Министерство образования Пензенской области

Государственное автономное профессиональное

образовательное учреждение Пензенской области

«Пензенский колледж информационных и промышленных технологий (ИТ-колледж)»

**ОТЧЕТ**

**ПО ПРАКТИКЕ ПО ПРОФИЛЮ СПЕЦИАЛЬНОСТИ**

**ПМ.12 «Выполнение работ по компетенции WordSkills»**

09.02.07

«Информационные системы и программирование»

Студентки Абрашкиной Алины Алексеевны

Группа: 19ИТ17

Руководитель практики \_\_\_\_\_\_\_\_\_/ *Вьюнов Даниил Алексеевич*/

Ответственное лицо организации (базы практики) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/*Масленов Анатолий Иванович/*

*М.П.*

Пенза,2022г

**Содержание**

|  |  |
| --- | --- |
| Задание по практике по профилю специальности |  |
| Дневник по практике по профилю специальности |  |
| Аттестационный лист по практике |  |
| Приложение А Приложение к договору |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Министерство образования Пензенской области  Государственное автономное профессиональное  образовательное учреждение Пензенской области  «Пензенский колледж информационных и промышленных технологий (ИТ-колледж)»   |  |  |  | | --- | --- | --- | | *Согласовано* |  | *Утверждаю* | | *Руководитель от предприятия*  *\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Масленов А.И.* |  | *Зам. директора по работе*  *с социальными партнерами*  *\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_* *И.Н.Шипова* |   **ЗАДАНИЕ**  **ПО ПРАКТИКЕ по профилю специальности**  **ПМ.12 «Выполнение работ по компетенции WordSkills»**  09.02.07  «Информационные системы и программирование»  Абрашкина Алина Алексеевна, 19ИТ17    Задание выдал руководитель: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/ *Вьюнов Даниил Алексеевич*/  С заданием ознакомлен \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/*Абрашкина Алина Алексеевна*/  Пенза, 2022 г. |  |

***Коды формируемых компетенцией***

|  |  |
| --- | --- |
| **Код** | **Наименование результата обучения** |
| ПК 12.1 | Организация и управление работой |
| ПК 12.2 | Компетенции общения и межличностных отношений |
| ПК 12.3 | Решение проблем, инновации, креативность |
| ПК 12.4 | Анализ и проектирование программных решений |
| ПК 12.5 | Разработка программных решений |
| ПК 12.6 | Тестирование программных решений |
| ПК 12.7 | Документирование программных решений |
| ОК 1 | Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам. |
| ОК 2 | Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности |
| ОК 3 | Планировать и реализовывать собственное профессиональное и личностное развитие |
| ОК 4 | Работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами |
| ОК 5 | Осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста |
| ОК 6 | Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных общечеловеческих ценностей |
| ОК 7 | Содействовать сохранению окружающей среды, ресурсосбережению, эффективно действовать в чрезвычайных ситуациях |
| ОК 8 | Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности |
| ОК 9 | Использовать информационные технологии в профессиональной деятельности |
| ОК 10 | Пользоваться профессиональной документацией на государственном и иностранном языках |
| ОК 11 | Планировать предпринимательскую деятельность в профессиональной сфере |

**Введение**

Практика является обязательным разделом Программы подготовки специалистов среднего звена. Она представляет собой вид учебной деятельности, направленной на формирование, закрепление, развитие практических навыков и компетенции в процессе выполнения определенных видов работ, связанных с будущей профессиональной деятельностью. При реализации ППССЗ предусматриваются следующие виды практик: учебная и производственная.

Производственная практика состоит из двух этапов: практики по профилю специальности и преддипломной практики.

**Основная часть**

1. Техника безопасности

1. Общие вопросы

1.2. Запрещается находиться возле ПК в верхней одежде, принимать пищу и курить, употреблять вовремя работы алкогольные напитки, а также быть в состоянии алкогольного, наркотического или другого опьянения.

1.3. Сотрудник должен знать месторасположение первичных средств пожаротушения и уметь ими пользоваться.

1.4. О каждом несчастном случае пострадавший или очевидец несчастного случая немедленно должен известить непосредственного руководителя.

1.5. Сотрудник должен знать местонахождение медицинской аптечки, правильно пользоваться медикаментами; знать инструкцию по оказанию первой медицинской помощи пострадавшим и уметь оказать медицинскую помощь. При необходимости вызвать скорую медицинскую помощь или доставить в медицинское учреждение.

1.6. За невыполнение данной инструкции виновные привлекаются к ответственности согласно правилам внутреннего распорядка или взысканиям, определенным Кодексом законов о труде Российской Федерации.

2. Действия перед началом работы

2.1. Перед включением используемого на рабочем месте оборудования сотрудник обязан:

2.1.1. Осмотреть и привести в порядок рабочее место, убрать все посторонние предметы, которые могут отвлекать внимание и затруднять работу.

2.1.2. Проверить правильность установки стола, стула, подставки под ноги, угол наклона экрана монитора, положения клавиатуры в целях исключения неудобных поз и длительных напряжений тела. Особо обратить внимание на то, что дисплей должен находиться на расстоянии не менее 50 см от глаз (оптимально 60-70 см).

2.1.3. Проверить правильность расположения оборудования.

2.1.4. Кабели электропитания, удлинители, сетевые фильтры должны находиться с тыльной стороны рабочего места.

2.1.5. Убедиться в отсутствии засветок, отражений и бликов на экране монитора.

2.1.6. Убедиться в том, что на устройствах ПК (системный блок, монитор, клавиатура) не располагаются сосуды с жидкостями, сыпучими материалами (чай, кофе, сок, вода и пр.).

2.1.7. Включить электропитание в последовательности, установленной инструкцией по эксплуатации на оборудование; убедиться в правильном выполнении процедуры загрузки оборудования, правильных настройках.

3. Требования охраны труда во время работы

3.1. В течение всего времени работы со средствами компьютерной и оргтехники сотрудник обязан:

• содержать в порядке и чистоте рабочее место;

• следить за тем, чтобы вентиляционные отверстия устройств ничем не были закрыты;

• выполнять требования инструкции по эксплуатации оборудования;

• соблюдать, установленные расписанием, трудовым распорядком регламентированные перерывы в работе, выполнять рекомендованные физические упражнения.

3.2. Запрещается во время работы:

• отключать и подключать интерфейсные кабели периферийных устройств;

• класть на устройства средств компьютерной и оргтехники бумаги, папки и прочие посторонние предметы;

• прикасаться к задней панели системного блока (процессора) при включенном питании;

• отключать электропитание во время выполнения программы, процесса;

• допускать попадание влаги, грязи, сыпучих веществ на устройства средств компьютерной и оргтехники;

• производить самостоятельно вскрытие и ремонт оборудования;

• производить самостоятельно вскрытие и заправку картриджей принтеров или копиров;

• работать со снятыми кожухами устройств компьютерной и оргтехники;

• располагаться при работе на расстоянии менее 50 см от экрана монитора.

3.3. При работе с текстами на бумаге, листы надо располагать как можно ближе к экрану, чтобы избежать частых движений головой и глазами при переводе взгляда.

3.4. Рабочие столы следует размещать таким образом, чтобы видео дисплейные терминалы были ориентированы боковой стороной к световым проемам, чтобы естественный свет падал преимущественно слева.

3.5. Освещение не должно создавать бликов на поверхности экрана.

3.6. Продолжительность работы на ПК без регламентированных перерывов не должна превышать 1-го часа. Во время регламентированного перерыва с целью снижения нервно- эмоционального напряжения, утомления зрительного аппарата, необходимо выполнять комплексы физических упражнений.

4. Требования охраны труда в аварийных ситуациях

4.1. Обо всех неисправностях в работе оборудования и аварийных ситуациях сообщать непосредственно руководителю.

4.2. При обнаружении обрыва проводов питания или нарушения целостности их изоляции, неисправности заземления и других повреждений электрооборудования, появления запаха гари, посторонних звуков в работе оборудования и тестовых сигналов, немедленно прекратить работу и отключить питание.

4.3. При поражении пользователя электрическим током принять меры по его освобождению от действия тока путем отключения электропитания и до прибытия врача оказать потерпевшему первую медицинскую помощь.

4.4. В случае возгорания оборудования отключить питание, сообщить эксперту, позвонить в пожарную охрану, после чего приступить к тушению пожара имеющимися средствами.

5. Требования охраны труда по окончании работы

5.1. По окончании работы сотрудник обязан соблюдать следующую последовательность отключения оборудования:

• произвести завершение всех выполняемых на ПК задач;

• отключить питание в последовательности, установленной инструкцией по эксплуатации данного оборудования.

• Убрать со стола рабочие материалы и привести в порядок рабочее место.

Министерство образования Пензенской области

Государственное автономное профессиональное

образовательное учреждение Пензенской области

**«**Пензенский колледж информационных и промышленных технологий (ИТ-колледж)»

**ДНЕВНИК**

**ПО ПРАКТИКЕ ПО ПРОФИЛЮ СПЕЦИАЛЬНОСТИ**

**ПМ.12 «Выполнение работ по компетенции WordSkills»**

09.02.07

«Информационные системы и программирование»

Студентки Абрашкиной Алины Алексеевны

Группа:19ИТ17

Срок прохождения практики «11» \_\_марта\_\_\_\_ 2022 г. по «25» \_\_\_\_марта\_\_\_ 2022 г.

Место прохождения практик: ООО «Родник»

Руководитель практики: /*Масленов Анатолий Иванович*/

*М.П.*

г. Пенза

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№ п/п** | **Дата** | **Виды работ, выполняемых студентом по программе во время практики** | **Количество часов** | **Отметка о выполнении** | **Подпись наставника** |
|  | 11.03.22 | Инструктаж по технике безопасности при работе с компьютерной техникой. Анализ исходных файлов данных, проектирование на их основе структуры данных. | 6 | Выполнено |  |
|  | 14.03.22 | Приведение исходных файлов данных к виду, подходящему для импорта. Импорт данных в базу данных. | 6 | Выполнено |  |
|  | 15.03.22 | Создание настольного приложения, различных окон, таблиц, форм для заполнения, чтение и запись в базу данных. | 6 | Выполнено |  |
|  | 16.03.22 | Разработка и реализация отчетов, необходимых пользователям приложений, с графиками и возможностью вывода на печать. | 6 | Выполнено |  |
|  | 17.03.22 | Создание ERD на основе анализа предоставленных документов, проектирование архитектуры программного продукта | 6 | Выполнено |  |
|  | 18.03.22 | Интеграционное тестирование, модульное тестирование. Разработка тест-кейсов | 6 | Выполнено |  |
|  | 19.03.22 | Разработка мобильного приложения под ОС Android(API). | 6 | Выполнено |  |
|  | 21.03.22 | Разработка графического интерфейса (клиентская часть) пользователя для веб-сайта. | 6 | Выполнено |  |
|  | 22.03.22 | Создание профессиональной презентации, демонстрирующей информационную систему заказчику, и ее представление. | 6 | Выполнено |  |
|  | 23.03.22 | Создание пакета сопровождающей документации по разрабатываемой информационной системе. | 6 | Выполнено |  |
|  | 24.03.22 | Оформление решения | 6 | Выполнено |  |
|  | 25.03.22 | Дифференцированный зачет | 6 | Выполнено |  |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/*Масленов Анатолий Иванович/*

*М.П.*

**Отчёт**

Тема:Информационный ресурс «AMONIC Airlines».

Цель: разработать десктопное приложение «AMONIC Airlines».

Задачи:

– изучить научно-техническую и справочную литературу по заданной теме и выполнить анализ предметной области;

–проанализировать и уточнить требования, предъявляемые к сопровождению программного продукта;

– разработать и протестировать приложение «AMONIC Airlines»;

– сформировать требуемую техническую документацию.

Сроки: с 11 марта по 25 марта

Ход работы:

**1.1** При разработки данного приложения «AMONIC Airlines», было написано техническое задание:

1. ВВЕДЕНИЕ

Данное техническое задание предъявлено к проекту «AMONIC Airlines».

Техническое задание разработано в соответствии с ГОСТом 19.201..

2. ОСНОВАНИЕ ДЛЯ РАЗРАБОТКИ

Основанием для разработки программы является задание, выданное на экзаменационную работу преподавателем по производственной практике Вьюновым Д.А.

Наименование работы «Информационный ресурс «AMONIC Airlines»»

3. НАЗНАЧЕНИЕ

### Информационный ресурс «AMONIC Airlines» предназначен для пользователей, как эффективное и легкодоступное средство для просмотра списка рейсов, который включает изменение данных.

4.ТРЕБОВАНИЯ К ПРОГРАММЕ ИЛИ ПРОГРАММНОМУ ИЗДЕЛИЮ

**Требования к функциональным характеристикам**

### Программа должна обеспечивать возможность выполнения перечисленных ниже функций:

– авторизация;

– осуществление просмотра информации;

– поиск;

– сортировка;

– редактирование данных.

**Требования к надежности**

Программа должна быть надежной и работать бесперебойно в течение долгого времени.

**Условия эксплуатации**

Программа должна эксплуатироваться в среде под управлением операционной системы Windows 7 x64 SP1 и выше.

**Требования к информационной и программной совместимости**

Программа должна быть совместима с как можно большим количеством устройств, иметь низкие системные требования для запуска даже на самых слабых устройствах.

**Системные требования**

Так как программа довольно проста и не обладает обширным функционалом, системные требования довольно лояльны:

* процессор - Intel Pentium D 940 (или аналог от AMD) и выше;
* видеокарта - VGA 640х480 (SVGA 800x600 и более);
* оперативная память - не менее 1 Гб;
* место на накопителе - не менее 100 Мб;
* операционная система - не старше Windows 7 SP1 64-bit.

**1.2** На основании технического задания, был разработан код программы:

public partial class AuthPage : Omom  
{  
int attemp = 0;  
int startTime = 0;  
int endTime = 10;  
DispatcherTimer timer = new DispatcherTimer();  
public AuthPage()  
{  
InitializeComponent();  
timer.Interval = new TimeSpan(0,0,0,1);  
timer.Tick += timer\_Tick;  
}  
  
private void timer\_Tick(object sender, EventArgs e)  
{  
startTime++;  
TimeLabel.Content = $"До разблокировки осталось {10 - startTime} секунд";  
if (startTime >= endTime)  
{  
this.IsEnabled = true;  
timer.Stop();  
TimeLabel.Content = "";  
}  
}  
  
private void LoginButton\_Click(object sender, RoutedEventArgs e)  
{  
string login = LoginText.Text;  
string password = PasswordText.Text;  
  
try  
{  
Users user = DataHelper.GetContext().Users.Where(n => [n.Email](https://vk.com/away.php?utf=1&to=http%3A%2F%2Fn.Email) == login && n.Password == password).Single();  
if(user.RoleID == 1)  
NavigationService.Navigate(new AdministratorPage(user));  
if (user.RoleID == 2)  
{  
if ([user.Active](https://vk.com/away.php?utf=1&to=http%3A%2F%2Fuser.Active) == true)  
NavigationService.Navigate(new UserPage(user));  
else  
[MessageBox.Show](https://vk.com/away.php?utf=1&to=http%3A%2F%2FMessageBox.Show)("Вы заблокированы в системе!");  
}  
  
}  
catch  
{  
[MessageBox.Show](https://vk.com/away.php?utf=1&to=http%3A%2F%2FMessageBox.Show)("Неправильные логин или пароль!");  
attemp++;  
if (attemp >= 3)  
{  
[MessageBox.Show](https://vk.com/away.php?utf=1&to=http%3A%2F%2FMessageBox.Show)("Данные были введены неправильно три и более раза подряд. Окно будет заблокировано следующие 10 секунд.");  
timer.Start();  
this.IsEnabled = false;  
}  
}  
}  
  
private void RadioButton\_Checked(object sender, RoutedEventArgs e)  
{  
LoginText.Text = "[j.doe@amonic.com](mailto:j.doe@amonic.com)";  
PasswordText.Text = "123";  
}  
  
private void UserRadioButton\_Checked(object sender, RoutedEventArgs e)  
{  
LoginText.Text = "[a.hobart@amonic.com](mailto:a.hobart@amonic.com)";  
PasswordText.Text = "6996";  
}  
  
private void Page\_IsVisibleChanged(object sender, DependencyPropertyChangedEventArgs e)  
{  
LoginText.Text = "";  
PasswordText.Text = "";  
}  
}  
}  
}

private Users \_user = new Users();  
public AddUserForm()  
{  
InitializeComponent();  
  
OfficesComboBox.ItemsSource = DataHelper.GetContext().Offices.ToList();  
  
DataContext = \_user;  
}  
  
private void SaveButton\_Click(object sender, RoutedEventArgs e)  
{  
StringBuilder errors = new StringBuilder();  
  
if (string.IsNullOrWhiteSpace([\_user.Email](https://vk.com/away.php?utf=1&to=http%3A%2F%2F_user.Email)))  
{  
errors.AppendLine("Введите Email.");  
}  
else  
{  
string email = [\_user.Email](https://vk.com/away.php?utf=1&to=http%3A%2F%2F_user.Email);  
if (DataHelper.GetContext().Users.Select(n=>n.Email).ToList().Contains(email))  
{  
errors.AppendLine("Такой Email уже существует!");  
}  
}  
  
if (string.IsNullOrWhiteSpace(\_user.FirstName))  
errors.AppendLine("Введите First name.");  
if (string.IsNullOrWhiteSpace(\_user.LastName))  
errors.AppendLine("Введите Last name.");  
if (string.IsNullOrWhiteSpace(\_user.Password))  
errors.AppendLine("Введите Password.");  
if (BirthDatePicker.SelectedDate == null)  
errors.AppendLine("Выберите дату рождения.");  
if (OfficesComboBox.SelectedItem == null)  
errors.AppendLine("Выберите Office.");  
  
if (errors.Length > 0)  
{  
[MessageBox.Show](https://vk.com/away.php?utf=1&to=http%3A%2F%2FMessageBox.Show)(errors.ToString(), "Ошибка!", MessageBoxButton.OK, MessageBoxImage.Error);  
return;  
}  
  
try  
{  
[\_user.Active](https://vk.com/away.php?utf=1&to=http%3A%2F%2F_user.Active) = true;  
\_user.RoleID = 2;  
  
DataHelper.GetContext().Users.Add(\_user);  
DataHelper.GetContext().SaveChanges();  
}  
catch(Exception ex)  
{  
[MessageBox.Show](https://vk.com/away.php?utf=1&to=http%3A%2F%2FMessageBox.Show)(ex.Message.ToString());  
}  
}  
}  
}

public partial class UserPage : Page  
{  
private Users \_user = new Users();  
public UserPage(Users user)  
{  
InitializeComponent();  
this.\_user = user;  
TrackingHelper.track = new Tracking  
{  
UserID = [user.ID](https://vk.com/away.php?utf=1&to=http%3A%2F%2Fuser.ID),  
LoginTime = [DateTime.Now](https://vk.com/away.php?utf=1&to=http%3A%2F%2FDateTime.Now" \t "_blank),  
Crashe = true  
};  
  
var crashe = user.Tracking.Where(n=>n.Crashe == true && n.CrasheReason == null).FirstOrDefault();  
  
if (crashe != null)  
{  
CrasheReportForm crasheReportForm = new CrasheReportForm(crashe);  
crasheReportForm.ShowDialog();  
}  
  
WelcomeLabel.Content = $"Hi {user.FirstName}, welcome to AMONIC Airlines";  
TimeSpentLabel.Content = $"Time spent on system: {[user.Tracking.Select](https://vk.com/away.php?utf=1&to=http%3A%2F%2Fuser.Tracking.Select" \t "_blank)(n=>n.OnSystemTime).Sum()/60}";  
CrashesNumberLabel.Content = $"Number of crashes: {user.Tracking.Where(n=>n.Crashe == true).Count()}";  
  
TrackingData.ItemsSource = DataHelper.GetContext().Tracking.Where(n=>n.UserID == [user.ID](https://vk.com/away.php?utf=1&to=http%3A%2F%2Fuser.ID)).ToList();  
  
DataHelper.GetContext().Tracking.Add(TrackingHelper.track);  
DataHelper.GetContext().SaveChanges();  
}  
  
private void ExitItem\_Click(object sender, RoutedEventArgs e)  
{  
NavigationService.GoBack();  
}  
  
private void Page\_IsVisibleChanged(object sender, DependencyPropertyChangedEventArgs e)  
{  
if (this.IsVisible == false)  
{  
TrackingHelper.SaveTrack();  
}  
}  
  
private void BookFlightButton\_Click(object sender, RoutedEventArgs e)  
{  
NavigationService.Navigate(new BookFlightPage(\_user));  
}  
  
private void Amentities\_Click(object sender, RoutedEventArgs e)  
{  
AmentitiesForm amentitiesForm = new AmentitiesForm();  
amentitiesForm.ShowDialog();  
}  
  
private void ReportAmentities\_Click(object sender, RoutedEventArgs e)  
{  
ReportAmenitiesForm reportAmenitiesForm = new ReportAmenitiesForm();  
reportAmenitiesForm.ShowDialog();  
}  
}  
}

public partial class CrasheReportForm : Window  
{  
Tracking \_tracking = new Tracking();  
public CrasheReportForm(Tracking tracking)  
{  
InitializeComponent();  
\_tracking = tracking;  
// LogoutLabel.Content = $"No logout reason for your last login on {tracking.LoginTime.Date.ToString("dd.MM.yyyy")} at {tracking.LoginTime.ToString("hh:mm")}";  
}  
  
private void SaveButton\_Click(object sender, RoutedEventArgs e)  
{  
\_tracking.CrasheReason = ReasonText.Text;  
DataHelper.GetContext().SaveChanges();  
this.Close();  
}  
}

public partial class ChangeRoleForm : Window  
{  
private Users \_user = new Users();  
public ChangeRoleForm(Users user)  
{  
InitializeComponent();  
\_user = user;  
OfficeComboBox.ItemsSource = DataHelper.GetContext().Offices.ToList();  
DataContext = user;  
  
if (user.RoleID == 1)  
{  
AdministratorRadio.IsChecked = true;  
}  
else  
{  
UserRadio.IsChecked = true;  
}  
}  
  
private void SaveButton\_Click(object sender, RoutedEventArgs e)  
{  
if (AdministratorRadio.IsChecked == true)  
{  
\_user.RoleID = 1;  
}  
else  
{  
\_user.RoleID = 2;  
}  
  
DataHelper.GetContext().SaveChanges();  
this.Close();  
}  
  
private void CancelButton\_Click(object sender, RoutedEventArgs e)  
{  
this.Close();  
}  
}  
}

public partial class ManageSchedulesPage : Page  
{  
public ManageSchedulesPage(Users user)  
{  
InitializeComponent();  
ReloadPage();  
}  
  
public void ReloadPage()  
{  
SchedulesData.ItemsSource = DataHelper.GetContext().Schedules.ToList();  
ToAirportCombo.ItemsSource = DataHelper.GetContext().Airports.ToList();  
FromAirportCombo.ItemsSource = DataHelper.GetContext().Airports.ToList();  
  
ToAirportCombo.SelectedItem = null;  
FromAirportCombo.SelectedItem = null;  
}  
  
private void ApplyButton\_Click(object sender, RoutedEventArgs e)  
{  
var sortSchedules = DataHelper.GetContext().Schedules.ToList();  
  
var from = (Countries)FromAirportCombo.SelectedItem;  
var to = (Countries)ToAirportCombo.SelectedItem;  
  
if (ToAirportCombo.SelectedItem == FromAirportCombo.SelectedItem &&  
FromAirportCombo.SelectedItem != null && ToAirportCombo.SelectedItem != null)  
{  
[MessageBox.Show](https://vk.com/away.php?to=http%3A%2F%2FMessageBox.Show&cc_key=)("To and From can`t be equal");  
return;  
}  
  
if(FromAirportCombo.SelectedItem != null)  
sortSchedules = sortSchedules.Where(n => n.Routes.Airports.Countries == from).ToList();  
if (ToAirportCombo.SelectedItem != null)  
sortSchedules = sortSchedules.Where(n => n.Routes.Airports1.Countries == to).ToList();  
if (FlightNumText.Text != "")  
sortSchedules = sortSchedules.Where(n=>n.FlightNumber == FlightNumText.Text).ToList();  
if (OutboundDatePicker.SelectedDate != null)  
{  
var date = ((DateTime)OutboundDatePicker.SelectedDate).Date;  
sortSchedules = sortSchedules.Where(n => [n.Date](https://vk.com/away.php?to=http%3A%2F%2Fn.Date&cc_key=" \t "_blank) == date).ToList();  
}  
if (SortCombo.SelectedItem != null)  
{  
if(SortCombo.Text == "Price")  
sortSchedules = sortSchedules.OrderBy(n => n.EconomyPrice).ToList();  
if (SortCombo.Text == "Confirmed")  
sortSchedules = sortSchedules.OrderBy(n => n.Confirmed).ToList();  
if (SortCombo.Text == "Date")  
sortSchedules = sortSchedules.OrderBy(n => [n.Date](https://vk.com/away.php?to=http%3A%2F%2Fn.Date&cc_key=" \t "_blank)).ToList();  
  
}  
  
SchedulesData.ItemsSource = sortSchedules;  
}  
  
private void FlightNumText\_PreviewTextInput(object sender, TextCompositionEventArgs e)  
{  
e.Handled = "0123456789".IndexOf(e.Text) < 0;  
}  
  
private void CancelFlightButton\_Click(object sender, RoutedEventArgs e)  
{  
if (SchedulesData.SelectedItem is Schedules schedules)  
{  
schedules.Confirmed = schedules.Confirmed == true ? false : true;  
ReloadPage();  
}  
else  
{  
[MessageBox.Show](https://vk.com/away.php?to=http%3A%2F%2FMessageBox.Show&cc_key=)("Select the schedule!");  
}  
}  
  
private void EditFlightButton\_Click(object sender, RoutedEventArgs e)  
{  
if (SchedulesData.SelectedItem is Schedules schedules)  
{  
EditScheduleForm editScheduleForm = new EditScheduleForm(schedules);  
editScheduleForm.ShowDialog();  
ReloadPage();  
}  
else  
{  
[MessageBox.Show](https://vk.com/away.php?to=http%3A%2F%2FMessageBox.Show&cc_key=)("Select the schedule!");  
}  
}  
  
private void ClearButton\_Click(object sender, RoutedEventArgs e)  
{  
ReloadPage();  
}  
  
private void ImportChangesButton\_Click(object sender, RoutedEventArgs e)  
{  
ImportChangesForm importChangesForm = new ImportChangesForm();  
importChangesForm.ShowDialog();  
ReloadPage();  
}  
}  
}

public partial class ImportChangesForm : Window  
{  
List<ImportChangesHelper> changesWithoutDublicate = new List<ImportChangesHelper>();  
public ImportChangesForm()  
{  
InitializeComponent();  
}  
  
public List<ImportChangesHelper> ProcessCSV(string path)  
{  
try  
{  
return File.ReadAllLines(path).Where(row => row != null).Select(ImportChangesHelper.ParseRow).ToList();  
  
}  
catch (System.IO.IOException)  
{  
[MessageBox.Show](https://vk.com/away.php?utf=1&to=http%3A%2F%2FMessageBox.Show)("Файл занят другим процессом!");  
return null;  
}  
}  
  
private void ImportChangesButton\_Click(object sender, RoutedEventArgs e)  
{  
try  
{  
ImportResultHelper.Clear();  
  
OpenFileDialog openFileDialog = new OpenFileDialog();  
openFileDialog.Filter = "CSV Files (\*.csv)|\*.csv";  
openFileDialog.ShowDialog();  
var changes = ProcessCSV(openFileDialog.FileName);  
changesWithoutDublicate = changes.Distinct().ToList();  
  
foreach (var schedule in changesWithoutDublicate)  
{  
if (schedule == null || schedule.Result != "OK") continue;  
  
Schedules schedules = new Schedules()  
{  
AircraftID = schedule.AircraftID,  
Confirmed = true,  
Date = [schedule.Date](https://vk.com/away.php?utf=1&to=http%3A%2F%2Fschedule.Date" \t "_blank),  
Time = schedule.Time,  
EconomyPrice = schedule.BasePrice,  
RouteID = DataHelper.GetContext().Routes.Where(n => n.Airports.IATACode == schedule.From  
&& n.Airports1.IATACode == [schedule.To](https://vk.com/away.php?utf=1&to=http%3A%2F%2Fschedule.To" \t "_blank)).Select(n => [n.ID](https://vk.com/away.php?utf=1&to=http%3A%2F%2Fn.ID)).First(),  
FlightNumber = schedule.FlightNumber.ToString(),  
};  
  
if (DataHelper.GetContext().Schedules.ToList().Contains(schedules))  
{  
ImportResultHelper.Dublicate++;  
continue;  
}  
  
if (schedule.Function == "ADD")  
{  
ImportResultHelper.Successful++;  
DataHelper.GetContext().Schedules.Add(schedules);  
}  
  
}  
DataHelper.GetContext().SaveChanges();  
}  
catch (Exception ex)  
{  
[MessageBox.Show](https://vk.com/away.php?utf=1&to=http%3A%2F%2FMessageBox.Show)(ex.Message);  
}  
  
SuccessfulLabel.Content = ImportResultHelper.Successful;  
DublicateLabel.Content = ImportResultHelper.Dublicate;  
RecordLabel.Content = ImportResultHelper.MissFields;



Рисунок 1 – Авторизация

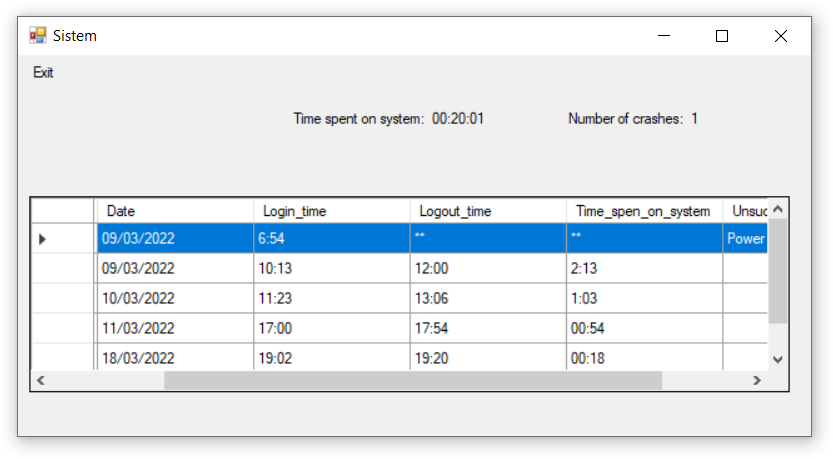


Рисунок 2 – Форма пользователя

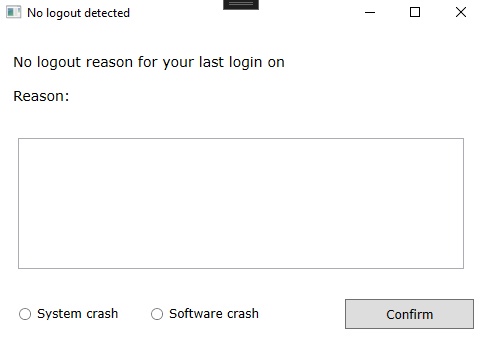


Рисунок 3 – Репорт от пользователя на краш

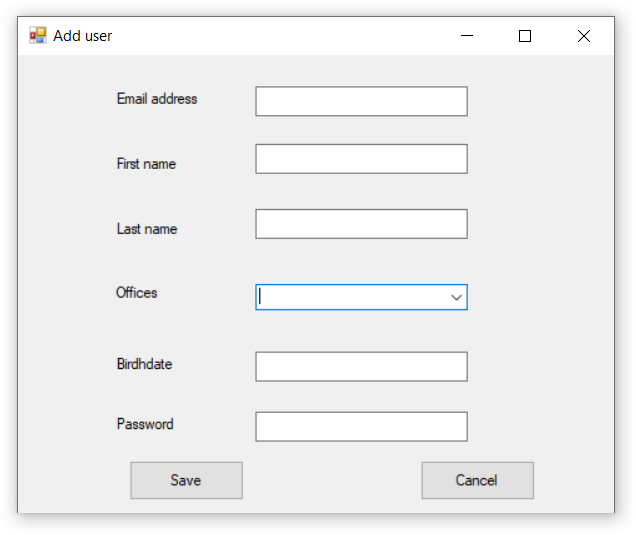


Рисунок 4 – Добавление пользователя

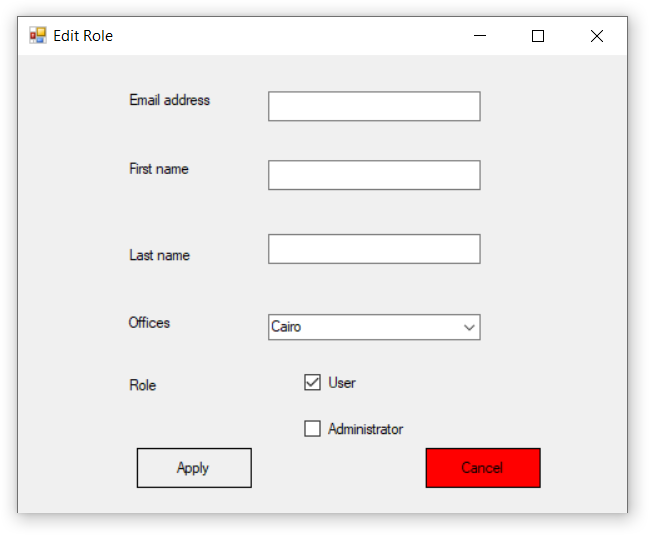


Рисунок 5 – Изменение роли пользователя

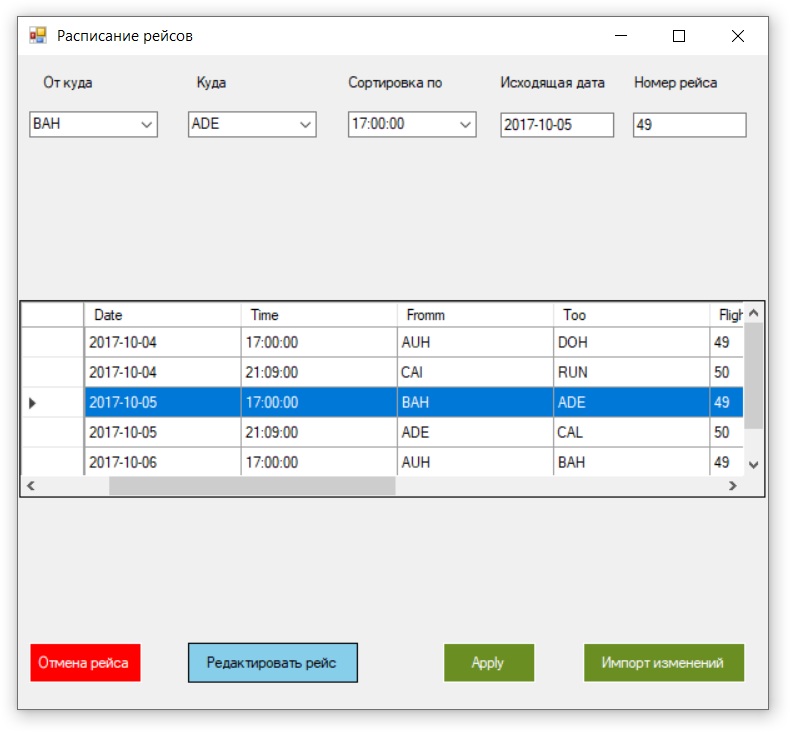


Рисунок 6 – Управление расписанием

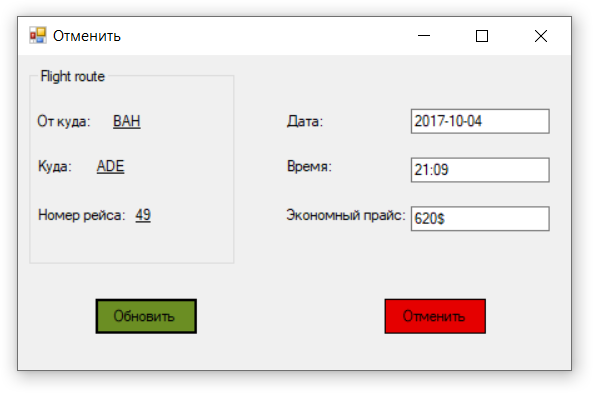


Рисунок 7 – Редактор рейса

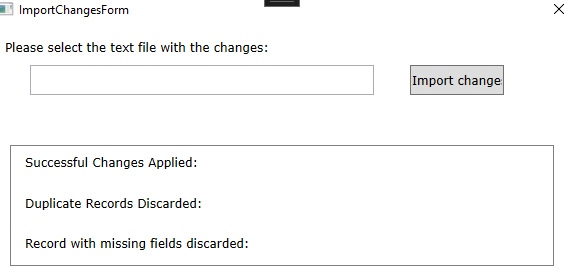


Рисунок 8 – Импорт изменений

**1.3** ER-диаграмма:

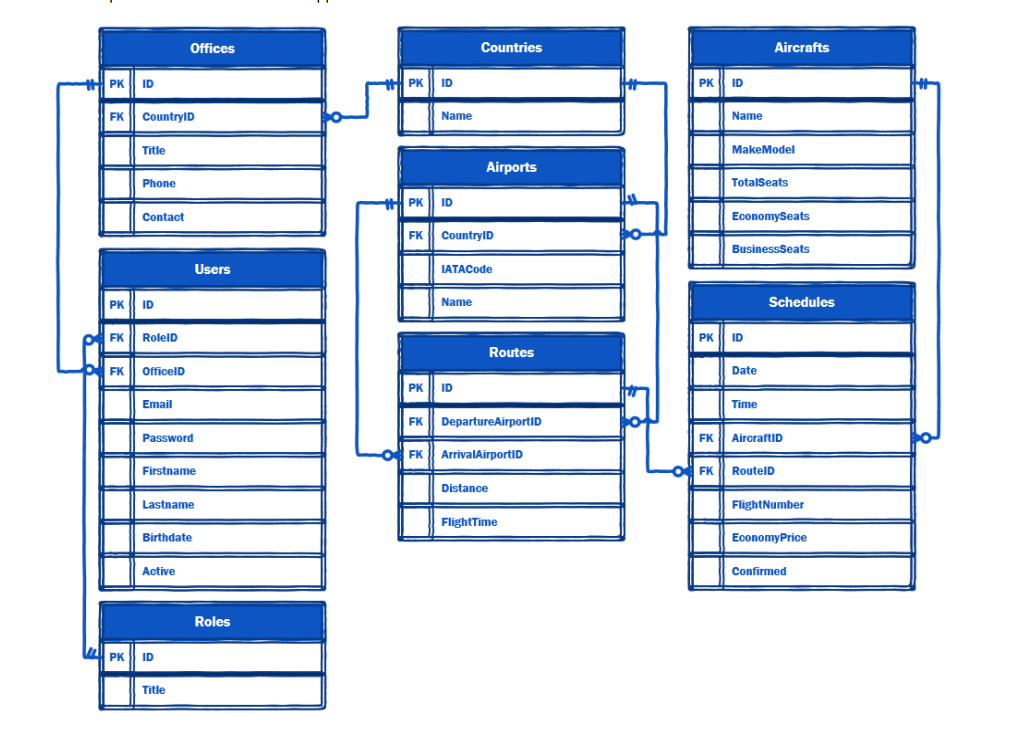
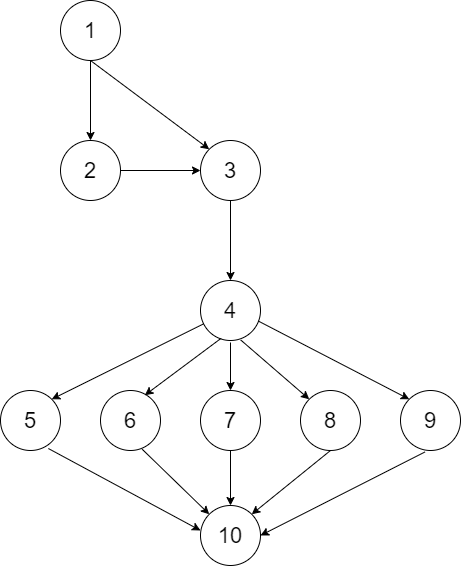
****

Рисунок 8 – ER-диаграмма

**1.4** Тестирование программы:

Был разработан тестовый граф

  
Рисунок 9 – Тестовый граф

1. Открытие программы
2. Регистрация
3. Авторизация
4. Переход на главную форму
5. Просмотр данных
6. Добавление данных
7. Удаление данных
8. Поиск данных
9. Фильтрация данных
10. Выход

Для работы по тестовому графу необходимо составить тестовые пути, по которым будет тестироваться программа. Тестовые пути по разработанной программе выглядят следующим образом:

Тестовый путь 1: (1-2-3-4-5-10)

Тестовый путь 2: (1-3-4-7-10)

**Аттестационный лист-характеристика по практике**

**по профилю специальности**

**Абрашкиной Алины Алексеевны**

обучающейся на 3 курсе по специальности СПО 09.02.07 «Информационные системы и программирование» успешно прошла производственную практику по профессиональному модуль **ПМ.12 «Выполнение работ по компетенции WordSkills»** в объеме 72 часов с «11» марта 2022 г. по «25» марта 2022 г. в организации ООО «Родник»

**Виды и качество выполнения работ:**

|  |  |
| --- | --- |
| **Виды и объем работ, выполненных обучающимся во время практики** | **Качество выполнения работ в соответствии с технологией и (или) требованиями организации, в которой проходила практика** |
| Подключение и адаптация конфигурации в соответствии с требованиями заказчика | Отлично |
| Добавление объектов метаданных в объекты конфигурации Создание обработки загрузки данных из файлов | Отлично |
| Реализация дополнительного функционала для каркасной учетной системы | Отлично |
| Создание нескольких тестовых сценариев для каркасной учетной системы | Хорошо |
| Подготовка к презентации и презентация разработанного программного решения заказчику | Отлично |

**Характеристика учебной и профессиональной деятельности обучающейся во время производственной практики**

За период прохождения производственной (по профилю специальности) практики обучающаяся посетила 11 дней, по уважительной причине отсутствовала 0 дней, пропуски без уважительной причины составили 0 дней.

Нарушения трудовой дисциплины, правил техники безопасности и производственной санитарии, а также правил внутреннего распорядка со стороны практиканта не отмечены.

В процессе прохождения производственной практики Абрашкина Алина Алексеевна проявила себя как работница исполнительная, ответственно относится к порученным заданиям. Стремится приобретать новые знания и грамотно использовать имеющиеся.

По итогам прохождения производственной практики по **ПМ.12 «Выполнение работ по компетенции «World Skills"**

студентка Абрашкина Алина Алексеевна

заслуживает оценку уровня профессиональных компетенций 5(отлично).

Руководитель практики \_\_\_\_\_\_\_\_\_ / *Вьюнов Даниил Алексеевич*/

«СОГЛАСОВАНО»

Ответственное лицо организации (базы практики)\_\_\_\_\_\_\_\_\_\_\_\_\_\_/*Масленов Анатолий Иванович/*

*М.П.*