

## Содержание

ВВЕДЕНИЕ .....	4
1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ .....	5
2 ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ .....	12
2.1 Построение ER-диаграммы .....	12
2.2 Построение схемы реляционной базы данных .....	13
2.3 Описание физической модели данных.....	15
2.4 Эскиз интерфейса .....	18
3 ОПИСАНИЕ ПРОГРАММНОЙ РЕАЛИЗАЦИИ.....	20
3.1 Описание структуры базы данных.....	20
3.2 Описание интерфейса пользователя и задач автоматизации .....	23
ВЫВОДЫ .....	40
Список литературы .....	41

## ВВЕДЕНИЕ

База данных (БД) — это организованная структура, предназначенная для хранения, изменения и обработки взаимосвязанной информации, преимущественно больших объемов. Базы данных активно используются для динамических сайтов со значительными объемами данных — часто это интернет-магазины, порталы, корпоративные сайты. Такие сайты обычно разработаны с помощью серверного языка программирования (как пример, PHP) или на основе CMS (как пример, WordPress), и не имеют готовых страничек с данными по аналогии с HTML-сайтами. Странички динамических сайтов формируются «на лету» в результате взаимодействия скриптов и баз данных после соответствующего запроса клиента к веб-серверу.

### Система управления базами данных

В контексте баз данных стоит рассмотреть понятие СУБД. Система управления базами данных (СУБД) — это комплекс программных средств, необходимых для создания структуры новой базы, ее наполнения, редактирования содержимого и отображения информации. Наиболее распространенными СУБД являются MySQL, PostgreSQL, Oracle, Microsoft SQL Server. Аренда виртуального хостинга от HOSTiQ.ua предполагает использование MariaDB — ответвление СУБД MySQL, а также PostgreSQL. Если же, например, вы планируете купить VPS или сервер в Европе или США, то вы сами сможете определить ПО для ваших баз данных.

Это СУБД примеры типа клиент-сервер, именно такие СУБД встречаются чаще всего в контексте понятия хостинга. Их особенности:

- расположение СУБД на сервере с базами данных;
- непосредственный доступ к БД;
- централизованная обработка клиентских запросов на обработку данных;
- высокий уровень надежности, доступности и безопасности;
- повышенная нагрузка на сервер.

## 1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

В курсовой работе необходимо разработать базу данных на тему “Учебная часть. Студенты”. База данных создаётся для снижения времени обработки информации и облегчения работы сотрудникам в учебной части. В каждом учебном заведении огромный поток данных, требующий компьютеризировать и систематизировать до наиболее комфортного использования.

В саму цель курсовой работы входит:

- Закрепление и систематизирование ранее полученных знаний по дисциплине “База данных”
- Закрепление и систематизирование ранее полученных знаний по дисциплине “Человеко-машинный интерфейс”
- Систематизирование приложения для функционирования автоматизированной информационной системы и реализации программного продукта для автоматизации процесса в учебной части ВУЗа.

Данное решение очень полезно для учебной части, так как позволяет следить сразу за всей успеваемостью студентов, и всеми их связывающими.

Разграничение доступа базы данных отсутствует, только администратор, в роли которого выступают работники учебной части.

В учебной части ВУЗа ведётся учёт студентов. Учёту подлежат студенты, студенческие группы, специальности, дисциплины, успеваемость студентов – оценки, получаемые студентами при сдаче экзаменов/ зачётов/ дифференцированных зачётов. Так же данный программный продукт должен производить учёт преподавателей и кафедр – комиссий, к которым они подлежат.

БД должна обеспечить хранение следующих объектов: Дисциплина, Группа, Кафедра, Преподаватель, Специальность, Студент, Успеваемость.

В соответствии с предметной областью, система строится с учётом следующих особенностей:

- Иметь удобный, интуитивно понятный интерфейс пользователю

- БД должна хранить отчисленных студентов (выпущенных/отчисленных)
- При отчислении студента, его можно восстановить только на контракт
- Каждый студент может находиться только в одной группе
- Каждая зачётная ведомость оформляется на одну группу
- Несколько преподавателей могут вести одну и ту же дисциплину у разных групп
- Абсолютно любой преподаватель, не смотря на ссылку к определённой комиссии, может принимать экзамен/зачёт/дифференцированный зачёт у любой группы, по любой дисциплине по причине экстренных случаев.
- Для разных отделений специальностей могут существовать как разные, так и одинаковые дисциплины.

Распишем базовые объекты этой предметной области.

Таблица “Студент” – это таблица с данными об учащихся в ВУЗе.

Каждый студент характеризуется следующими сведениями:

- Группа
- ФИО
- Дата рождения
- Город проживания
- Адрес проживания
- Дата поступления
- Номер телефона
- Номер зачётки
- Тип обучения (Бюджет/Контракт)
- Статус (Учится, Отчислен, Выпущен)

Студенты могут иметь одинаковое ФИО, но каждый из них различается по своему уникальному номеру зачётки.

Таблица “Дисциплина” – это таблица с данными об дисциплинах, преподаваемых в ВУЗе.

Каждая дисциплина характеризуется следующими сведениями:

- Шифр
- Название дисциплины
- Количество часов
- Количество лабораторных
- Семестр
- Тип оценивания дисциплины (Экзамен, Дифференцированный зачёт, Зачёт)

Дисциплины могут иметь одинаковое название, тип оценивания но, при этом, иметь разный семестр.

Таблица “Группы” – это таблица с данными об студенческих группах которые есть в ВУЗе.

Каждая студенческая группа характеризуется следующими сведениями:

- Специальность (Сокр. название)
- Курс
- Номер группы
- Год образования группы
- Классный руководитель
- Статус группы (есть/выпустилась)

Каждая группа уникальна по совокупности всех сведений. Могут повторяться номера, года образования, курс, специальность, статус группы. Но никак всё вместе.

Таблица “Успеваемость” – это таблица с данными об успеваемости, оценках студентов, которые учатся/учились в ВУЗе.

Каждая оценка характеризуется следующими сведениями:

- ФИО студента
- ФИО преподавателя
- Название дисциплины
- Дата оценивания
- Оценка

Таблица “Специальность” – это таблица с данными обо всех специальностях, которые есть в ВУЗе.

Каждая специальность характеризуется следующими сведениями:

- Полное название
- Короткое название (шифр из двух букв)
- Заведующий

Таблица “Преподаватель” – это таблица с данными о преподавателях, работающих в ВУЗе.

Каждый преподаватель характеризуется следующими сведениями:

- ФИО
- Категория (1-ая, 2-ая, 3-я)
- Принадлежность к определённой кафедре (комиссии)

Больше нам ничего не требуется учитывать от преподавателей, так как нам важна только его ФИО при учёте успеваемости студентов. Соответственно, если у нас будут одинаковые по ФИО преподаватели, то в базе данных будет храниться один из них для меньшего занятия ресурсов.

Таблица “Кафедра” – это таблица с данными о всех кафедрах, комиссиях в ВУЗе.

Каждая кафедра характеризуется следующими сведениями:

- Название
- Номер телефона
- Заведующий

В программном продукте необходимо учесть следующие особенности:

1. Недопустим ввод цифр в полях:

- ФИО студента
- Город проживания студента
- Тип обучения студента (Бюджет/Контракт)
- Название дисциплины

- Оценивание дисциплины (Зачёт, Экзамен, Дифференцированный зачёт)
- ФИО преподавателя
- Название кафедры
- Название и шифр специальности

2. Недопустим ввод букв и отрицательных чисел в полях:

- Номер телефона студента
- Номер зачётки студента
- Дата рождения студента
- Дата поступления студента в ВУЗ
- Шифр дисциплины
- Семестр преподавания дисциплины
- Количество часов дисциплины
- Количество лабораторных работ дисциплины
- Оценка для студента по дисциплине
- Дата оценивания успеваемости
- Номер группы
- Курс группы
- Год образования группы
- Категория преподавателя
- Номер телефона кафедры

3. Формат даты рождения студентов, даты их поступления и даты аттестации – ДД/ММ/ГГГГ.

4. Дата рождения студентов находится в диапазоне от 35 лет до 14 от текущей даты. (Текущий год -35 и -14 лет).

5. Так же, недопустимо любое пустое значение полей при добавлении их в базу данных для большей точности и откидывания всевозможных конфликтов.

6. Поле “Номер зачётки” – уникальное для каждого студента.

7. Поля “Название предмета” и “Семестр” в совокупности являются уникальными для каждой дисциплины.
8. Поля “Номер зачётки студента”, “Название предмета”, “Семестр” в совокупности являются уникальными для каждой оценки.
9. Поля “Шифр специальности”, “Курс”, “Номер”, “Год образования” в совокупности являются уникальными для каждой группы.
10. Поле “ФИО преподавателя” является уникальным для каждого преподавателя.
11. Поле “Название кафедры” является уникальным для каждой кафедры.
12. Поле “Шифр” и “Название” по отдельности являются уникальными для каждой специальности.

Информационная система для работы с БД должна поддерживать следующие функции:

1. Просмотр, добавление, редактирование, удаление информации в БД.
2. Просмотр каждой информации в более удобном окошке с дальнейшей печатью.
3. Поиск студентов по заданным параметрам (ФИО, Номер зачётки, Группа, Тип обучения (Бюджет/Контракт), Год рождения).
4. Отображение количества бюджетников, контрактников.
5. Показ отчисленных студентов.
6. Возможность восстановить отчисленного студента на первый курс на контрактной основе.
7. Поиск дисциплин по заданным параметрам (Название, Оценивание (Экзамен, Зачёт, Дифференцированный зачёт), Семестр (1-8)).
8. Отображение суммарного количества дисциплин и за каждый семестр.
9. Поиск успеваемости по заданным параметрам (ФИО студента, Номер зачётки, ФИО преподавателя, Название дисциплины, Семестр (1-8), Оценивание (Экзамен, Зачёт, Дифференцированный зачёт)).
10. Отображение количества всех оценок и оценок от 1 до 5 и подсчёта среднего балла в успеваемости.



11. Возможность просмотра оценок отчисленных студентов.
12. Поиск успеваемости отчисленных студентов по тем же параметрам.
13. Отображение количества всех оценок и оценок от 1 до 5 и подсчёта среднего балла в успеваемости отчисленных студентов.
14. Поиск группы по заданным параметрам (Шифр, Курс, Классный руководитель).
15. Отображение количества групп.
16. Возможность перевода всех групп на следующий курс (Возможно только раз в году, летом).
17. Возможность просмотра выпущенных групп.
18. Поиск преподавателей по заданным параметрам (ФИО).
19. Отображение количества всех преподавателей и отдельно каждой категории.
20. Поиск кафедры по заданным параметрам (Название кафедры).
21. Отображение количества кафедр.
22. Поиск специальности по заданным параметрам (Название специальности).
23. Отображение количества специальностей.

Заключительным шагом является установление соответствия между сущностями, характеристиками предметной областями, отношениями и атрибутами в нотации выбранной СУБД.

## 2 ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ

### 2.1 Построение ER-диаграммы

В ходе анализа предметной области выделены такие сущности:

- Специальность
- Преподаватель
- Кафедра
- Дисциплина
- Студент

Каждый экземпляр сущности “Студент” будет соответствовать конкретному студенту. В ВУЗе каждому учащемуся присваивается уникальный номер зачётной книги, который и будет однозначно идентифицировать студента.

Студенческие группы формируются из множества студентов. Поэтому целесообразно соединить “Студент” и “Группа” по связи один-ко-многим.

Аналогично, с сущностями “Преподаватель” и “Кафедра”, связь между ними будет один-ко-многим, “Специальность” и “Группа”, “Дисциплина” и “Успеваемость”, “Студент” и “Успеваемость”, “Преподаватель” и “Успеваемость”.

ER-диаграмма, на основе выявленных ранее сущностей и связей между ними представлена на Рисунке 2.1

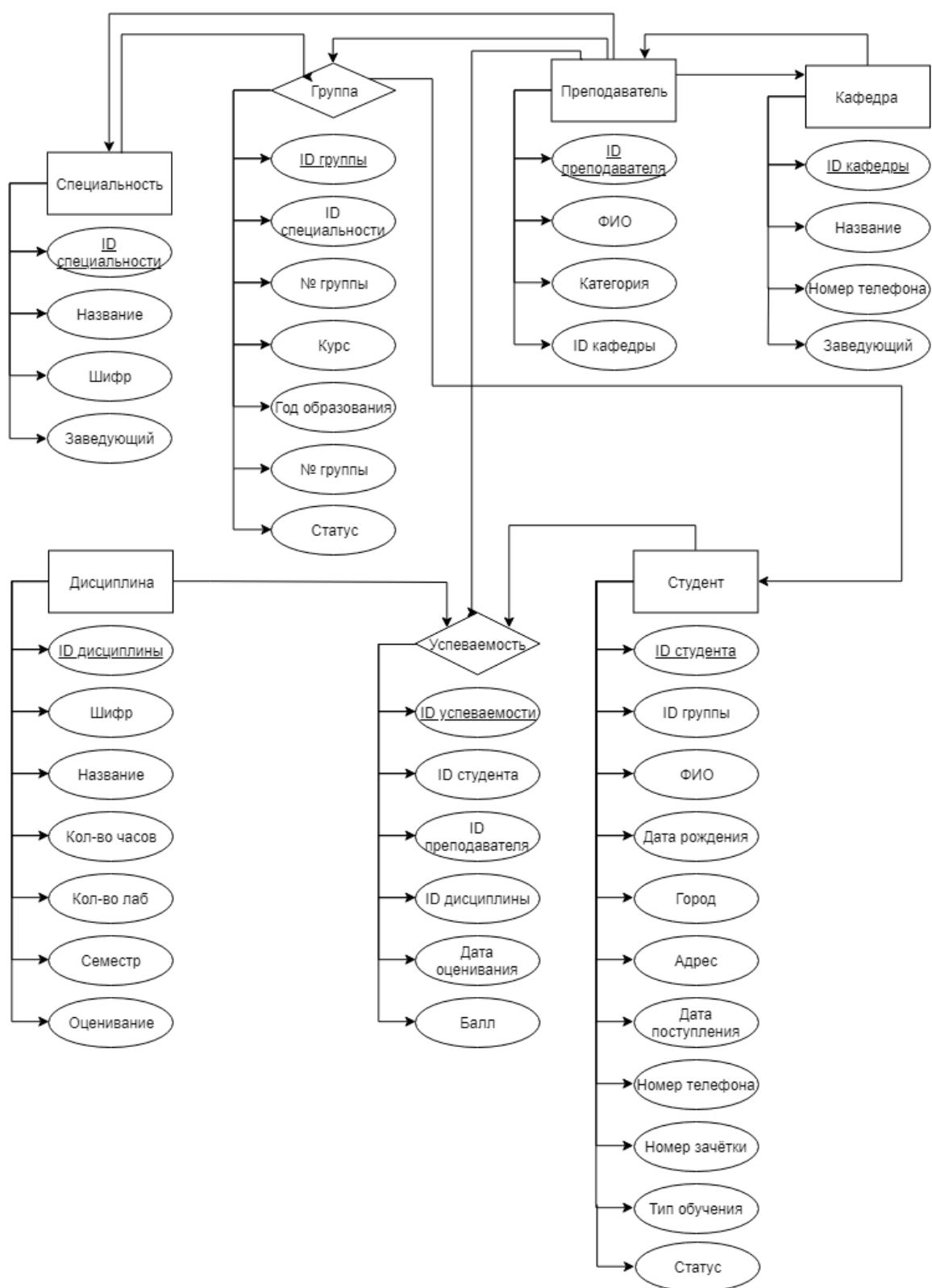


Рисунок 2. 1 - ER-диаграмма

## 2.2 Построение схемы реляционной базы данных

Преобразуем сущности ER-модели в реляционные отношения (таблицы 2.1 – 2.7).

Таблица 2. 1 - Специальность

ID специальности	Название	Шифр	ID преподавателя

Таблица 2. 2 - Кафедра

ID кафедры	Название	Номер	ID преподавателя

Таблица 2. 3 - Дисциплина

ID дисциплины	Шифр	Название	Кол-во часов	Кол-во лаб	Семестр	Оценивание

Таблица 2. 4 - Студент

ID студента	ID группы	ФИО	Дата рождения	Город	Адрес	Дата поступления	Номер телефона	Номер зачётки	Тип обучения	Статус

Таблица 2. 5 - Преподаватель

ID преподавателя	ФИО	Категория	ID кафедры

Таблица 2. 6 - Группа

ID группы	ID специальности	Номер группы	Курс	Год образования	Кл. рук.	Статус

Таблица 2. 7 - Успеваемость

ID успеваемости	ID студента	ID преподавателя	ID дисциплины	Дата оценивания	Балл

### 2.3 Описание физической модели данных

В качестве СУБД была выбрана MySQL и визуальное приложение для работы с ней Workbench.

Первое, что стоит отметить – Workbench дает возможность визуально проектировать базу данных, т.е. составлять схему БД. Визуальное представление БД всегда дает куда большую информацию, чем сухой список таблиц. В таком варианте сразу видно, каким образом связаны между собой таблицы, можно группировать таблицы по каким-либо параметрам и отражать это на схеме. Во-вторых, программа имеет встроенный редактор SQL-кода, с помощью которого можно быстро внести любые правки в SQL-запросы. При этом возможно строить запросы любой сложности, получать различные выборки из таблиц, связывать их, создавать новые таблицы и редактировать существующие, работать с ключами, полями, записями.

В качестве языка был выбран C#. Выделяются такие достоинства данного языка:

1. C# является полностью объектно-ориентированным языком, где даже типы, встроенные в язык, представлены классами.
2. Реализация, сочетающая построение надежного и эффективного кода, является немаловажным фактором, способствующему успеху C#.

3. Мощная библиотека каркаса поддерживает удобство построения различных типов приложения на C#, позволяя легко строить Web-службы, другие виды компонентов, достаточно просто сохранять и получать информацию из базы данных и других хранилищ данных.

Ниже представлена физическая модель данных, которая содержит все детали необходимые СУБД для создания базы данных:

- Наименование таблиц и столбцов
- Типы данных
- Определение первичных и внешних ключей
- Ограничения

По созданным реляционным отношениям была создана физическая модель базы данных (таблицы 2.8 – 2.14).

Таблица 2.8 - Группа

<b>GRYP</b>			
Id	INT(11)	PRIMARY KEY, NOT NULL	ID группы
Sp_id	INT(11)	FOREGIN KEY, NOT NULL	Код специальности
G_kyrs	INT(1)	NOT NULL	Курс
G_numer	INT(1)	NOT NULL	Номер
G_born	INT(1)	NOT NULL	Год образования
P_id	INT(11)	FOREGIGN KEY, NOT NULL	Код преподавателя
G_status	INT(1)	NOT NULL	Статус

Таблица 2. 9 - Студент

<b>STYDENT</b>			
Id	INT(11)	PRIMARY KEY, NOT NULL	ID студента
G_id	INT(11)	FOREIGN KEY, NOT NULL	Код группы
St_fio	VARCHAR(50)	NOT NULL	ФИО
St_data	DATE	NOT NULL	Дата рождения
St_town	VARCHAR(50)	NOT NULL	Город
St_adres	VARCHAR(100)	NOT NULL	Адрес
St_postyp	DATE	NOT NULL	Дата поступления
St_nomer	VARCHAR(10)	NOT NULL	Номер телефона
St_zach	INT(7)	NOT NULL	Номер зачётки
St_opl	VARCHAR(20)	NOT NULL	Тип обучения
St_status	INT(11)	NOT NULL	Статус

Таблица 2. 10 - Преподаватель

<b>PREPOD</b>			
Id	INT(11)	PRIMARY KEY, NOT NULL	ID преподавателя
P_fio	VARCHAR(50)	NOT NULL	ФИО
P_kategory	VARCHAR(10)	NOT NULL	Категория
K_id	INT(11)	FOREIGN KEY, NOT NULL	Код кафедры

Таблица 2. 11 - Кафедра

<b>KAFEDRA</b>			
Id	INT(11)	PRIMARY KEY, NOT NULL	ID кафедры
K_name	VARCHAR(100)	NOT NULL	Название
K_nomer	VARCHAR(10)	NOT NULL	Номер телефона
P_id	INT(11)	FOREGIN KEY, NOT NULL	Код преподавателя

Таблица 2. 12 - Специальность

<b>SPECLY</b>			
Id	INT(11)	PRIMARY KEY, NOT NULL	ID специальности
Sp_name	VARCHAR(50)	NOT NULL	Название
Sp_nazv	VARCHAR(2)	NOT NULL	Шифр
P_id	INT(11)	FOREIGN KEY, NOT NULL	Код преподавателя

Таблица 2. 8 - Дисциплина

<b>DISP</b>			
Id	INT(11)	PRIMARY KEY, NOT NULL	ID дисциплины
D_shifr	INT(3)	NOT NULL	Шифр
D_name	VARCHAR(40)	NOT NULL	Название
D_norm	INT(3)	NOT NULL	Кол-во часов
D_lab	INT(3)	NOT NULL	Кол-во л/р
D_sem	INT(1)	NOT NULL	Семестр
D_type	VARCHAR(40)	NOT NULL	Тип оценивания

Таблица 2.14 - Успеваемость

<b>YSPEH</b>			
Id	INT(11)	PRIMARY KEY, NOT NULL	ID оценки
St_id	INT(11)	FOREIGN KEY, NOT NULL	Код студента
P_id	INT(11)	FOREIGN KEY, NOT NULL	Код преподавателя
D_id	INT(11)	FOREIGN KEY, NOT NULL	Код дисциплины
Y_data	DATE	NOT NULL	Дата оценивания
Y_ball	INT(1)	NOT NULL	Оценка

## 2.4 Эскиз интерфейса

Эскиз интерфейса - это визуальный набросок будущего приложения. Он дает возможность сгруппировать объекты и правильно распределить функции. От удобства интерфейса зависит то, насколько клиент «привяжется» к системе, насколько ему удобно будет работать. Интерфейс приложения для работы с БД будет разработан в виде форм, панелей с кнопками, полями и переключателями. Далее на рисунках 2.2 – 2.4 представлены эскизы окон, с которыми будет работать пользователь во время эксплуатации приложения.



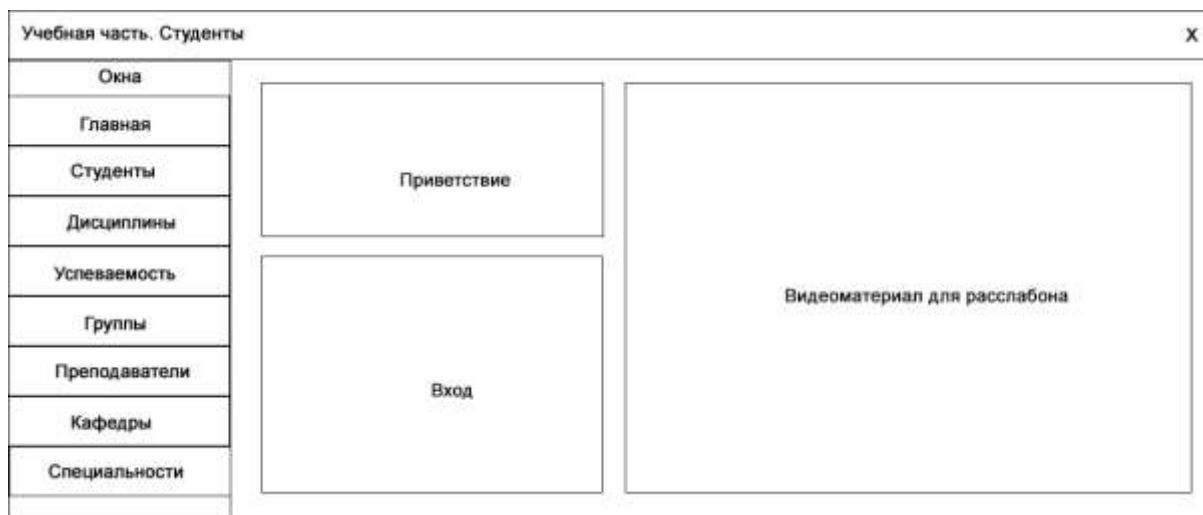


Рисунок 2. 2 – Эскиз интерфейса страницы “Главное меню”



Рисунок 2. 3 – Эскиз интерфейса страницы “Главное меню” после входа

Добавление/Изменение данных

Группа	Шифр	Курс	№	Год	Дата рождения																																										
ФИО	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<div>← апрель 2005 г. →</div> <table border="1"> <tr> <td>пн</td><td>вт</td><td>ср</td><td>чт</td><td>пт</td><td>сб</td><td>вс</td> </tr> <tr> <td>28</td><td>29</td><td>30</td><td>31</td><td>1</td><td>2</td><td>3</td> </tr> <tr> <td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td> </tr> <tr> <td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td> </tr> <tr> <td>18</td><td>19</td><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td> </tr> <tr> <td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td></td><td></td> </tr> </table>	пн	вт	ср	чт	пт	сб	вс	28	29	30	31	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29		
пн	вт	ср	чт	пт	сб	вс																																									
28	29	30	31	1	2	3																																									
4	5	6	7	8	9	10																																									
11	12	13	14	15	16	17																																									
18	19	20	21	22	23	24																																									
25	26	27	28	29																																											
Город	<input type="text"/>																																														
Адрес	<input type="text"/>																																														
Телефон	<input type="text"/>																																														
№ зачётки	<input type="text"/>																																														
Бюджет\Контракт	<input type="text"/>																																														
<input type="button" value="Добавить"/>					<input type="button" value="Отмена"/>																																										

Рисунок 2. 4 – Эскиз интерфейса для добавления, изменения данных о студентах

## 3 ОПИСАНИЕ ПРОГРАММНОЙ РЕАЛИЗАЦИИ

### 3.1 Описание структуры базы данных

В соответствии с физической моделью БД, была создана структура таблиц СУБД MySQL. Далее представлен текст из консоли MySQL-сервера, показывающий, в каком виде создавались и хранятся таблицы данной БД:

```
CREATE TABLE `disp` (  
  `Id` int(11) NOT NULL AUTO_INCREMENT,  
  `D_shifr` int(3) NOT NULL,  
  `D_name` varchar(40) NOT NULL,  
  `D_norm` int(3) NOT NULL,  
  `D_lab` int(3) NOT NULL,  
  `D_sem` int(1) NOT NULL,  
  `D_type` varchar(40) NOT NULL,  
  PRIMARY KEY (`Id`)  
) ENGINE=InnoDB AUTO_INCREMENT=9 DEFAULT CHARSET=utf8mb4  
COLLATE=utf8mb4_0900_ai_ci;
```

```
CREATE TABLE `gryp` (  
  `Id` int(11) NOT NULL AUTO_INCREMENT,  
  `Sp_id` int(11) NOT NULL,  
  `G_kyrs` int(1) NOT NULL,  
  `G_number` int(1) NOT NULL,  
  `G_born` int(1) NOT NULL,  
  `P_id` int(11) DEFAULT NULL,  
  `G_status` int(1) NOT NULL,  
  PRIMARY KEY (`Id`),  
  KEY `Sp_id_idx` (`Sp_id`),
```

```

KEY `Pr_id_idx` (`P_id`),
CONSTRAINT `Pr_id` FOREIGN KEY (`P_id`) REFERENCES `prepod` (`id`),
CONSTRAINT `Sp_id` FOREIGN KEY (`Sp_id`) REFERENCES `specly` (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=86 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci;

```

```

CREATE TABLE `kafedra` (
  `Id` int(11) NOT NULL AUTO_INCREMENT,
  `K_name` varchar(100) NOT NULL,
  `K_nomer` varchar(10) NOT NULL,
  `P_id` int(11) DEFAULT NULL,
  PRIMARY KEY (`Id`),
  KEY `Teacher_id_idx` (`P_id`),
  CONSTRAINT `Teacher_id` FOREIGN KEY (`P_id`) REFERENCES `prepod`
(`id`)
) ENGINE=InnoDB AUTO_INCREMENT=38 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci;

```

```

CREATE TABLE `prepod` (
  `Id` int(11) NOT NULL AUTO_INCREMENT,
  `P_fio` varchar(50) NOT NULL,
  `P_kategory` varchar(10) NOT NULL,
  `K_id` int(11) NOT NULL,
  PRIMARY KEY (`Id`),
  KEY `O_id_idx` (`K_id`),
  CONSTRAINT `K_id` FOREIGN KEY (`K_id`) REFERENCES `kafedra` (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=28 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci;

```

```

CREATE TABLE `specly` (

```

```

`Id` int(11) NOT NULL AUTO_INCREMENT,
`Sp_name` varchar(50) NOT NULL,
`Sp_nazv` varchar(2) NOT NULL,
`P_id` int(11) DEFAULT NULL,
PRIMARY KEY (`Id`),
KEY `Prepod_id_idx` (`P_id`),
CONSTRAINT `Prepod_id` FOREIGN KEY (`P_id`) REFERENCES `prepod`
(`id`)
) ENGINE=InnoDB AUTO_INCREMENT=21 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci;

```

```

CREATE TABLE `stydent` (
  `Id` int(11) NOT NULL AUTO_INCREMENT,
  `G_id` int(11) NOT NULL,
  `St_fio` varchar(50) NOT NULL,
  `St_data` date NOT NULL,
  `St_town` varchar(50) NOT NULL,
  `St_adres` varchar(100) NOT NULL,
  `St_postyp` date NOT NULL,
  `St_nomer` varchar(10) NOT NULL,
  `St_zach` varchar(7) NOT NULL,
  `St_opl` varchar(20) NOT NULL,
  `St_status` int(11) NOT NULL,
  PRIMARY KEY (`Id`),
  UNIQUE KEY `St_zach_UNIQUE` (`St_zach`),
  KEY `G_id_idx` (`G_id`),
  CONSTRAINT `G_id` FOREIGN KEY (`G_id`) REFERENCES `gryp` (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=10 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci;

```

```

CREATE TABLE `yspeh` (
  `Id` int(11) NOT NULL AUTO_INCREMENT,
  `St_id` int(11) NOT NULL,
  `P_id` int(11) NOT NULL,
  `D_id` int(11) NOT NULL,
  `Y_data` date NOT NULL,
  `Y_ball` int(1) NOT NULL,
  PRIMARY KEY (`Id`),
  KEY `St_id_idx` (`St_id`),
  KEY `P_id_idx` (`P_id`),
  KEY `D_id_idx` (`D_id`),
  CONSTRAINT `D_id` FOREIGN KEY (`D_id`) REFERENCES `disp` (`id`),
  CONSTRAINT `P_id` FOREIGN KEY (`P_id`) REFERENCES `prepod` (`id`),
  CONSTRAINT `St_id` FOREIGN KEY (`St_id`) REFERENCES `stydent` (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=14 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci;

```

### 3.2 Описание интерфейса пользователя и задач автоматизации

Чтобы приступить к работе с приложением, для начала нужно запустить файл UchebnayaChast.exe. При открытии данного файла, появится окно Главной страницы (рисунок 3.1).



Рисунок 3. 1 – Главное меню

Главная страница содержит панель со входом в систему, повторяющийся видеоролик и неактивные кнопки слева. Чтобы панель слева стала активной, необходимо выполнить вход, написав пароль. После входа, Главная страница будет иметь вид, отображенный на рисунке 3.2.



Рисунок 3. 2 - Главное меню после входа

Как можно заметить, после входа, панель слева стала активной. Справа сверху появился замочек, нажав на который, мы вернёмся на Главную страницу с требованием ввести пароль. Своеобразная защита от злоумышленников.

На протяжении работы всего программного продукта, сверху будет находиться панель с названием самого продукта (слева сверху) и кнопкой выхода (справа сверху).

Слева на панели мы видим несколько кнопок: Главная (Где мы сейчас находимся), “Студенты”, “Дисциплины”, “Успеваемость”, “Группы”, “Преподаватели”, “Кафедры”, “Специальности”.

Нажав на кнопку “Студенты”, появится панель, отраженная на рисунке 3.3.

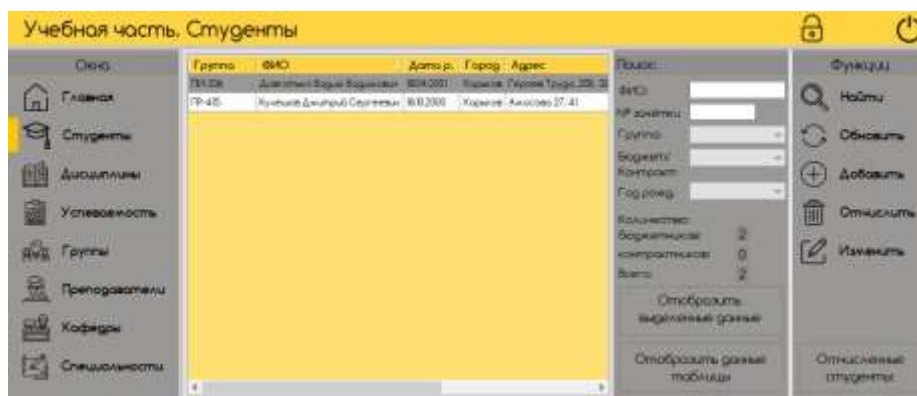


Рисунок 3. 3 - Студенты

Данная панель позволяет просматривать в реальном времени всех учащихся студентов, занесённых в базу данных. Также она позволяет выполнять поиск по требуемым параметрам (ФИО студента, Номер зачётки, Группа, Тип обучения (Бюджет/Контракт), Год рождения при помощи вспомогательной кнопки “Найти”. Отображает количество всех студентов, бюджетников и контрактников.

На данной панели есть возможность Обновить таблицу кнопкой “Обновить”, добавить новых студентов при помощи кнопки “Добавить”, изменить данные про студента при помощи кнопки “Изменить” либо же отчислить его.

При нажатии на кнопку “Отобразить выделенные данные” либо “Отобразить данные таблицы” мы сможем просмотреть информацию о студентах в более удобном виде. Попробуем, нажав на кнопку “Отобразить данные таблицы” и появится следующая форма (рисунок 3.4).

Группа	ФИО	Дата р.	Город	Адрес	Дата пост.	Телефон	№ зачётки	Б\К
ЭН-28	Александров Борис Владимирович	18.04.2000	Харьков	Григорьева Тараса, 33, 32	18.04.2000	267700000	87545	Бюджет
ЭН-48	Куницын Дмитрий Сергеевич	18.05.2000	Харьков	Алексеев 27, 41	18.05.2000	0660891800	20867	Бюджет

Скриншот на рабочий стол

Кол-во

Б:

К:

Всего:

Закрывать

Рисунок 3. 4 - Более удобный просмотр студентов

Данная форма позволяет в раскрытом виде просмотреть информацию о студентах, не вынуждая “скроллить” информацию слева-направо. Так же здесь находится такой же подсчёт количества студентов, бюджетников и контрактников.

При нажатии на кнопку “Скриншот на рабочий стол”, выполняется скрин данной области, например, для дальнейшей печати.

Вернёмся обратно, нажав на кнопку “Закрывать”:

Теперь попробуем добавить нового студента, нажав на кнопку “Добавить”. При этом появляется форма с пустыми полями для ввода новых данных (рисунок 3.5).

Добавление/Изменение данных

Группа:

ФИО:

Город:

Адрес:

Телефон:

№ зачётки:

Бюджет\Контракт:

Дата рождения

апрель 2005 г.

пн	вт	ср	чт	пт	сб	вс
28	29	30	31	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29		

Добавить
Отмена

Рисунок 3. 5 - Форма добавления/изменения данных о студенте



Данная форма вызывается так же при изменении данных о студенте только уже с заполненными полями.

Чтобы добавить/изменить данные студента, требуется заполнить все поля. Если какое-то поле останется незаполненным, то при нажатии на кнопку “Добавить”, пользователю выдаст сообщение о том, что определённое поле не заполнено, указывая какое. Поле Группы является полностью выпадающим списком, состоящим из Шифра, Номера, Курса и Года образования группы. Если выбранной группы не окажется, пользователю выдаст сообщение, что данной группы не существует и следует либо создать такую группу, либо переопределить студента в другую.

Поле ФИО может повторяться, ведь оно не является индивидуальным для каждого человека. Уникальным является для каждого студента номер зачётной книги. Если же попытаться ввести уже существующий в базе данных номер зачётной книги, пользователю выдаст сообщение, что такая зачётка уже есть.

При корректном вводе данных, студент будет успешно добавлен в базу.

И теперь панель “Студенты” будет иметь следующий вид (рисунок 3.6).

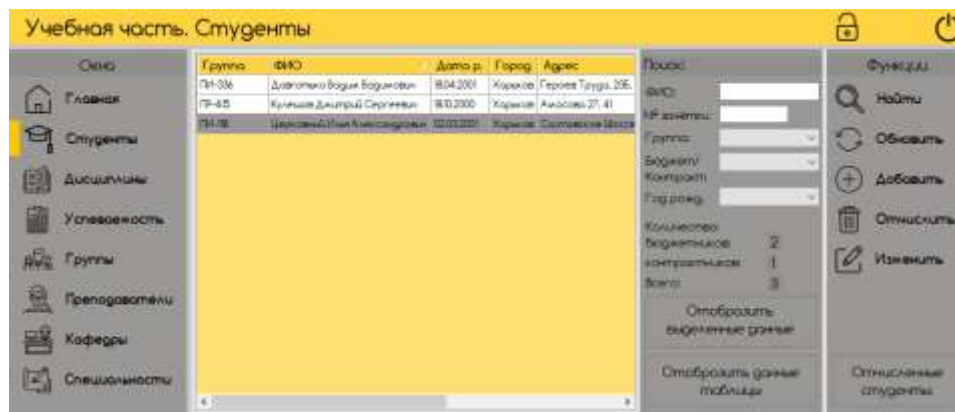


Рисунок 3. 6 - Студенты

Попробуем отчислить только что созданного студента. Для этого нажмём кнопку “Отчислить”. Студент переместится на форму Отчисленных студентов. Чтобы это проверить, нажмём на кнопку “Отчисленные студенты”, при этом перейдем на форму, отраженную на рисунке 3.7

Группа	ФИО	Дата р.	Город	Адрес	Дата пост.	Телефон	№ зачетки	Б\К	Статус
ТН-08	Церковный Илья Александрович	09.03.2000	Харьков	Салтовское Шоссе 260, 34	09.03.2000	0683830505	220617	Контракт	Выпущен

Рисунок 3. 7 - Просмотр отчисленных студентов

Как видим, студента действительно занесло в список отчисленных студентов. Здесь мы можем восстановить каждого студента обратно на учёбу, но уже на контрактной основе и на первый курс. Давайте попробуем. Для этого нажмём на кнопку “Восстановить студента” нам откроется следующая форма (рисунок 3.8).

**Восстановление студента**

Группа:  Шифр:  Курс:  №:  Год:

ФИО:

Город:

Адрес:

Телефон:

№ зачётки:

Бюджет\Контракт:

Дата рождения:

Рисунок 3.8- Форма восстановления студента

Да, действительно, открылась форма (рисунок 3.9), где нельзя поменять курс, тип обучения. Тут находятся такие же ограничения на ввод, какие и при добавлении студента. После восстановления студента, вся его прошлая успеваемость стирается для избегания конфликтов приложения. Есть возможность дать ему как тот же номер зачётки, так и новый. На выбор пользователя.

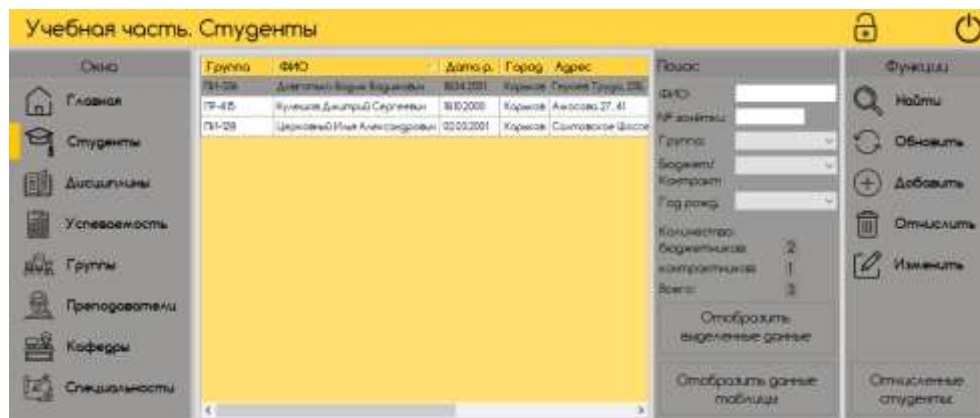


Рисунок 3. 9 - Студенты

Итак, мы восстановили студента, обновили информацию таблицы нажатием кнопки “Обновить”. Здесь был рассмотрен весь функционал панели “Студенты”.

Перейдём на следующую панель, панель “Дисциплины”. Для этого нажмём на соответствующую кнопку “Дисциплины” и перед нами отразится информация о дисциплинах (рисунок 3.10).



Рисунок 3. 0 - Дисциплины

Данная панель позволяет просматривать все дисциплины, преподаваемые в ВУЗе. Справа от таблицы мы имеем довольно похожий функционал как на прошлой панели “Студенты” за исключением того, что здесь уже кнопка “Удалить” и отсутствует кнопка отчисленных студентов.

Также она позволяет осуществлять поиск по заданным параметрам (Название дисциплины, Тип оценивания (Зачёт, Экзамен, Дифференцированный зачёт), Семестр(1-8)).

Ниже можно найти отображение в реальном времени количества дисциплин в сумме и за каждый семестр. Опять же, здесь находятся две кнопки, “Отобразить выделенные данные” и “Отобразить данные таблицы”. Давайте теперь нажмём кнопку “Отобразить выделенные данные” и перед нами откроется следующая форма (рисунок 3.11).



Рисунок 3. 51 - Более обширный и удобный просмотр дисциплин

Открылась форма, где в таблице находятся все те дисциплины, которые мы заранее выделили курсором. В нашем случае – одна дисциплина. Справа так же есть уже знакомый функционал, количество дисциплин, возможность сделать скриншот и вернуться обратно.

Давайте вернёмся и попробуем добавить новую дисциплину. Для этого нажмём на кнопку добавить и появится окно для ввода новой дисциплины (рисунок 3.12).

Рисунок 3.12 - Форма добавления/изменения данных дисциплин

При нажатии кнопки “Изменить”, открывается аналогичная форма, но уже с заполненными полями. Она позволяет добавлять/изменять информацию о дисциплине.

Чтобы добавить/изменить данные дисциплины, требуется заполнить все поля. Если какое-то поле останется незаполненным, пользователю выведет соответствующее сообщение, что следует заполнить все пробелы. Уникальными значениями являются совокупность Названия дисциплины и семестр её преподавания. При нажатии на кнопку “Добавить”, система проверит есть ли такая дисциплина уже в базе данных или нет. Если есть, то выведет соответствующее сообщение пользователю, что следует добавить другую дисциплину, так как эта уже есть.

Поле “Название” может повторяться, ведь одна и та же дисциплина может читаться на нескольких курсах.

Поля “Семестр” и “Оценивание” являются полностью выпадающими списками.

При корректном вводе данных, новая дисциплина добавится в базу данных, либо произойдёт обновление данных, если мы нажимали кнопку “Изменить”:

Теперь попробуем удалить эту дисциплину, ведь мы добавили её по ошибке. Нажмём кнопку Удалить и нам выведет диалоговое сообщение, действительно ли

мы хотим удалить данную дисциплину. Нажимаем “ОК” и видим следующее окно (рисунок 3.13)



Рисунок 3.13 - Дисциплины

Действительно, дисциплина удалилась! Но если бы до этого данная дисциплина имела определённую успеваемость, то удалить её уже нельзя было. Для этого следовало сначала удалить все успеваемости по этому предмету, а только уже потом дисциплину. Но это не имеет смысла.

Перейдём на следующую вкладку, “Успеваемость”. Для этого нажмём на кнопку “Успеваемость” и перед нами окажется следующая панель (рисунок 3.14).

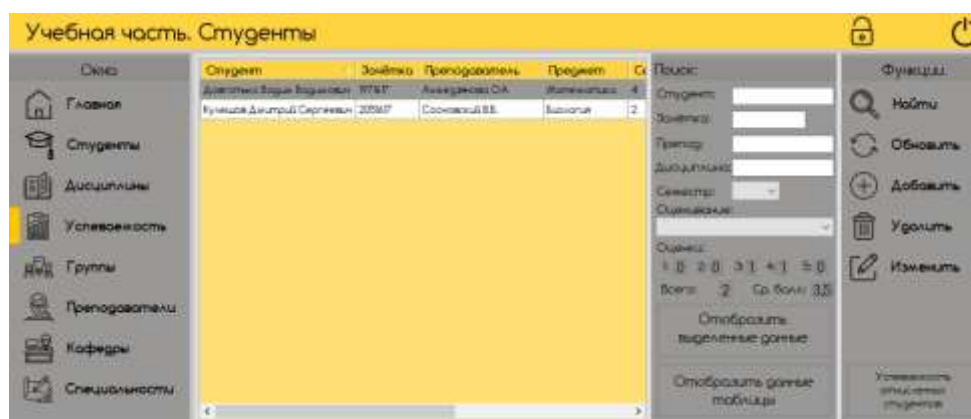


Рисунок 3.14 - Успеваемость

Данная панель позволяет просматривать успеваемость каждого из студентов. Функции абсолютно такие же, как и у вкладки “Дисциплины” разве что добавилась кнопка просмотра Успеваемости отчисленных студентов.

На данной панели возможен поиск по следующим параметрам: ФИО студента, Номер зачётки, ФИО преподавателя, Название дисциплины, Семестр, Оценивание (Зачёт, Экзамен, Дифференцированный зачёт). Так же в реальном времени отображается количество всех оценок, средний балл и количество каждой оценки от одного до пяти.

Попробуем добавить новую успеваемость, нажав кнопку “Добавить” (рисунок 3.15).

The screenshot shows a web form titled "Добавление/Изменение данных" (Addition/Change of data). The form contains the following fields and controls:

- Студент** (Student): A text input field.
- Преподаватель** (Teacher): A text input field.
- Дисциплина** (Discipline): A text input field.
- Дата оценивания** (Evaluation date): A date picker showing "апрель 2019 г." (April 2019). The calendar grid shows days from 1 to 28, with the 29th highlighted. Below the calendar, it says "Сегодня: 29.04.2019" (Today: 29.04.2019).
- Зачётка** (Credit): A text input field.
- Балл** (Score): A dropdown menu.
- Семестр:** (Semester): A dropdown menu.
- Buttons:** "Добавить" (Add) and "Отмена" (Cancel).

Рисунок 3.15 - Форма добавления/изменения данных об успеваемости студента

Данная форма открывается так же при нажатии кнопки “Изменить”, но уже с заполненными полями. Если какое-то поле останется незаполненным при нажатии на кнопку “Добавить”, то пользователю выведется соответствующее сообщение.

Проверкой на то, существует ли уже такая специальность, служит номер зачётки, Название дисциплины и Семестр.

Невозможно добавить Дисциплину Чётного семестра, если на данный момент Месяц от сентября и до декабря. И наоборот.

При корректном вводе, пользователь сможет добавить успеваемость студента. И она отобразится в панели “Успеваемость”.

Перейдём на вкладку “Успеваемость отчисленных студентов”, нажав на кнопку “Успеваемость отчисленных студентов” (рисунок 3.16)

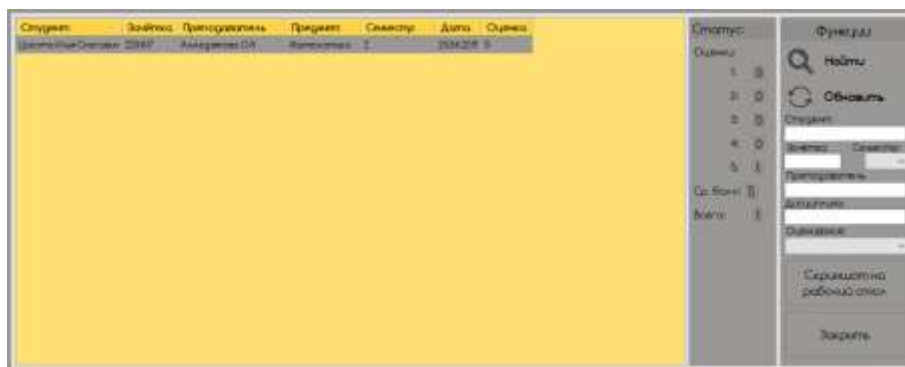


Рисунок 3.16 - Просмотр успеваемости отчисленных студентов

Данная форма хранит всю информацию про успеваемость отчисленных студентов с возможностью сортировки как на обычной панели. Сделано это для того, чтоб при необходимости предоставить человеку через N-ный промежуток времени ведомость с его успеваемостью. Для этого в поиске по ФИО и зачётке вписываем его данные, берём, к примеру, оценивание Экзамен и получаем данные о всех его экзаменах. Нажимаем на кнопку “Скриншот на рабочий стол”, получаем скриншот, который можно использовать для дальнейшей печати.

Вернёмся обратно, перейдём на вкладку “Группы”, нажав на кнопку “Группы” (рисунок 3.17).



Учебная часть. Студенты

Шифр	Курс	№	Год	Кл. рук.	Студентов	Б	К
ЕТ	1	1	8		0	0	0
ЕТ	2	1	7		0	0	0
ЕТ	3	1	6		0	0	0
КР1	1	2	8		0	0	0
КР1	2	2	7		0	0	0
КР1	4	3	5		0	0	0
КР1	4	2	5		0	0	0
КР1	4	1	5		0	0	0
КР1	3	3	4		0	0	0
КР1	3	2	4		0	0	0
КР1	1	1	8		0	0	0
КР1	1	2	8		0	0	0
КР1	2	1	7		0	0	0
КР1	2	3	7		0	0	0
ОМ	1	1	8		0	0	0

Поиск:  
Шифр:   
Курс:   
Кл. рук.:   
Количество групп: 47

Отобразить выделенные данные  
Отобразить данные таблицы

Функции:  
Найти  
Обновить  
Добавить  
Удалить  
Изменить  
На сл. курс  
Выпущенные группы

Рисунок 3.17- Группы

Здесь отображены все группы, занесённые в базу данных.

На панели “Группы” возможно осуществить поиск по Шифру, Курсу и классному руководителю.

Отображается в реальном времени суммарное количество групп.

В самой таблице указывается количество студентов в каждой группе, количество бюджетников и, соответственно, контрактников.

Из функционала добавились следующие кнопки: Перевести на следующий курс и просмотр выпущенных групп. Кнопка “На сл. курс” активна только летом и активировать её можно только один раз за год. При этом все группы переходят на следующий курс, а четвёртый курс становится выпущенной группой, что можно проверить, нажав на кнопку “Выпущенные группы” (рисунок 3.18).

Шифр	Курс	№	Год	Ст	Б	К
П1	1	1	8	1	0	1

Скриншот на рабочий стол

Кол-во групп: 1

Заккрыть

Рисунок 3.18 - Просмотр выпущенных групп

Здесь у нас находится группа ПИ-118 просто для проверки, что здесь находятся выпущенные группы. После этого она, конечно же, отсюда уберётся. А выпущенные группы добавятся сюда летом, достаточно будет всего лишь нажать на кнопку перевода на след. Курс.

Вернёмся назад и попробуем добавить новую группу. Нажмём на кнопку “Добавить”, отразится форма представленная на рисунке 3.19

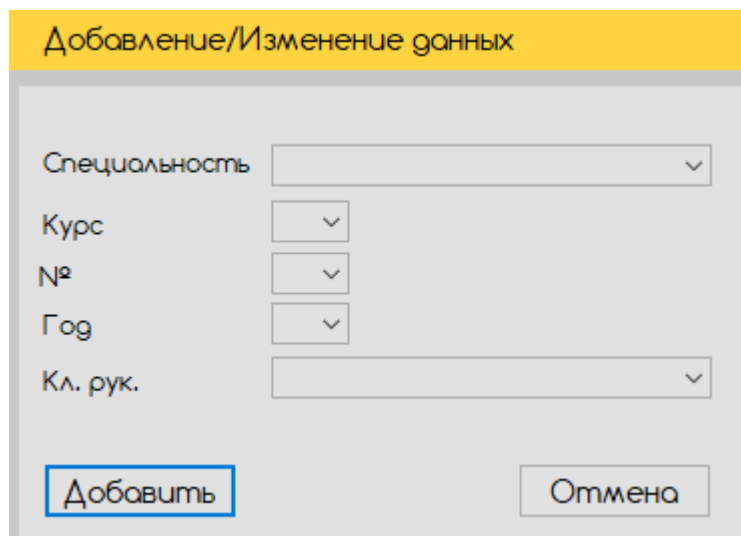


Рисунок 3.19- Форма добавления/изменения данных группы

Перед нами откроется форма добавления/изменения данных группы, которая состоит полностью из выпадающих списков. Это сделано для облегчения работы сотрудника. Всё, что нам требуется: выбрать из каждого списка, то что нужно и нажать кнопку добавить. Система не даст возможности добавить такую же группу, которая выпустилась либо же которая ещё учится.

Требуется, чтобы все списки были с выбранными значениями. В противном случае выведется соответствующее диалоговое окно.

При корректном вводе, новая группа добавится в базу и у нас обновится информация об этой таблице.

Перейдём на следующую вкладку “Преподаватели” (рисунок 3.20), нажав на кнопку “Преподаватели”.

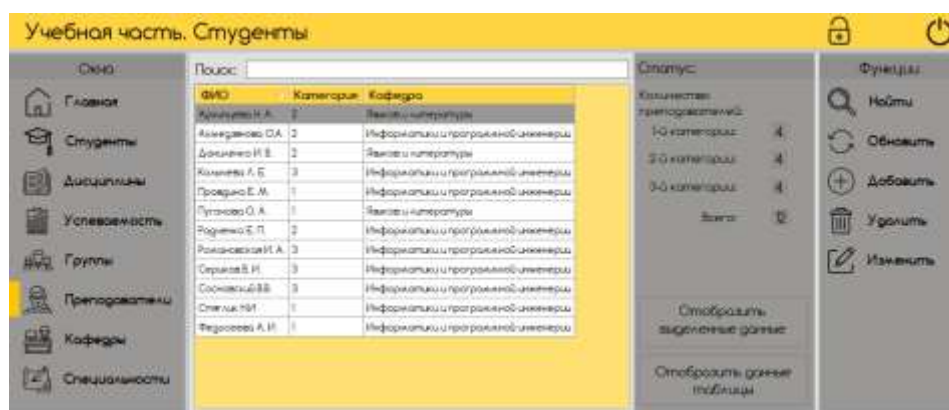


Рисунок 3.20- Преподаватели

На данной панели отображается количество всех преподавателей и отдельно каждой категории. Возможен поиск преподавателей по ФИО, просмотр в отдельном окне в более удобном формате данных.

Добавление данных происходит при помощи формы Добавления.

При этом из трёх элементов, требующих к заполнению, два являются выпадающими списками. Проверка на уникальность преподавателя идёт только по ФИО. Всё потому, что нам от преподавателя нужна всего лишь фамилия и инициалы для того, чтоб занести его в заведующие специальностей, кафедр, быть классным руководителем, принимать зачёты.

При корректном заполнении всех элементов, новый преподаватель добавится в базу данных.

Точно так же и с изменением данных.

Перейдём на вкладку “Кафедры” (рисунок 3.21).

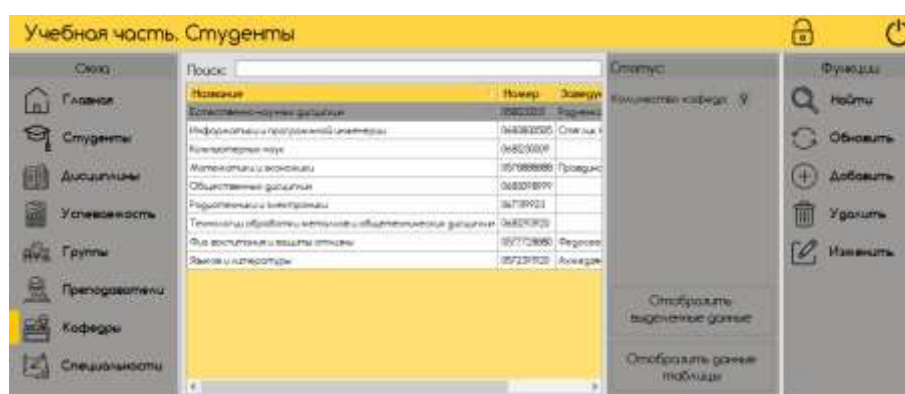
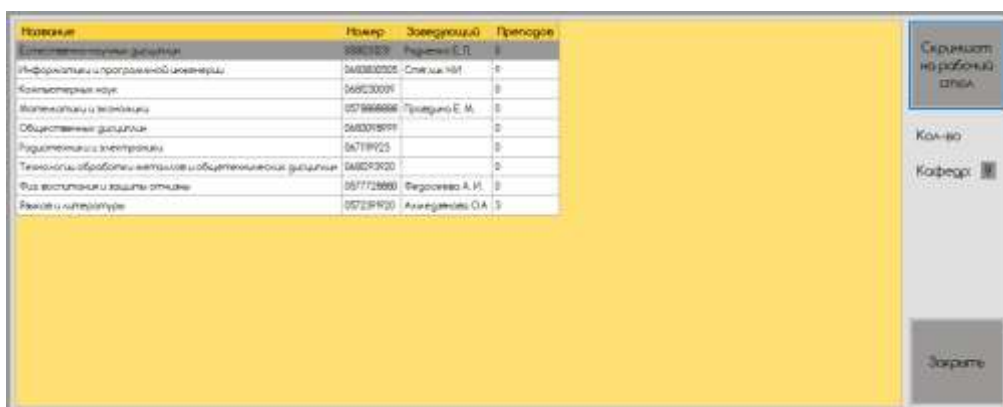


Рисунок 3.21 - Кафедры

Данная панель позволяет просматривать количество всех кафедр, осуществлять поиск по названию кафедры. Так же таблица показывает количество преподавателей в каждой кафедре. Для этого лучше стоит воспользоваться кнопкой “Отобразить данные таблицы” (рисунок 3.22).



Название	Номер	Заведующий	Преподав
Кафедра информационных технологий	06030303	Руденко Е.П.	4
Информационные технологии	06030303	Степанов И.И.	1
Компьютерная наука	06030303		0
Математика и механика	05780606	Тришину Е. М.	0
Общественные дисциплины	06030303		0
Радиоэлектроника	06710925		0
Технологии обработки металлов и общетехнические дисциплины	06030303		0
Физ. воспитание и защита от огня	05772800	Березовая А. И.	0
Языки и литература	05720920	Александров О.А.	3

Скриншот на рабочий стол

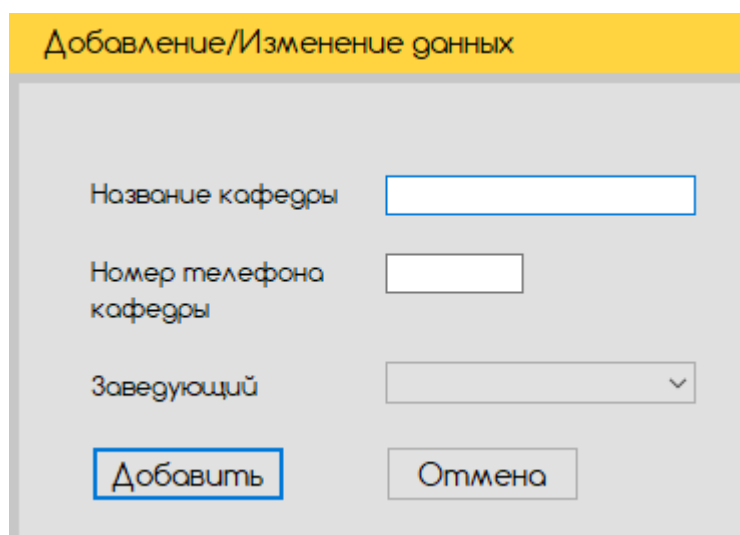
Кол-во Кафедр: 1

Закрыть

Рисунок 3.22- Форма просмотра в более удобном и обширном формате кафедр

Как видим, последняя колонка как раз и указывает на количество преподавателей.

Вернёмся и попробуем добавить новую кафедру (рисунок 3.23)



Добавление/Изменение данных

Название кафедры

Номер телефона кафедры

Заведующий

Рисунок 3.23 - Форма добавления кафедр

Как мы видим, здесь тоже три элемента, требующих к вводу. Но из них уже только один – выпадающий список. В списке заведующий следует выбрать одну из предложенных фамилий.

Перейдём на вкладку “Специальности” (рисунок 3.24).

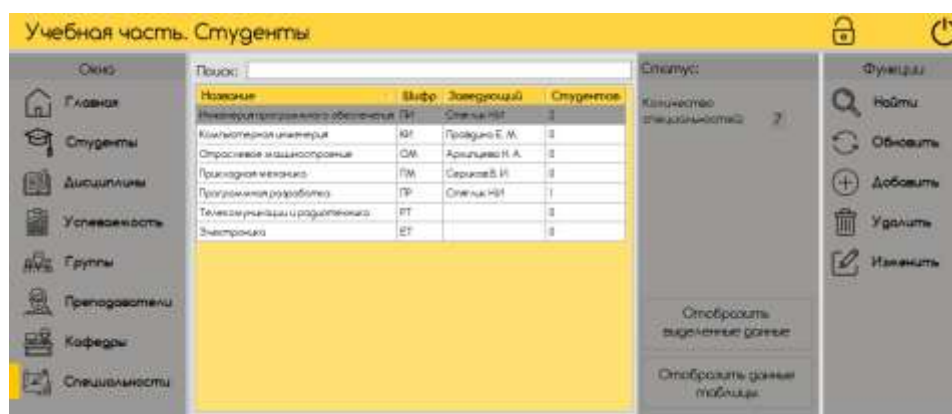


Рисунок 3.24 - Специальности

Здесь мы можем просмотреть информацию про каждую из специальностей, количество специальностей, количество студентов, учащихся на каждой из них.

Откроем форму добавления (рисунок 3.25).

Рисунок 3.25 - Форма добавления/изменения данных о специальности

Как видим, здесь аналогичная ситуация с формой добавления Кафедр.

## ВЫВОДЫ

В результате выполнения курсового проекта, было получено много опыта и знаний по использованию программных средств, таких как: Visual Studio, MySQL WorkBench. Из неявного и мутного представления предметной области сформировалась чёткая база, на основе которой, в дальнейшем, можно будет создавать более качественные программные продукты для коммерческой выгоды и собственного развития в качестве “изобретателя”.

В ходе изобретения программного продукта, он потерпел несколько раз кардинальных изменений:

- Вначале этому способствовало неправильное видение готовой задачи.
- После – некорректное и неудобное использование всех объектов, которое привело к полнейшей путанице.
- И наконец: истребление недочётов, своеобразных “дыр” в проекте.

К концу разработки ПО, возник ряд вопросов, которые требуются для большей функциональности приложения. Так же был рассмотрен лист с требуемыми запросами для минимальной оценки и стало понятно, что проект не готов до конца. Все тонкости были учтены и предусмотрены для безопасного использования программного продукта. Срок работы продукта оценён приблизительно в 5-10 лет без каких-либо возможных конфликтов.

## Список литературы

1. <https://metanit.com/sharp/tutorial/>
2. <https://stackoverflow.com/>
3. <http://www.cyberforum.ru>
4. <http://qaru.site>