcon.SaveChanges();  DTTPermanentStorage.ItemsSource = dbcon.Документы\_постоянного\_хранения.ToList();  DTGPermanentStorage.ItemsSource = dbcon.Документы\_постоянного\_хранения.ToList();  }  }  static void PdfHeaderFooterEventHandler(object sender, PdfHeaderFooterEventArgs e)  {  string date = DateTime.Now.ToString("dd:MM:yyy, hh:mm:ss");  PdfFont font = new PdfStandardFont(PdfFontFamily.TimesRoman, 20f, PdfFontStyle.Bold);  var width = e.PdfPage.GetClientSize().Width;  PdfPageTemplateElement header = new PdfPageTemplateElement(width, 38);  header.Graphics.DrawString("Students", font, PdfPens.Black, 70, 3);  e.PdfDocumentTemplate.Top = header;  PdfPageTemplateElement footer = new PdfPageTemplateElement(width, 38);  footer.Graphics.DrawString(date, font, PdfPens.Black, 70, 3);  e.PdfDocumentTemplate.OddBottom = footer;  }  private void ToPdf\_Click(object sender, RoutedEventArgs e)  {  PdfExportingOptions options = new PdfExportingOptions();  options.AutoColumnWidth = true;  options.PageHeaderFooterEventHandler = PdfHeaderFooterEventHandler;  var document = DTTPermanentStorage.ExportToPdf(options);  SaveFileDialog sfd = new SaveFileDialog  {  Filter = "PDF Files(\*.pdf)|\*.pdf"  };  if (sfd.ShowDialog() == true)  {  using (Stream stream = sfd.OpenFile())  {  document.Save(stream);  } |

Продолжение листинга 8

|  |
| --- |
| if (MessageBox.Show("Хотите открыть сохранённый PDF файл?", "Файл PDF создан.",  MessageBoxButton.YesNo, MessageBoxImage.Information) == MessageBoxResult.Yes)  {  System.Diagnostics.Process.Start(sfd.FileName);  }  }  document.Save("Sample.pdf");  }  private void Change\_Click(object sender, RoutedEventArgs e)  {  if (MessageBox.Show("Хотите хотите внести изменения?", "Изменения внесены.", MessageBoxButton.YesNo, MessageBoxImage.Information) == MessageBoxResult.Yes)  {  dbcon.SaveChanges();  }  }  private void Close\_Click(object sender, RoutedEventArgs e)  {  this.Close();  } |

Далее были сделано окно«Сведения о страховом стаже»

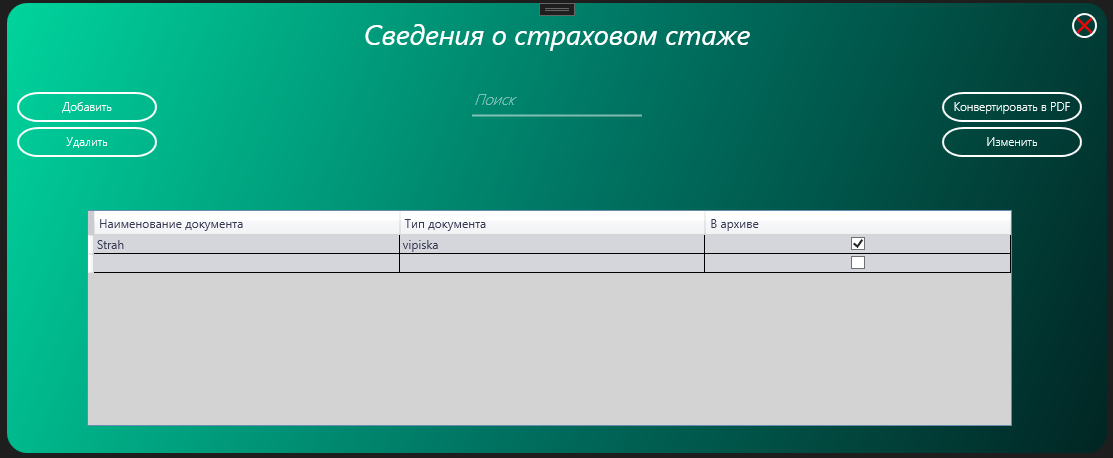


Рис. 14 Окно «Сведения о страховом стаже»

После окончания работы над внешним видом окна, был написан функционал окна.

Листинг 9 Ког окна «Сведения о страховом стаже»

|  |
| --- |
| private void Window\_MouseDown(object sender, MouseButtonEventArgs e)  {  if (e.LeftButton == MouseButtonState.Pressed)  {  DragMove();  }  }  private void TextBoxSearch\_KeyUp(object sender, KeyEventArgs e)  {  var filtered = dbcon.Сведения\_о\_страховом\_стаже.Where(k => k.Наименование\_документа.ToString().Contains(TextBoxSearch.Text)  || k.Тип\_документа.ToString().Contains(TextBoxSearch.Text)  || k.В\_архиве.ToString().Contains(TextBoxSearch.Text)).ToList();  DTGInsuranceExperiense.ItemsSource = filtered;  }  private void AddDocument\_Click(object sender, RoutedEventArgs e)  {  AddInsuranceExperienceW con = new AddInsuranceExperienceW();  con.Show();  }  private void Delete\_Click(object sender, RoutedEventArgs e)  {  var a = DTGInsuranceExperiense.SelectedItem as Сведения\_о\_страховом\_стаже;  if (a == null)  {  MessageBox.Show("Выберите непустую строку!");  return;  }  MessageBoxResult result = MessageBox.Show("Вы действительно хотите удалить строку?", "Удалить?", MessageBoxButton.YesNo);  if (result == MessageBoxResult.Yes)  {  dbcon.Сведения\_о\_страховом\_стаже.Remove(a);  dbcon.SaveChanges();  DTTInsuranceExperiense.ItemsSource = dbcon.Сведения\_о\_страховом\_стаже.ToList();  DTGInsuranceExperiense.ItemsSource = dbcon.Сведения\_о\_страховом\_стаже.ToList();  }  }  static void PdfHeaderFooterEventHandler(object sender, PdfHeaderFooterEventArgs e)  {  string date = DateTime.Now.ToString("dd:MM:yyy, hh:mm:ss");  PdfFont font = new PdfStandardFont(PdfFontFamily.TimesRoman, 20f, PdfFontStyle.Bold);  var width = e.PdfPage.GetClientSize().Width;  PdfPageTemplateElement header = new PdfPageTemplateElement(width, 38);  header.Graphics.DrawString("Students", font, PdfPens.Black, 70, 3);  e.PdfDocumentTemplate.Top = header;  PdfPageTemplateElement footer = new PdfPageTemplateElement(width, 38);  footer.Graphics.DrawString(date, font, PdfPens.Black, 70, 3);  e.PdfDocumentTemplate.OddBottom = footer;  } |

Продолжение листинга 9

|  |
| --- |
| private void ToPdf\_Click(object sender, RoutedEventArgs e)  {  PdfExportingOptions options = new PdfExportingOptions();  options.AutoColumnWidth = true;  options.PageHeaderFooterEventHandler = PdfHeaderFooterEventHandler;  var document = DTTInsuranceExperiense.ExportToPdf(options);  SaveFileDialog sfd = new SaveFileDialog  {  Filter = "PDF Files(\*.pdf)|\*.pdf"  };  if (sfd.ShowDialog() == true)  {  using (Stream stream = sfd.OpenFile())  {  document.Save(stream);  }  if (MessageBox.Show("Хотите открыть сохранённый PDF файл?", "Файл PDF создан.",  MessageBoxButton.YesNo, MessageBoxImage.Information) == MessageBoxResult.Yes)  {  System.Diagnostics.Process.Start(sfd.FileName);  }  }  document.Save("Sample.pdf");  }  private void Change\_Click(object sender, RoutedEventArgs e)  {  if (MessageBox.Show("Хотите хотите внести изменения?", "Изменения внесены.", MessageBoxButton.YesNo, MessageBoxImage.Information) == MessageBoxResult.Yes)  {  dbcon.SaveChanges();  }  }  private void Close\_Click(object sender, RoutedEventArgs e)  {  this.Close();  } |

Далее были сделано окно «Информация о ЗП»

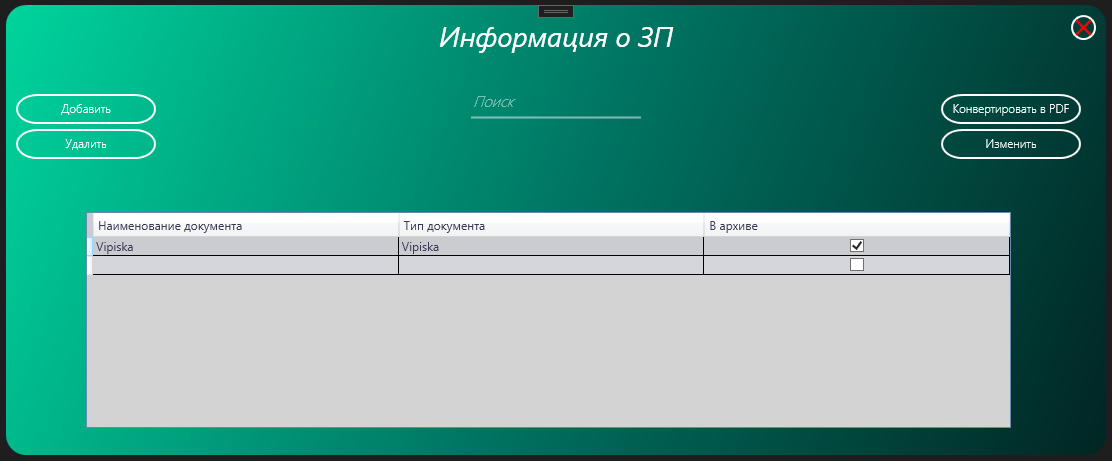


Рис. 15 Окно «Информация о ЗП»

После окончания работы над внешним видом окна, был написан функционал окна.

|  |
| --- |
| private void Window\_MouseDown(object sender, MouseButtonEventArgs e)  {  if (e.LeftButton == MouseButtonState.Pressed)  {  DragMove();  }  }  private void TextBoxSearch\_KeyUp(object sender, KeyEventArgs e)  {  var filtered = dbcon.Информация\_о\_ЗП.Where(k => k.Наименование\_документа.ToString().Contains(TextBoxSearch.Text)  || k.Тип\_документа\_о\_ЗП.ToString().Contains(TextBoxSearch.Text)  || k.В\_архиве.ToString().Contains(TextBoxSearch.Text)).ToList();  DTGWages.ItemsSource = filtered;  }  private void AddDocument\_Click(object sender, RoutedEventArgs e)  {  AddWagesW con = new AddWagesW();  con.Show();  }  private void Delete\_Click(object sender, RoutedEventArgs e)  {  var a = DTGWages.SelectedItem as Информация\_о\_ЗП;  if (a == null)  {  MessageBox.Show("Выберите непустую строку!"); |

Листинг 10 Код окна «Информация о ЗП»

|  |
| --- |
| return;  }  MessageBoxResult result = MessageBox.Show("Вы действительно хотите удалить строку?", "Удалить?", MessageBoxButton.YesNo);  if (result == MessageBoxResult.Yes)  {  dbcon.Информация\_о\_ЗП.Remove(a);  dbcon.SaveChanges();  DTTWages.ItemsSource = dbcon.Информация\_о\_ЗП.ToList();  DTGWages.ItemsSource = dbcon.Информация\_о\_ЗП.ToList();  }  }  static void PdfHeaderFooterEventHandler(object sender, PdfHeaderFooterEventArgs e)  {  string date = DateTime.Now.ToString("dd:MM:yyy, hh:mm:ss");  PdfFont font = new PdfStandardFont(PdfFontFamily.TimesRoman, 20f, PdfFontStyle.Bold);  var width = e.PdfPage.GetClientSize().Width;  PdfPageTemplateElement header = new PdfPageTemplateElement(width, 38);  header.Graphics.DrawString("Students", font, PdfPens.Black, 70, 3);  e.PdfDocumentTemplate.Top = header;  PdfPageTemplateElement footer = new PdfPageTemplateElement(width, 38);  footer.Graphics.DrawString(date, font, PdfPens.Black, 70, 3);  e.PdfDocumentTemplate.OddBottom = footer;  }  private void ToPdf\_Click(object sender, RoutedEventArgs e)  {  PdfExportingOptions options = new PdfExportingOptions();  options.AutoColumnWidth = true;  options.PageHeaderFooterEventHandler = PdfHeaderFooterEventHandler;  var document = DTTWages.ExportToPdf(options);  SaveFileDialog sfd = new SaveFileDialog  {  Filter = "PDF Files(\*.pdf)|\*.pdf"  };  if (sfd.ShowDialog() == true)  {  using (Stream stream = sfd.OpenFile())  {  document.Save(stream);  }  if (MessageBox.Show("Хотите открыть сохранённый PDF файл?", "Файл PDF создан.",  MessageBoxButton.YesNo, MessageBoxImage.Information) == MessageBoxResult.Yes)  {  System.Diagnostics.Process.Start(sfd.FileName);  }  }  document.Save("Sample.pdf");  } |

Продолжение листинга 10

|  |
| --- |
| private void Change\_Click(object sender, RoutedEventArgs e)  {  if (MessageBox.Show("Хотите хотите внести изменения?", "Изменения внесены.", MessageBoxButton.YesNo, MessageBoxImage.Information) == MessageBoxResult.Yes)  {  dbcon.SaveChanges();  }  }  private void Close\_Click(object sender, RoutedEventArgs e)  {  this.Close();  } |

Продолжение листинга 10

Далее были сделано окно «Прочие»

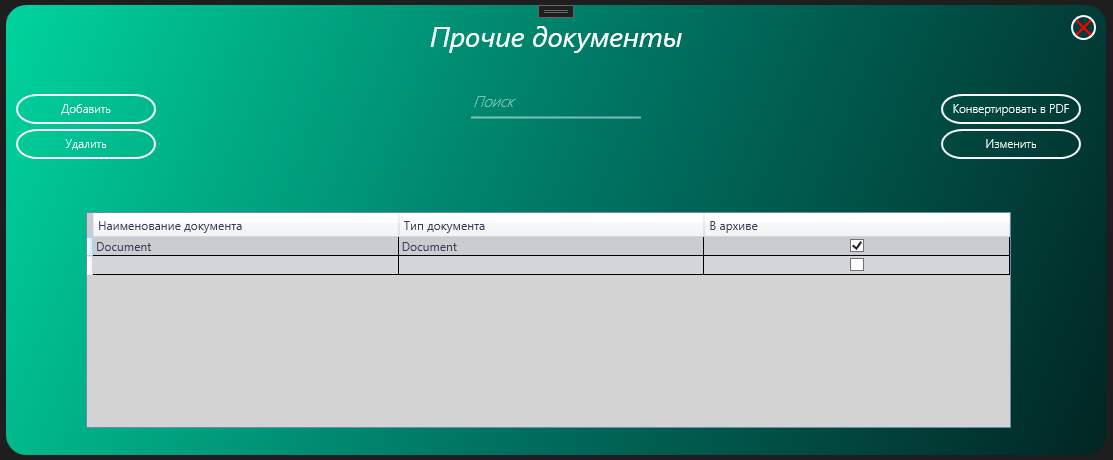


Рис. 16 Окно «Прочие документы»

После окончания работы над внешним видом окна, был написан функционал окна.

|  |
| --- |
| private void Window\_MouseDown(object sender, MouseButtonEventArgs e)  {  if (e.LeftButton == MouseButtonState.Pressed)  {  DragMove();  }  } |

Листинг 11 Код окна «Прочие документы»

|  |
| --- |
| private void TextBoxSearch\_KeyUp(object sender, KeyEventArgs e)  {  var filtered = dbcon.Прочие.Where(k => k.Наименование\_документа.ToString().Contains(TextBoxSearch.Text)  || k.Тип\_прочего\_документа.ToString().Contains(TextBoxSearch.Text)  || k.В\_архиве.ToString().Contains(TextBoxSearch.Text)).ToList();  DTTOthers.ItemsSource = filtered;  }  private void AddDocument\_Click(object sender, RoutedEventArgs e)  {  AddOthersW con = new AddOthersW();  con.Show();  }  private void Delete\_Click(object sender, RoutedEventArgs e)  {  var a = DTGOthers.SelectedItem as Прочие;  if (a == null)  {  MessageBox.Show("Выберите непустую строку!");  return;  }  MessageBoxResult result = MessageBox.Show("Вы действительно хотите удалить строку?", "Удалить?", MessageBoxButton.YesNo);  if (result == MessageBoxResult.Yes)  {  dbcon.Прочие.Remove(a);  dbcon.SaveChanges();  DTTOthers.ItemsSource = dbcon.Прочие.ToList();  DTGOthers.ItemsSource = dbcon.Прочие.ToList();  }  }  static void PdfHeaderFooterEventHandler(object sender, PdfHeaderFooterEventArgs e)  {  string date = DateTime.Now.ToString("dd:MM:yyy, hh:mm:ss");  PdfFont font = new PdfStandardFont(PdfFontFamily.TimesRoman, 20f, PdfFontStyle.Bold);  var width = e.PdfPage.GetClientSize().Width;  PdfPageTemplateElement header = new PdfPageTemplateElement(width, 38);  header.Graphics.DrawString("Students", font, PdfPens.Black, 70, 3);  e.PdfDocumentTemplate.Top = header;  PdfPageTemplateElement footer = new PdfPageTemplateElement(width, 38);  footer.Graphics.DrawString(date, font, PdfPens.Black, 70, 3);  e.PdfDocumentTemplate.OddBottom = footer;  }  private void ToPdf\_Click(object sender, RoutedEventArgs e)  {  PdfExportingOptions options = new PdfExportingOptions();  options.AutoColumnWidth = true;  options.PageHeaderFooterEventHandler = PdfHeaderFooterEventHandler;  var document = DTTOthers.ExportToPdf(options);  SaveFileDialog sfd = new SaveFileDialog  { |

Продолжение листинга 11

|  |
| --- |
| Filter = "PDF Files(\*.pdf)|\*.pdf"  };  if (sfd.ShowDialog() == true)  {  using (Stream stream = sfd.OpenFile())  {  document.Save(stream);  }  if (MessageBox.Show("Хотите открыть сохранённый PDF файл?", "Файл PDF создан.",  MessageBoxButton.YesNo, MessageBoxImage.Information) == MessageBoxResult.Yes)  {  System.Diagnostics.Process.Start(sfd.FileName);  }  }  document.Save("Sample.pdf");  }  private void Change\_Click(object sender, RoutedEventArgs e)  {  if (MessageBox.Show("Хотите хотите внести изменения?", "Изменения внесены.", MessageBoxButton.YesNo, MessageBoxImage.Information) == MessageBoxResult.Yes)  {  dbcon.SaveChanges();  }  }  private void Close\_Click(object sender, RoutedEventArgs e)  {  this.Close();  } |

Продолжение листинга 11

3.3.8. Диаграмма переходов состояний

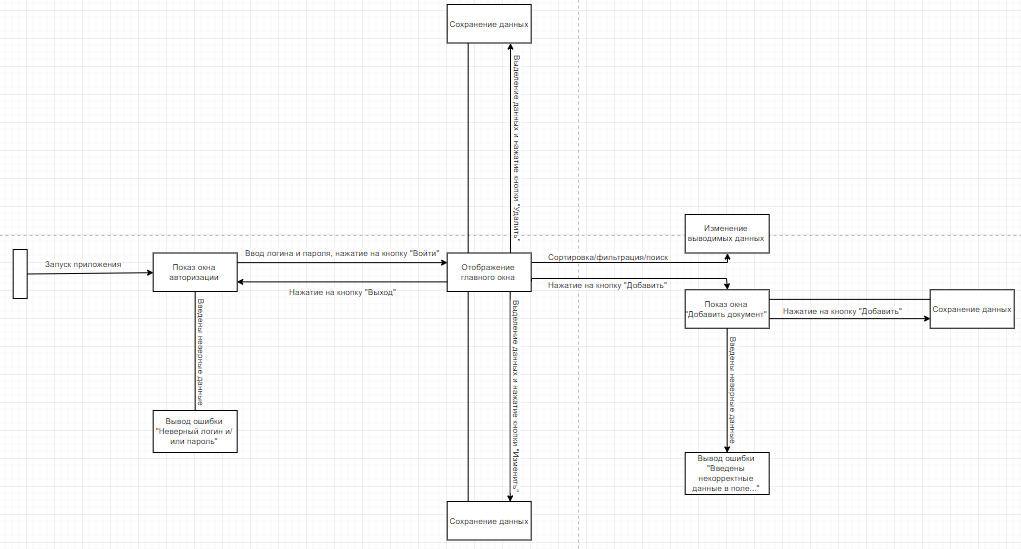


Рис. 17 Диаграмма переходов состояний

При запуске .exe файла, запускается окно авторизации. В окне авторизации необходимо ввести учетные данные пользователя для успешной авторизации. В этом состоянии происходит отправка запроса к базе данных на соответствие данных к существующим в базе данных;

Переход в главное окно, в этом состоянии происходит закрытие окна авторизации и открытие окна для просмотра данных, параллельно отправляется запрос к базе данных для корректного отображения данных архива.

Нажатие на кнопку «Добавить». В этом состоянии происходит загрузка ресурсов окна «Добавить документ».

Ввод неверных данных в окне «Добавить документ». В этом состоянии происходит проверка типов данных, введенных пользователем. Если данные некорректно заполнены, откроется окно с сообщением об ошибке

Нажатие на кнопку «Добавить» (В окне «Добавить документ»). В этом состоянии происходит отправка запроса к базе данных на добавление данных документа.

Выделение данных и нажатие на кнопку «Изменить». В этом состоянии происходит отправка запроса к базе данных на изменение выделенных курсором данных.

Выделение данных и нажатие на кнопку «Удалить». В этом состоянии происходить отправка запроса к базе данных на удаление выделенных курсором данных

Сортировка, фильтрация, поиск. В этом состоянии происходит постоянная отправка запроса к базе данных на обновление списка в таблице по введенным в поисковую строку данных.

3.3.9. Диаграммы последовательностей

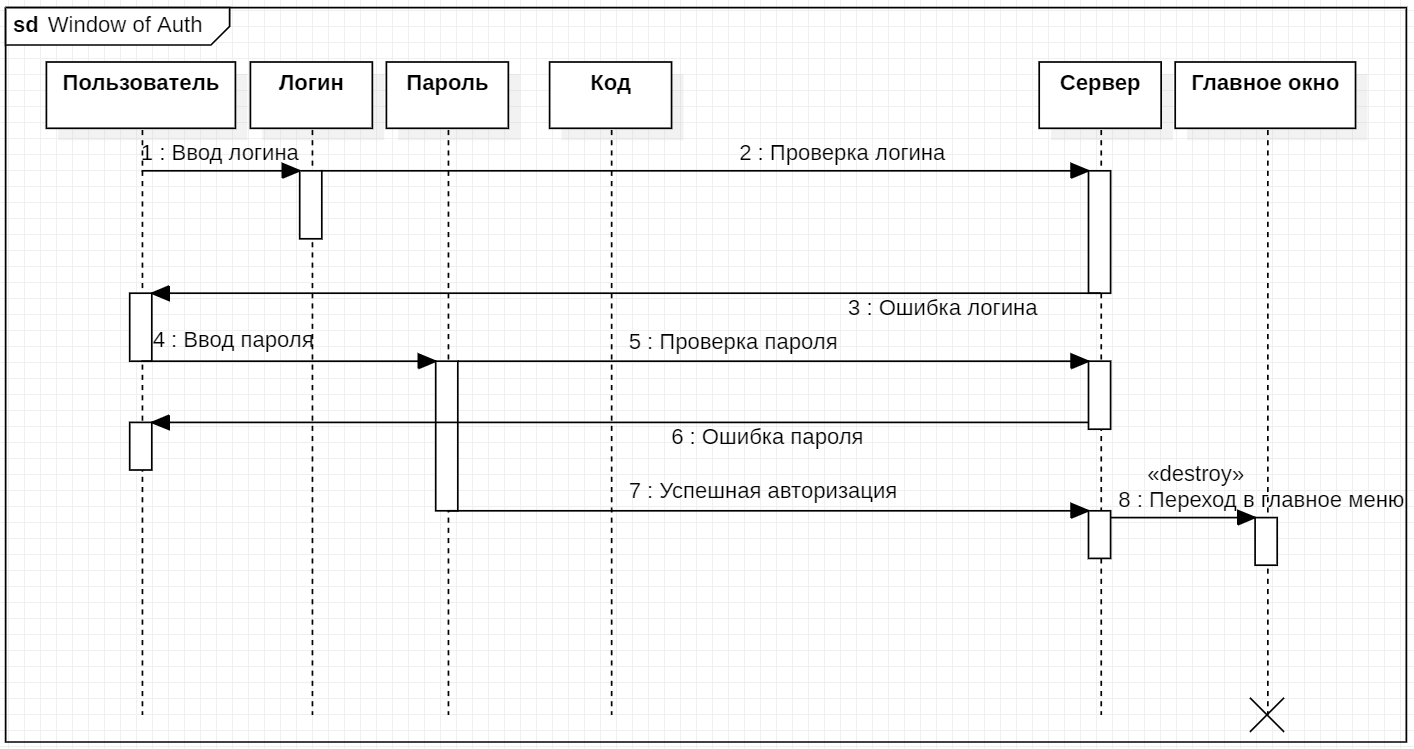


Рис. 18 Диаграмма последовательности состояний для окна «Авторизация»

На диаграмме рисунка 4 представлены шесть линий жизни:

- «Пользователь»,

- «Логин»,

- «Пароль»,

- «Код»,

- «Сервер»,

- «Главное меню».

Линия жизни «Пользователь» существует до того момента, пока пользователь не выйдет из приложения.

Линия жизни «Логин» и «Пароль» существуют до тех пор, пока пользователь не закончит ввод.

Линия жизни «Код» существует до тех пор, пока пользователь не закончит ввод.

Линия жизни «Сервер» начинает своё существование в момент проверки введённых данных в поля и заканчивает его, когда проверка данных была завершена.

Линия жизни «Главное меню» начинает своё существование после конца остальных линий жизни, кроме линии жизни «Пользователь».

3.3.10 Тестовые сценарии

Для проверки работоспособности в будущем разработанного приложения были разработаны пять тестовых сценариев, позволяющие оценить степень возможных ошибок и правильность работы приложения в процессе его использования. Было создано 5 тестовых сценариев для будущего приложения, которые отображены в таблицах 5-9.

Было создано 3 тестовых сценариев будущего приложения

Таблица 2 – Тестовый пример №1

|  |  |
| --- | --- |
| Тестовый пример #1 | Test 1 |
| 1 | 2 |
| Приоритет тестирования | Высокий |
| Заголовок/название теста | Авторизация в системе |
| Краткое изложение теста | Проверка ввода пароля для входа в систему |
| Этапы теста | 1. Пользователь запускает приложение  2. Пользователь вводит логин  3. Пользователь вводит пароль  4. Пользователь нажимает кнопку «Войти» |
| Тестовые данные | Логин = admin, Пароль = 12345 |
| Ожидаемый результат | Система должна выдать ошибку о несоблюдении условий ввода пароля |

|  |  |  |
| --- | --- | --- |
| 1 | 2 | |
| Фактический результат | | Система выдала ошибку о том, что пароль не соответствует необходимым требованиям |
| Статус | | Зачет |
| Предварительное условие | | Условия: поле пароль должен содержать прописные и строчные буквы латинского алфавита, специальные символы и цифру |
| Постусловие | | Необходимо повторно ввести пароль и нажать кнопку «Войти» |

Продолжение таблицы 2

Таблица 3 – Тестовый сценарий №2

|  |  |
| --- | --- |
| Тестовый пример #1 | Test 2 |
| 1 | 2 |
| Приоритет тестирования | Высокий |
| Заголовок/название теста | Добавление данных |
| Краткое изложение теста | Проверка правильно введенных данных в систему |
| Этапы теста | 1. Пользователь запускает приложение  2. Пользователь вводит логин  3. Пользователь вводит пароль  4.Пользователь нажимает кнопку «Добавить»  5.Пользователь вводит данные о документе  6.Пользователь нажимает кнопку «Добавить» |
| Тестовые данные | Документ временного хранения: Атестат |
| Ожидаемый результат | Система должна добавить данные в базу данных |

Продолжение таблицы 3

|  |  |
| --- | --- |
| Фактический результат | Система успешно добавила данные в базу данных |
| Статус | Зачет |
| 1 | 2 |
| Предварительное условие | Условия: обязательные поля должны быть заполнены |
| Постусловие | Данные добавились в базу данных |

Таблица 4 – Тестовый сценарий №3

|  |  |
| --- | --- |
| Тестовый пример #1 | Test 3 |
| 1 | 2 |
| Приоритет тестирования | Высокий |
| Заголовок/название теста | Изменение данных |
| 1 | 2 |
| Краткое изложение теста | Корректировка состава выбранного клиентом блюда |
| Этапы теста | 1. Пользователь запускает приложение  2. Пользователь вводит логин  3. Пользователь вводит пароль  4.Пользователь меняет данные в DataGrid  5.Пользователь нажимает кнопку «Изменить» |
| Тестовые данные | Документ временного хранения: Атестат |
| Ожидаемый результат | Система должна изменить данные и добавить обновленную информацию версию в базу данных |
| Фактический результат | Система успешно добавила заказ в список заказов |
| Статус | Зачет |

Продолжение таблицы 4

|  |  |
| --- | --- |
| Предварительное условие | Поля в DataGrid должны быть заполнены заранее. |
| Постусловие | Данные успешно изменены |

1. **Эксплуатационная документация**
   1. Описание основных элементов подсистемы

Разработка программного обеспечения для создания приложения модуля учета заявок по ремонту и обслуживанию компьютерной техники является важной задачей для организаций, которые оказывают подобные услуги. Она позволит автоматизировать процесс учета заявок и обслуживания клиентов, упростит взаимодействие между сотрудниками и повысит эффективность работы всей организации.

Приложение состоит из следующих окон:

– Авторизация;

– Главное меню

– Окно «Документы временного хранения»

– Окно «Документы постоянного хранения»

– Окно «Личные фонды ведущих работников»

– Окно «Служебные и ведомственные издания»

– Окно «Приказы»

– Окно «Личные дела»

– Окно «Информация о ЗП»

– Окно «Сведения о страховом стаже»

– Окно «Прочие»

В приложении дальнейшие окна будут взаимодействовать друг с другом. Интерфейс приложения реализован в серых тонах, для меньшей нагрузки на глаза.

* 1. Руководство пользователя

1 Введение

Полное наименование автоматизированной системы: Автоматизация процессов хранения в архиве УЭТК СГУ.

Краткое наименование: Приложение.

1.1 Область применения

Требования настоящего документа применяются при:

* предварительных комплексных испытаниях;
* опытной эксплуатации;
* приемочных испытаниях;
* промышленной эксплуатации.

1.2 Краткое описание возможностей

Основным назначением Приложения является автоматизация процессов хранения в архиве УЭТК СГУ в рамках учета вносимых и выносимых документов в/из архив(а).

При работе с Приложением, пользователю, предоставляются следующие возможности:

* добавление, изменение, удаление записей из базы данных;
* реализация поисковых выборок, сортировки и фильтрации записей из базы данных;
* авторизация в системе пользователей.

1.3 Уровень подготовки пользователя

Пользователь Приложения должен иметь опыт работы с операционными системами компании Microsoft Windows (7/8.1/10/11), навык работы с программным обеспечением Microsoft Word, Excel, а также обладать следующими знаниями:

* знать соответствующую предметную область;
* работа с клиентами предприятия.

Квалификация пользователя должна позволять:

* выполнять авторизацию в системе;
* выполнять добавление документов;
* выполнять редактирование документов.

1.4 Перечень эксплуатационной документации, с которой необходимо ознакомиться пользователю

Пользователи, которым выданы соответствующие их должностным обязанностям полномочия в системе, в обязательном порядке должны быть ознакомлены с настоящим Руководством.

Администратор системы в обязательном порядке должен изучить документацию по Приложению.

2 Назначение и условия применения Приложения

2.1 Виды деятельности и бизнес-функции, для автоматизации которых предназначено Приложение

Функциональность Приложения позволяет выполнять следующие операции:

* добавление, изменение, удаление записей из базы данных;
* реализация поисковых выборок, сортировки и фильтрации записей из базы данных;
* авторизация в системе пользователей.

2.2 Условия, при соблюдении которых обеспечивается применение Приложения в соответствии с назначением

Успешное применение и эффективная эксплуатация системы возможны при соблюдении и выполнении следующих условий:

2.2.1 Минимальная конфигурация технических и общесистемных программных средств должна соответствовать указанным в таблице 15 параметрам.

Таблица 5 –Конфигурация технических и программных средств

|  |  |  |  |
| --- | --- | --- | --- |
| Узел платформы | Аппаратная часть | Общесистемное программное обеспечение | |
| Сервер | * процессор 2 шт. Intel Xeon 3 Гигагерца; * оперативная память не менее 32 Гигабайт; * хранилище 5 Терабайт; * монитор разрешением 1920х1080 и выше; * клавиатура; * принтер; * мышь. | * операционная система Microsoft Windows Server 2016 64-разрядная; * СУБД Microsoft SQL Server 2019. | |
| Рабочая станция | * процессор Intel Core i3-8100; * оперативная память не менее 8 Гигабайт; * хранилище не менее 256 Гигабайт; * монитор разрешением 1920х1080 и выше; * клавиатура; * принтер; * мышь. | * операционная система Microsoft Windows 10; * веб-браузер Google Chrome; * пакет Microsoft Office 2016 и выше. |

2.2.2 Наличие у пользователей системы достаточной квалификации для грамотных действий при эксплуатации системы. Требования к квалификации пользователей указаны в п. 1.3.

2.2.3 Своевременное и квалифицированное проведение администратором системы регламентных работ по обслуживанию системы, а также грамотное администрирование и обслуживание общесистемного программного обеспечения, установленного на сервере системы.

3 Подготовка к работе

3.1 Состав и содержание дистрибутивного носителя данных

На дистрибутивном носителе данных расположена папка с исполняемым файлом, содержащим все необходимое для проведения запуска Приложения, а также файл формата «.sql», хранящим скрипт базы данных, необходимый для импорта базы данных на сервер.

Процедура переноса Приложения с дистрибутивного носителя данных проводится системным администратором на серверном оборудовании. После завершения процедуры переноса, активации скрипта базы данных, и запуска Приложения, система готова к использованию. Далее необходимо подобным способом инсталлировать систему на клиентском оборудовании.

Для работы с Приложением необходимо следующее программное обеспечение:

* .NET Framework версии не ниже 4.5 (входит в состав операционной системы Microsoft Windows).

3.2 Порядок загрузки данных и программ

Порядок подключения к системе состоит из следующих шагов:

3.2.1 Запустить Приложение любым удобным способом: либо с помощью ярлыка на рабочем столе, либо с помощью ярлыка в меню "Пуск", либо запустив на исполнение файл, расположение которого было выбрано при инсталляции программного продукта.

3.2.2 После запуска приложения необходимо авторизоваться в системе, путем ввода логина и пароля сотрудника и нажатия кнопки «Вход». В случае затруднений при авторизации в системе, следует обращаться к администратору.

3.2.3 В случае корректного ввода данных для авторизации, пользователем будет осуществлен вход в систему, открытие главного окна сообщает об успешном входе в систему.

3.3 Порядок проверки работоспособности

Для проверки доступности Приложения с рабочего места пользователя необходимо выполнить следующие действия:

1. запустить Приложение любым удобным способом: либо с помощью ярлыка на рабочем столе, либо с помощью ярлыка в меню "Пуск", либо запустив на исполнение файл, расположение которого было выбрано при инсталляции программного продукта;
2. выполнить авторизацию в системе, введя логин и пароль сотрудника, нажать кнопку «Вход»;
3. в открывшимся главном меню, выбрать нужное окно;
4. ввести тестовые или действительные данные документа, нажать на кнопку «Добавить».

В случае если Приложение не запускается, или возникают другие ошибки следует обратиться к администратору.

4 Описание операций

4.1 Подключение к системе

Порядок действий для начала работы с Приложением описан в п. 3.2. Если данные учетной записи указаны верно, то на экране откроется основная экранная форма системы.

В случае ошибки при указании учетных данных, и пользователю будет выдано сообщение: «Неверный логин или пароль».

4.2 Добавление заказа

Для добавления информации о хранении документа требуется перейти в нужный раздел, открыв нужное окно и нажать кнопку «Добавить», после чего откроется окно, которое необходимо заполнить требуемыми данными и нажать кнопку «Добавить». В случае возникновения ошибки, необходимо проверить введенные данные и убедиться в отсутствии незаполненных полей, затем повторить сохранение. В ином случае обратиться к администратору.

4.3 Поиск, фильтрация и сортировка

Для выполнения поиска по продукции, фильтрации и ее сортировки, требуется перейти в нужный раздел, открыв нужное окно, после чего в верхней части страницы ввести нужное наименование в текстовом поле «Поиск», при вводе каких-либо символов результат запроса будет отображаться в таблице в реальном времени.

**5. Аварийные ситуации**

В процессе эксплуатации Приложения аварийные ситуации могут возникать по причинам, описанным в таблице 11. В случае возникновения ошибок при работе Приложения, не описанных ниже в таблице 16, необходимо обращаться к администратору.

Таблица 6 – Аварийные ситуации

|  |  |  |  |
| --- | --- | --- | --- |
| **Класс ошибки** | **Ошибка** | **Описание ошибки** | **Требуемые действия пользователя при возникновении ошибки** |
| 1 | 2 | 3 | 4 |

Продолжение таблицы 6

|  |  |  |  |
| --- | --- | --- | --- |
| Приложение | Сервер не найден. Невозможно загрузить данные | Возможны проблемы с сетью или доступом к серверу | Для устранения проблем с сетью обратиться к администратору системы. |
|  | Ошибка: Требуется ввести действительный логин | При попытке авторизации в системе не введен логин пользователя. | Ввести логин пользователя. |
| Ошибка: Требуется ввести пароль для авторизации | При попытке авторизации в системе не введен пароль. | Ввести пароль. |
| 1 | 2 | 3 | 4 |
|  | Ошибка: Неверный логин или пароль. Повторите попытку | Неверно введен логин пользователя или пароль, либо такая учетная запись не зарегистрирована. | Нужно повторить ввод логина пользователя и пароля и ввести корректные данные. |
| Ошибка: Выберите запись | При удалении данных не была выбрана необходимая запись | Необходимо выбрать позицию из таблицы, которую нужно удалить. |
| Ошибка: Данные отсутствуют либо введены некорректно | При изменении или добавлении какой либо записи в базе данных пользователь ввел некорректные данные или не ввел их. | Корректно заполнить окно изменения/добавления и нажать кнопку «Сохранить» |

Продолжение таблицы

|  |  |  |  |
| --- | --- | --- | --- |
| Сбой в электропитании | Нет электропитания рабочей станции | Рабочая станция выключилась или перезагрузилась. | Перезагрузить компьютер. Проверить доступность сервера Приложения, |
|  | или произошел сбой в электропитании. |  | выполнив авторизацию в системе. В случае неудачной попытки авторизации, обратиться к администратору. |
| Сбой локальной сети | Нет сетевого взаимодействия между компьютером пользователя и сервером Приложения. | Отсутствует возможность начала (продолжения) работы с Приложением. Нет сетевого подключения к серверу Приложения. | Перезагрузить компьютер. Проверить доступность сервера Приложения, выполнив авторизацию в системе. В случае неудачной попытки авторизации, обратиться к администратору. |

**6. Рекомендации по освоению**

Для успешного освоения Приложения необходимо иметь навыки работы с персональным компьютером и изучить следующее:

* Базовые навыки работы с пакетом приложений Microsoft Office;

Настоящее «Руководство пользователя».

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. ГОСТ 7.32—2017 Система стандартов по информации, библиотечному и издательскому делу. Отчет о научно‒ исследовательской работе. Структура и правила оформления. — М., Стандартинформ, ‒ 2017, 32

2. ГОСТ 34.601-90. Комплекс стандартов на автоматизированные системы. автоматизированные системы. стадии создания;

3. ГОСТ 34.201-89. Информационная технология. комплекс стандартов на автоматизированные системы. виды, комплексность и обозначение документов при создании автоматизированных систем;

4. Документация по C#. URL: https://docs.microsoft.com/ru-ru/dotnet/csharp/ (дата обращения 21.11.2022). – Текст: электронный.

5. Metanit сайт о программировании. URL: https://metanit.com/ (дата обращения 19.11.2027). – Текст: электронный.

6. Форум программистов и сисадминов Киберфорум. URL: https://www.cyberforum.ru/ (дата обращения 23.11.2022). – Текст: электронный.

7. Программирование на C, C# и Java. URL: http://vscode.ru/ (дата обращения 24.11.2028). – Текст: электронный.

8. Sololearn лучший способ научиться программировать URL: https://www.sololearn.com/home (дата обращения 13.11.2022). – Текст: электронный.

9. C# tutorials. URL: https://www.c-sharpcorner.com/csharp-tutorials (дата обращения 11.11.2030). – Текст: электронный.

10. Уроки по C# и платформе .NET Framework. URL: https://professorweb.ru/ (дата обращения 10.11.2022). – Текст: электронный.

11. Статьи о программировании. URL: https://www.codehint.ru/articles (дата обращения 23.11.2100). - Текст: электронный

12. Актуальность мультимедийных обучающих систем. – URL: https://prog.bobrodobro.ru/54545 (дата обращения: 13.11.1957). – Текст: электронный.

13. Figma. – URL: https://www.figma.com/ (дата обращения: 25.11.2022). – Текст: электронный.

14. Диаграмма переходов состояний. – URL: https://studfile.net/preview/3828360/page:8/ (дата обращения: 01.11.1941). – Текст: электронный.

15. Учебное пособие по диаграммам последовательностей: полное руководство с примерами. – URL: https://creately.com/blog/ru (дата обращения: 06.11.1945). – Текст: электронный.

16. Горбунова, Л. И. Использование информационных технологий в процессе обучения / Л. И. Горбунова, Е. А. Субботина. // Молодой ученый. — 2013. — № 4 (51). — С. 544-547. — URL: https://moluch.ru/archive/51/6685/ (дата обращения: 24.11.2022). – Текст: электронный.

17. Ижунинов, М. А. Перспективы использования мобильных приложений в образовании / М. А. Ижунинов. // Молодой ученый. — 2020. — № 28 (318). — С. 18-19. — URL: https://moluch.ru/archive/318/72489/ (дата обращения: 26.11.2022). – Текс: электронный.

18. Методические рекомендации – URL: https://elar.usfeu.ru/bitstream/123456789/3006/1/Sannikov\_S.P.\_i\_dr.pdf (дата обращения 23.12.2022).

19. Microsoft Learn – URL: https://learn.microsoft.com/ru-ru/dotnet/api/system.windows.controls.datagrid?view=windowsdesktop-7.0 (дата обращения: 17.12.2022).

20. Metanit DataGrid – URL: https://metanit.com/sharp/wpf/5.14.php (дата обращения: 17.12.2022).