Министерство образования Московской области

Государственное образовательное учреждение высшего образования Московской области

«Государственный гуманитарно-технологический университет»

**Ликино-Дулевский политехнический колледж – филиал ГГТУ**

**КУРСОВОЙ ПРОЕКТ**

Тема: Разработка программного модуля для учета выдачи форменной одежды сотрудникам предприятия

МДК 01.01 «Разработка программных модулей»

**Выполнил:**

Ульянов Андрей Дмитриевич\_\_\_\_\_\_\_\_\_

студент группы ИСП.20А\_\_\_\_\_\_\_\_\_\_\_\_\_

09.02.07 Информационные системы и программирование)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

очной формы обучения\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Руководитель:**

Кузьмина Елена Евгеньевна­­­­­­­­­­­­­­­­­­\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Оценка

\_\_\_\_\_\_\_\_\_/\_\_\_­­­­­­\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/

Подпись руководителя

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

Ликино-Дулево

2023 год

**Введение**

Программный модуль разработан для учета выдачи форменной одежды сотрудникам предприятия, предназначен для автоматизации процесса учета и контроля над выдачей и заменой форменной одежды. Он может быть использован в различных организациях, где необходимо учетное ведение выдачи специальной одежды или других материальных ценностей. Программный модуль позволяет упростить и ускорить процесс выдачи форменной одежды, уменьшить вероятность ошибок, связанных с ручным ведением учета, а также повысить контроль за использованием форменной одежды. Объектом, в котором может использоваться данный программный модуль, является организация, предоставляющая своим сотрудникам специальную одежду в рамках их трудовой деятельности.

1. **Назначение разработки**

Функциональное назначение программы, разработанной для учета выдачи форменной одежды сотрудникам предприятия, заключается в автоматизации процесса учета и контроля над выдачей и возвратом форменной одежды. Конкретные функции программы включают в себя создание электронной базы данных о сотрудниках, учет количества выданной одежды, генерацию отчетов о выдаче, а также возможность редактирования и удаления данных.

Эксплуатационное назначение программы заключается в обеспечении эффективной работы процесса выдачи и учета форменной одежды в организации. Программа позволяет сократить время и уменьшить вероятность ошибок при ведении учета, что улучшает качество и надежность данного процесса. Кроме того, благодаря электронной базе данных, можно легко отслеживать историю выдачи и возврата форменной одежды для каждого сотрудника, что упрощает контроль за ее использованием.

1. **Требования к программному модулю**
   1. **Требования к функциональным характеристикам**

Задача на разработку программы: разработать программное обеспечение для автоматизации процесса учета выдачи форменной одежды сотрудникам предприятия. Программа должна обеспечивать создание базы данных о сотрудниках и выданной им форменной одежде, учет количества выданной одежды для каждого сотрудника, отслеживание истории выдачи одежды, генерацию отчетов о выдаче форменной одежды, возможность редактирования и удаления данных, а также взаимодействие с другими программами для обмена данными. Программа должна обеспечить точность и своевременность ведения учета, а также предоставить возможность получения нужной информации в удобном формате.

Состав выполняемых функций должен включать:

Создание базы данных о сотрудниках и выданной им форменной одежде;

Учет количества выданной одежды для каждого сотрудника;

Отслеживание истории выдачи форменной одежды;

Генерация отчетов о выдаче форменной одежды;

Возможность редактирования и удаления данных.

Организация входных и выходных данных:

Входные данные должны включать информацию о сотрудниках, выданной им форменной одежде, дате выдачи и замены, а также другую необходимую информацию;

Выходные данные должны включать отчеты о выдаче форменной одежды.

Описание применяемых математических методов:

Программа может использовать математические методы для вычисления цены за единицу одежды учитывая индивидуальную скидку работника, вычисления даты замены одежды, а также для генерации отчетов.

Описание алгоритма функционирования программы:

Программа должна обеспечивать создание и обновление базы данных о сотрудниках и выданной им форменной одежде;

Программа должна автоматически обновлять данные о количестве выданной одежды;

Программа должна предоставлять возможность генерации отчетов о выдаче форменной одежды.

Возможные взаимодействия с другими программами:

Программа может взаимодействовать с другими программами для формирования отчетов, например с программой Microsoft Word.

**2.2 Требования к составу и параметрам технических средств**

.

|  |  |
| --- | --- |
| Название | Характеристика |
| Процессор Intel Core i3-2130 Sandy Bridge LGA1155, 2 x 3400 МГц | Сокет - LGA1155  Количество ядер - 2 x 3400 МГц  Объем кэша - L2/L3 0.50 МБ/3 МБ  Тепловыделение - 65 Вт |
| Модуль памяти DDR3 4GB Patriot Memory PSD34G13332 | Стандарт – ddr3  Объем памяти – 4гб |
| 19.5" Монитор AOC e2070Swn, 1600x900, 76 Гц, TN, черный | Тип монитора - ЖК  Диагональ - 19.5 "  Макс. Разрешение -1600x900  Соотношение сторон - 16:9 |
| Клавиатура Canyon CNS-HKB5RU | Тип клавиатуры - мембранная  Интерфейс - USB |
| Мышь компьютерная Logitech B100 белая (910-003360) | Тип мыши – оптическая  Интерфейс - USB  Разрешение dpi - 800  Тип питания - USB |
| Жесткий диск Western Digital WD1000VE 100Gb 5400 IDE 2,5" HDD | Емкость – 100 Гб  Форм-фактор – 2.5" |

**2.3 Требования к информационной и программной совместимости**

Требования к информационным структурам на входе и выходе:

Входные данные: данные о сотрудниках предприятия, данные о выданной форменной одежде.

Выходные данные: отчеты о выдаче форменной одежды.

Методы решения:

Создание базы данных о сотрудниках и выданной им форменной одежде.

Разработка функций для учета количества выданной и возвращенной одежды для каждого сотрудника.

Разработка функций для отслеживания истории выдачи форменной одежды.

Разработка функций для генерации отчетов о выдаче форменной одежды.

Разработка функций для редактирования и удаления данных.

Разработка функций для взаимодействия с другими программами для обмена данными.

Исходные коды и язык программирования:

Язык программирования - C#.

Исходные коды программы должны быть написаны на языке C# с использованием среды разработки Visual Studio.

Программные средства:

Microsoft Word - для написания технического задания.

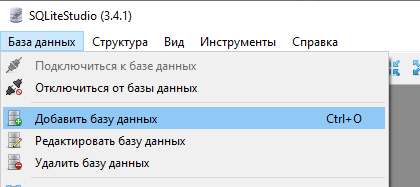
Среда разработки Visual Studio - для разработки программного обеспечения.

База данных SQLite- для хранения и управления базой данных о сотрудниках и выданной им форменной одежде.

1. **Руководство программиста**
   1. **Разработка базы данных**

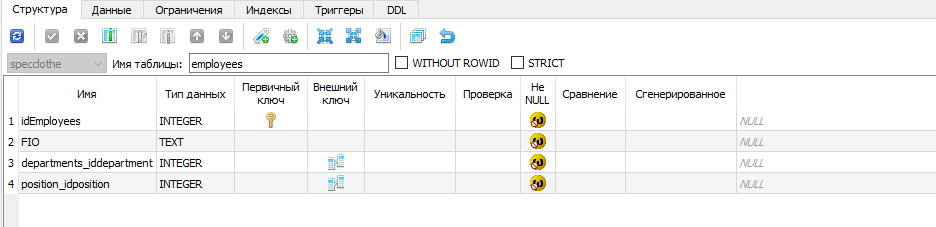
Разработка базы данных происходит в SQLite Studio.

Создание Базы данных:



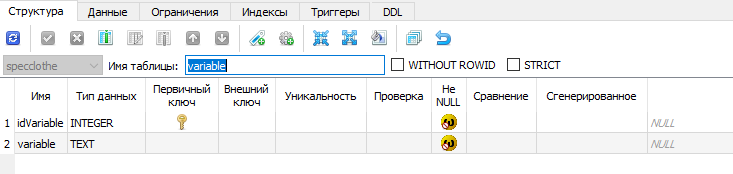
«Рис 1. Создание базы данных»

Создание таблицы «Employees»:



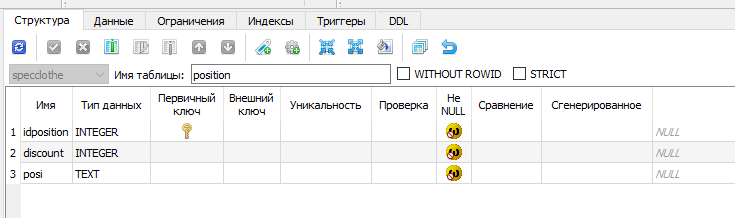
«Рис 2. Создание таблицы «Employees»»

Создание таблицы «Variable»:



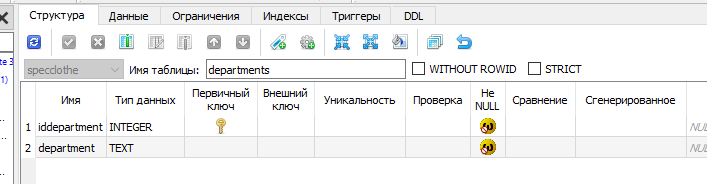
«Рис 3. Создание таблицы «Variable»»

Создание таблицы «Position»:



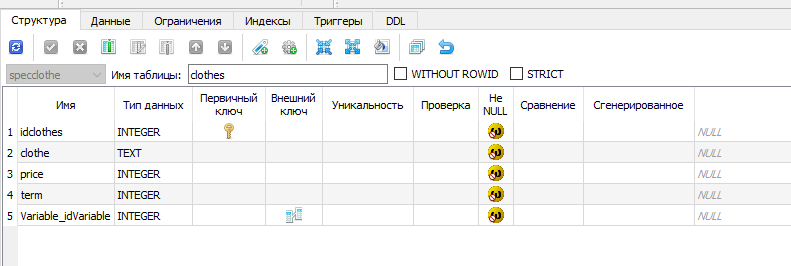
«Рис 4. Создание таблицы «Position»»

Создание таблицы «Departments»:



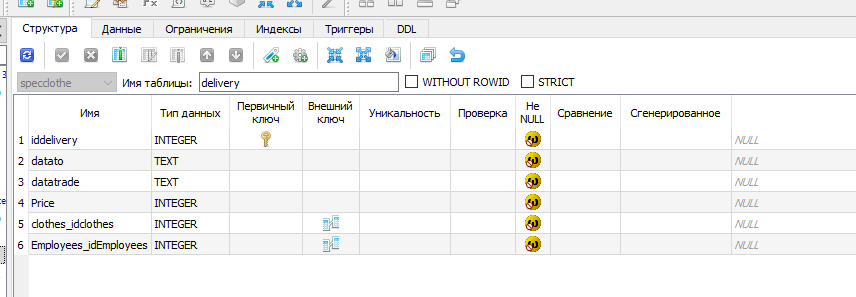
«Рис 5. Создание таблицы «Departments»»

Создание таблицы «Сlothes»:



«Рис 6. Создание таблицы «Сlothes»»

Создание таблицы «Delivery»:



«Рис 7. Создание таблицы «Delivery»»

* 1. **Разработка программного модуля**

Функция добавления и удаления записей в базе данных:

{// удаление нескольких пользователей

var usersForRemoving = DGrid.SelectedItems.Cast<Employee>().ToList();

if (MessageBox.Show($"Удалить {usersForRemoving.Count()} записей?",

"Внимание", MessageBoxButton.YesNo, MessageBoxImage.Question) == MessageBoxResult.Yes)

try

{

SpecclotheContext.GetContext().Employees.RemoveRange(usersForRemoving);

SpecclotheContext.GetContext().SaveChanges();

MessageBox.Show("Данные удалены");

DGrid.ItemsSource = SpecclotheContext.GetContext().Employees.ToList();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message.ToString());

}

}

Функция фильтрации записей базы данных:

try

{

if (Combo.SelectedValue.ToString() == "Виды")

{

List<Employee> del = SpecclotheContext.GetContext().Employees.ToList();

foreach (var d in del)

{

d.DepartmentsIddepartmentNavigation = SpecclotheContext.GetContext().Departments.Where(x => x.Iddepartment == d.DepartmentsIddepartment).ToList()[0];

}

DGrid.ItemsSource = del;

//TxbCountSearchItem.Text = dbISP19AEntities.GetContext().User.Count().ToString();

}

else

{

List<Employee> del = SpecclotheContext.GetContext().Employees.ToList();

foreach (var d in del)

{

d.DepartmentsIddepartmentNavigation = SpecclotheContext.GetContext().Departments.Where(x => x.Iddepartment == d.DepartmentsIddepartment).ToList()[0];

}

DGrid.ItemsSource = del;

DGrid.ItemsSource = SpecclotheContext.GetContext().Employees.

Where(x => x.DepartmentsIddepartmentNavigation.Department1 == Combo.SelectedValue.ToString()).ToList();

//TxbCountSearchItem.Text = dbISP19AEntities.GetContext().User.

// Where(x => x.LastName == CmbFiltr.SelectedValue.ToString()).Count().ToString();

}

}

catch (Exception ex) { }

Функция поиска записи:

string search = Search.Text;

if (Search.Text != null)

{

DGrid.ItemsSource = SpecclotheContext.GetContext().Employees.

Where(x => x.Fio.Contains(search)

|| x.DepartmentsIddepartmentNavigation.Department1.Contains(search)

|| x.PositionIdpositionNavigation.Posi.Contains(search)

|| x.IdEmployees.ToString().Contains(search)).ToList();

}

Функция добавления записи:

StringBuilder error = new StringBuilder(); //объект для сообщения об ошибке

//проверка полей объекта

if (string.IsNullOrWhiteSpace(\_currentemployee.Fio))

error.AppendLine("Укажите имя");

if (string.IsNullOrWhiteSpace(\_currentemployee.PositionIdposition.ToString()))

error.AppendLine("Укажите фамилию");

if (error.Length > 0)

{

MessageBox.Show(error.ToString());

return;

}

//если пользователь новый

if (\_currentemployee.IdEmployees == 0)

SpecclotheContext.GetContext().Employees.Add(\_currentemployee); //добавить в контекст

try

{

SpecclotheContext.GetContext().SaveChanges(); // сохранить изменения

// dbISP19AEntities.GetContext().ChangeTracker.Entries().ToList().ForEach(p => p.Reload());

MessageBox.Show("Данные сохранены");

manager.MainFrame.Navigate(new WorkerPage());

}

catch (Exception ex)

{

MessageBox.Show(ex.Message.ToString());

}

Функция вычисления цены с учетом скидки и даты замены:

\_currentdelivery.ClothesIdclothesNavigation = SpecclotheContext.GetContext().Clothes.Where(x => x.Idclothes == int.Parse(clothe.SelectedValue.ToString())).ToList()[0];

\_currentdelivery.EmployeesIdEmployeesNavigation = SpecclotheContext.GetContext().Employees.Where(x => x.IdEmployees == int.Parse(employee.SelectedValue.ToString())).ToList()[0];

\_currentdelivery.EmployeesIdEmployeesNavigation.PositionIdpositionNavigation = SpecclotheContext.GetContext().Positions.Where(x => x.Idposition == \_currentdelivery.EmployeesIdEmployeesNavigation.PositionIdposition).ToList()[0];

StringBuilder error = new StringBuilder(); //объект для сообщения об ошибке

\_currentdelivery.Price = \_currentdelivery.ClothesIdclothesNavigation.Price - (\_currentdelivery.ClothesIdclothesNavigation.Price \* (\_currentdelivery.EmployeesIdEmployeesNavigation.PositionIdpositionNavigation.Discount / 100));

\_currentdelivery.Datatrade =Convert.ToDateTime(\_currentdelivery.Datato).AddMonths(Convert.ToInt32(\_currentdelivery.ClothesIdclothesNavigation.Term)).Date.ToString("dd.MM.yyyy");

Функция вывода отчета:

List<Delivery> rem = DGrid.ItemsSource as List<Delivery>;

var application = new Microsoft.Office.Interop.Word.Application();

Microsoft.Office.Interop.Word.Document document = application.Documents.Add();

Microsoft.Office.Interop.Word.Paragraph paragraph = document.Paragraphs.Add();

Microsoft.Office.Interop.Word.Paragraph tableparagraph = document.Paragraphs.Add();

Microsoft.Office.Interop.Word.Range tableRange = tableparagraph.Range;

Microsoft.Office.Interop.Word.Table infoTable = document.Tables.Add(tableRange, rem.Count + 1, 5);

infoTable.Borders.InsideLineStyle = infoTable.Borders.OutsideLineStyle

= Microsoft.Office.Interop.Word.WdLineStyle.wdLineStyleSingle;

/\* infoTable.Range.Cells.VerticalAlignment

= Microsoft.Office.Interop.Word.WdCellVerticalAlignment.wdCellAlignVerticalCenter;\*/

Microsoft.Office.Interop.Word.Range cellRange;

cellRange = infoTable.Cell(1, 1).Range;

cellRange.Text = "Дата выдачи";

cellRange = infoTable.Cell(1, 2).Range;

cellRange.Text = "Дата замены";

cellRange = infoTable.Cell(1, 3).Range;

cellRange.Text = "Одежда";

cellRange = infoTable.Cell(1, 4).Range;

cellRange.Text = "Сумма";

cellRange = infoTable.Cell(1, 5).Range;

cellRange.Text = "Получатель";

infoTable.Rows[1].Range.Bold = 1;

for (int i = 0; i < rem.Count; i++)

{

cellRange = infoTable.Cell(i + 2, 1).Range;

cellRange.Text = rem[i].Datato.ToString();

cellRange = infoTable.Cell(i + 2, 2).Range;

cellRange.Text = rem[i].Datatrade.ToString();

cellRange = infoTable.Cell(i + 2, 3).Range;

cellRange.Text = rem[i].ClothesIdclothesNavigation.Clothe1.ToString();

cellRange = infoTable.Cell(i + 2, 4).Range;

cellRange.Text = rem[i].Price.ToString();

cellRange = infoTable.Cell(i + 2, 5).Range;

cellRange.Text = rem[i].EmployeesIdEmployeesNavigation.Fio.ToString();

}

/\* tableRange.InsertParagraphAfter();\*/

tableRange.Borders.Enable = 1;

tableRange.Columns.AutoFit();

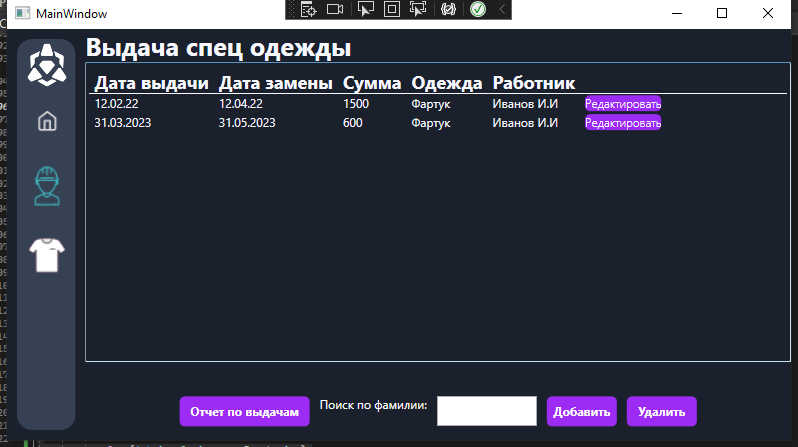
application.Visible = true;

* 1. **Текст программы**

Текст программы находится в данном репозитории - [itachineuchixa/SpecClothes2 (github.com)](https://github.com/itachineuchixa/SpecClothes2)

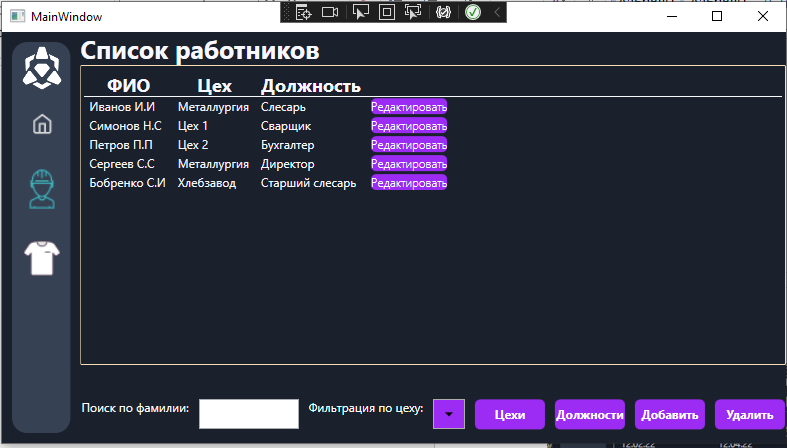
1. **Руководство пользователя**
   1. **Выполнение программного модуля**

При открытии программы появится форма главного меню, где находятся все элементы управления:



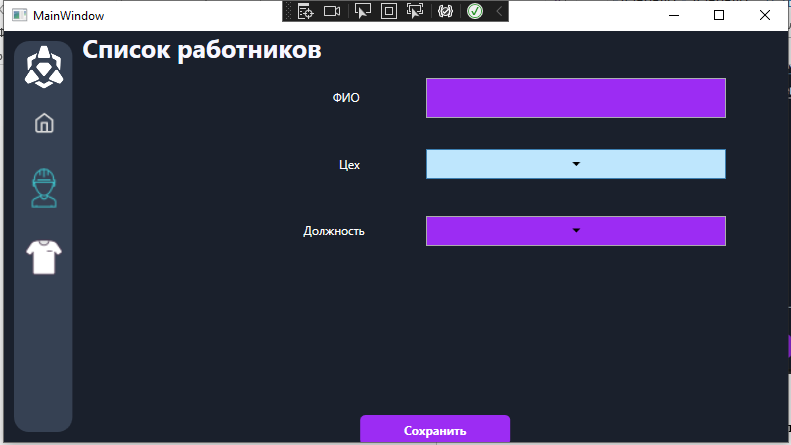
«Рис 8. Главное меню»

Окно со списком работников предприятия:



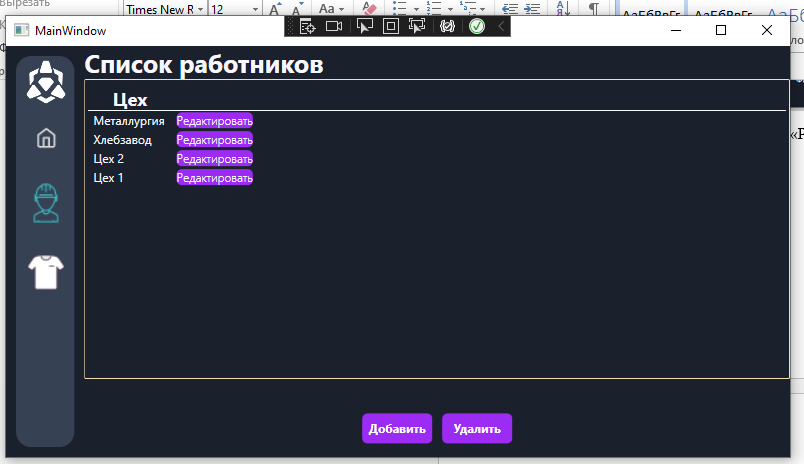
«Рис 9. Список работников»

Добавление работника:



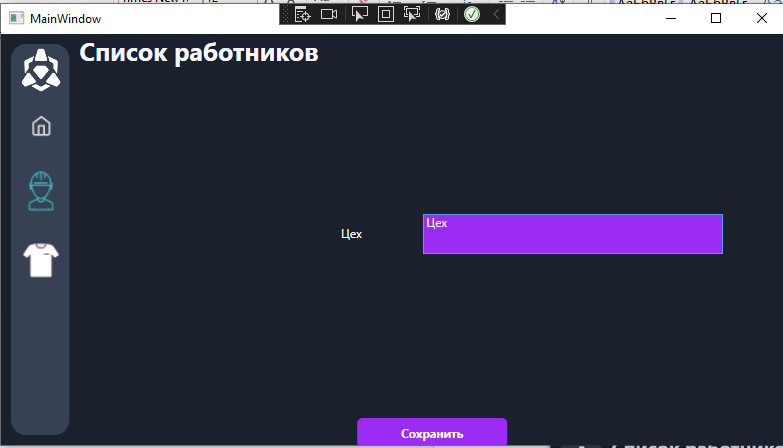
«Рис 10. Добавление работника»

Список цехов:



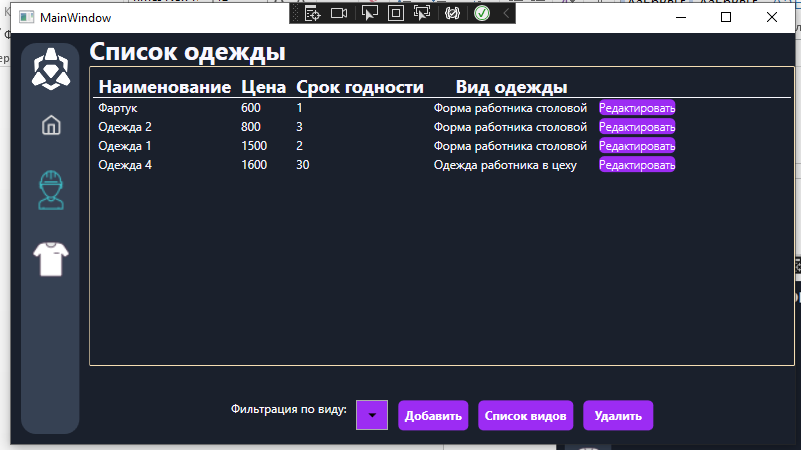
«Рис 11. Список цехов»

Добавление цеха:



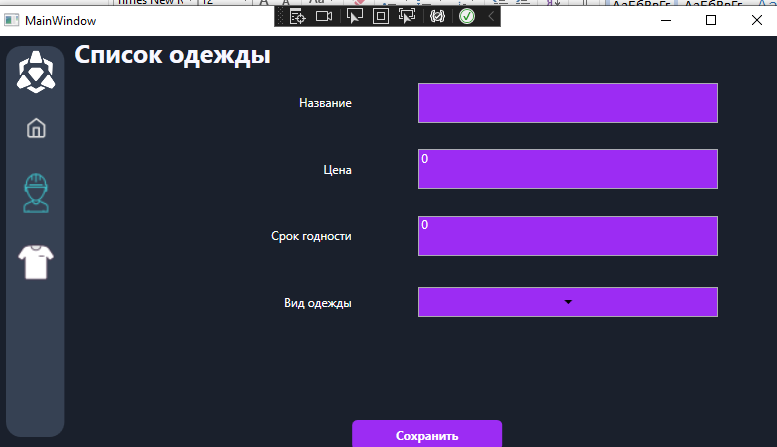
«Рис 12. Добавление цеха»

Список форменной одежды:



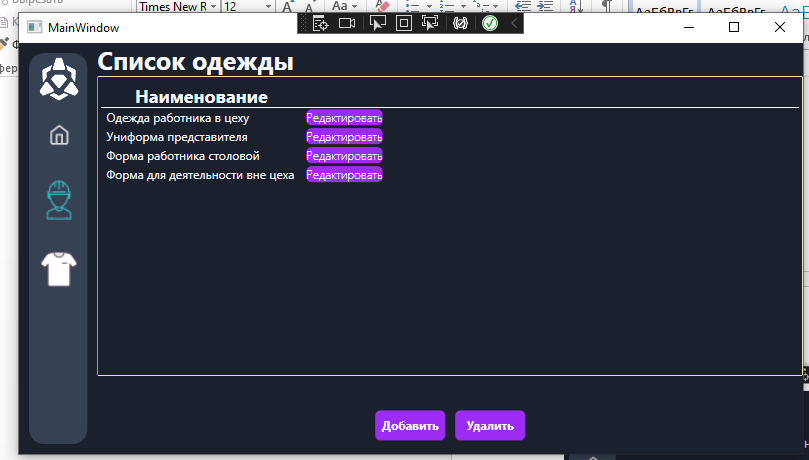
«Рис 13. Список форменной одежды»

Окно добавления одежды:



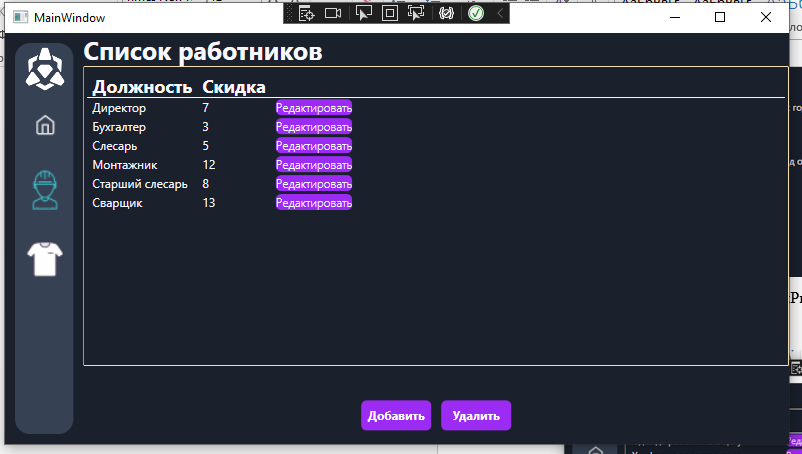
«Рис 14. Добавление одежды»

Список видов одежды:



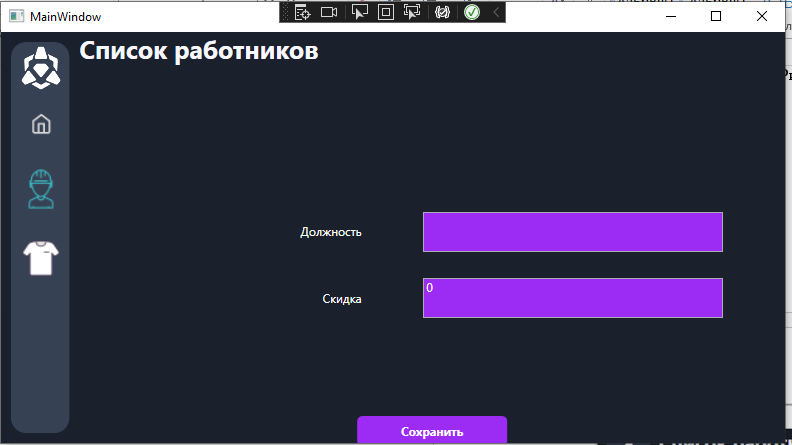
«Рис 15. Список видов одежды»

Список должностей:



«Рис 16. Список должностей»

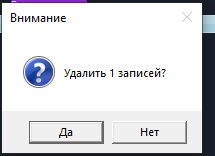
Добавление должности:



«Рис 17. Добавление должности»

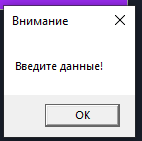
* 1. **Сообщения пользователю**

Сообщение пользователю об удалении записи:



«Рис 18. Сообщение об удалении записи»

Уведомления пользователя в случае если он не ввел данные:



«Рис 19. Уведомление пользователя»

**5**. **Тестирование программного модуля**