

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение  
высшего образования

**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г.ЧЕРНЫШЕВСКОГО»**

Кафедра математического и компьютерного моделирования

**Цифровой ВУЗ. Студенты**

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студента 4 курса 411 группы

направление 01.03.02 — Прикладная информатика и информатика

механико-математического факультета

Наземнов Глеб Андреевич

Зав. кафедрой  
зав. каф., д.ф.-м.н., доцент

Ю.А. Блинков

Зав. кафедрой  
зав. каф., д.ф.-м.н., доцент

Ю.А. Блинков

Саратов 2020

**Введение.** На сегодняшний день существует большое количество технологий, облегчающих разработку и объединение программных компонентов. Одними из таких технологий являются Python и MongoDB, используемые для разработки приложений, служащими в качестве информационных систем.

Целью данной выпускной квалификационной работы бакалавра является знакомство с востребованными технологиями, используемыми при разработке приложений, такими как Python, PySide(PyQT) и MongoDB, а также разработка системы информационного контроля для кафедры математического и компьютерного моделирования.

Для достижения поставленной цели, необходимо решить несколько задач:

1. Рассмотреть программу, которую требуется доработать
2. Подробно изучить способы построения приложений с использованием СУБД MongoDB.
3. Реализовать функцию переводов различными способами(перевод одного студента,перевод всех студентов кафедры)

**Основное содержание** работы содержит 3 раздела:

1. Введение
- 2.Выбор и обоснование используемого ПО
- 3.Проектирование и создание интерфейса пользователя к базе "Студент"

Во **введении** формулируется цель работы и решаемые задачи.

В **первом разделе** рассматриваются способы написания информационных систем

Во **втором разделе** описываются средства написания кода используемые уже в готовой кодовой базе и сравнение их с другими, содержит описание используемого ПО(СУБД)

В **третьем разделе** Описывается проектирование кодовой базы.

### **Описание предметной области**

**Информационные системы** Информационные системы – это системы, предназначенные для хранения, поиска и обработки информации, то есть для удовлетворения конкретных информационных потребностей в рамках определенной предметной области.

Существуют различные классификации систем, каждая из которых предназначена для удовлетворения информационных потребностей в рамках конкретной предметной области.

### **Постановка задачи по доработке**

В рамках данной бакалаврской работы, требовалось доработать нативное настольное приложение использующееся на кафедре математического и компьютерного моделирования. Приложение предназначалось для учета учебных работ студентов, и для учета сотрудников. Необходимо доработать часть приложения связанную со студентами. Нужда в доработке возникла, из-за невозможности перевода студентов на следующий курс и учета этих студентов.

Требовалось:

- доработать приложение
- изучить MongoDB
- перевести на новую версию библиотеки
- добавить возможность перевода студентов на следующий курс

### **Технологии, используемые в работе**

**Python** Python - высокоуровневый язык программирования общего назначения, ориентированный на повышение производительности разработчика и читаемости кода. Синтаксис ядра Python минималистичен. В то же время стандартная библиотека включает большой объём полезных функций.

В данной работе используется библиотека PySide основанная в свою очередь на технологии Qt. Qt позволяет запускать написанное с его помощью программное обеспечение в большинстве современных операционных систем путём простой компиляции программы для каждой системы без изменения исходного кода. В свою очередь PySide является привязкой к Qt. Библиотека использует основные функции и API. Является полностью объектно-ориентированным, расширяемым и поддерживающим технику компонентного программирования.

Одним из преимуществ проекта Qt является наличие качественной документации. Статьи документации снабжены большим количеством примеров. Исходный код самой библиотеки хорошо форматирован, подробно комментирован и легко читается, что также упрощает изучение Qt.

**MongoDB** MongoDB - документоориентированная система управления базами данных (СУБД) с открытым исходным кодом, не требующая описания схемы таблиц. Классифицирована как NoSQL, использует JSON-подобные документы и схему базы данных. Написана на языке C++.

В данной работе MongoDB используется для хранения данных о студентах и сотрудниках, и для хранения различных работ студентов (курсовые, рефераты, практики, дипломы)

## Разработка информационной системы

**Постановка задачи и анализ требований** Помимо изучения выше указанных технологий и анализа уже существующего кода, выпускная работа подразумевает доработку связанную с хранением и учетом студентов кафедры. Важнейшими функциями, которые должны быть реализованы в системе, являются:

- Перевод всех студентов на следующий курс;
- Перевод студентов на следующий курс с учетом, отчислений и академических отпусков.;

Помимо реализации поставленного функционала, требовалось проанализировать уже готовый код, научиться писать нативные десктопные приложения. Разобраться в правилах и принципах взаимодействия приложений на Python с документоориентированными СУБД в частности MongoDB.

**Структура приложения** Данное приложение написано на Python с использованием библиотеки PySide. Знакомство с данным приложением необходимо начинать с основ построения интерфейса с помощью данной технологии. Для начала нужно было изучить все основы компоновки и написания кода.

```
1
2
3 import sys
4 from PySide2.QtWidgets import (QLineEdit, QPushButton, QApplication,
5 QVBoxLayout, QDialog)
6
7
8 class Form(QDialog):
9     def __init__(self, parent=None):
10         super(Form, self).__init__(parent)
11         # Create widgets
12         self.edit = QLineEdit("Write my name here")
```

```

13         self.button = QPushButton("Show Greetings")
14         # Create layout and add widgets
15         layout = QVBoxLayout()
16         layout.addWidget(self.edit)
17         layout.addWidget(self.button)
18         # Set dialog layout
19         self.setLayout(layout)
20         # Add button signal to greetings slot
21         self.button.clicked.connect(self.greetings)
22
23         # Greets the user
24         def greetings(self):
25             print("Hello %s" % self.edit.text())
26
27
28 if __name__ == '__main__':
29     # Create the Qt Application
30     app = QApplication(sys.argv)
31     # Create and show the form
32     form = Form()
33     form.show()
34     # Run the main Qt loop
35     sys.exit(app.exec_())

```

Выше приведена выкладка с простейшей программой на библиотеке PySide выводящая окошко, с полем ввода имени, при нажатии на кнопку в консоль выводится значение введенное в поле. Интерфейс представлен в соответствии с рисунком 1.

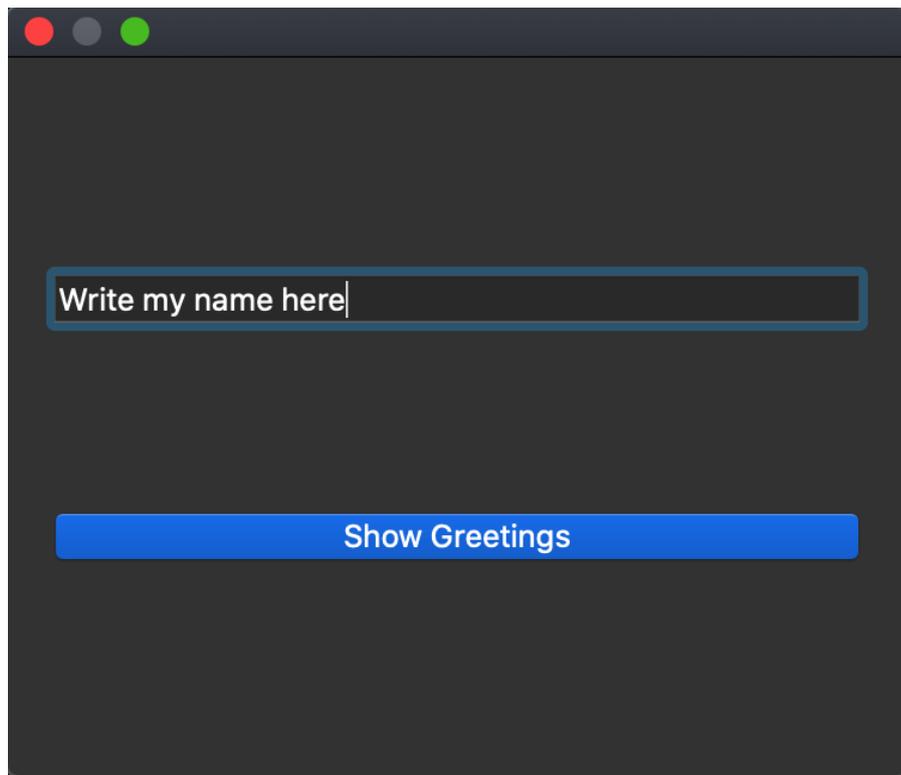


Рисунок 1 — Пример простого приложения

Для подключения, к СУБД требуется установить и подключить Mongo Driver( "монго драйвер"). Mongo Driver - это набор инструкций, написанный под конкретный язык программирования, для работы с данной СУБД. Драйвер называется pymongo, от сочетания python и mongo.

```
1 | import pymongo
```

Вставляем код выше в начало документа. После подключения пакета требуется определить коллекции и базы данных.

```
1 | #подключениек СУБД
2 | client = pymongo.MongoClient("mongodb://localhost:27017")
3 |
4 | #проверка то что клиент валиден и доступен
5 | print(client.isValid)
6 |
7 | db = client["МиКМ2"]
8 | #db.authenticate("", "") если на сервере установлен пароль, требуется войти
9 |
10 | #подключение файловой системы
11 | fs = gridfs.GridFS(db)
12 |
```

```

13 | #получение коллекции "Студенты"
14 | students = db["Студенты"]

```

MongoDB GridFS - правила хранения больших по объему файлов в базе данных MongoDB. GridFS позволяет хранить файлы больше 16Мб (предельный размер документа в MongoDB) за счет использования коллекций. GridFS разбивает большие файлы на небольшие части. Эти части сохраняются в одну коллекцию (fs.chunks), а метаданные о файле в другую коллекцию (fs.files). Когда происходит запрос к файлу, GridFS делает запрос в коллекцию с частями файла и затем возвращает файл целиком.

В выкладке кода выше, можно заметить что используется некая строка `gridfs.GridFS(db)`. Эта строка обеспечивает хранение больших по объему файлов, в случае данной бакалаврской работы - это работы студентов(курсовые, дипломные, рефераты).

```

1 | if __name__ == '__main__':
2 |     import sys
3 |
4 |     app = QtWidgets.QApplication(sys.argv)
5 |     QtCore.QLocale.setDefault(QtCore.QLocale("ru_RU"))
6 |
7 |     translator = QtCore.QTranslator()
8 |     translator.load('/usr/share/qt5/translations/qt_ru.qm')
9 |     app.installTranslator(translator)
10 |
11 |     font = QtWidgets.QApplication.font()
12 |     font.setPointSize(font.pointSize() - 1)
13 |     QtWidgets.QApplication.setFont(font)
14 |
15 |     global wizard
16 |     wizard = Wizard()
17 |     wizard.show()
18 |     sys.exit(app.exec_())
19 |

```

Далее требуется сделать имплементацию класса `QWidget`.

```

1 | class Wizard(QtWidgets.QWizard):
2 |     def __init__(self, parent=None):
3 |         super(Wizard, self).__init__(parent)
4 |
5 |         self.setWindowTitle('Студенты')

```

```

6         self.student = None
7         self.year = None
8
9         self.addPage(ChooseStudent(self))
10        self.addPage(StudentYear(self))
11
12        self.currentIdChanged.connect(self.update)
13
14        self.resize(QtCore.QSize(900, 600))
15
16    def update(self, _id):
17        if _id >= 0:
18            self.page(_id).update()
19

```

Виджет `QWidget` это базовый класс для всех объектов интерфейса пользователя в `PyQt5`. Он предоставляет конструктор по умолчанию для `QWidget`. Конструктор по умолчанию не имеет родителя. Виджет без родителей называется окном. Класс `Wizard` устанавливает нужные значения. Например значения студентов и годов обучения в `None`, что бы объявить поле класса. Далее в функции `main` вызывается метод класса - `show()`.

Описании Студентов в базе данных, определено на выкладке ниже. Так как в `MongoDB` используется формат `BSON`(аналогичный `JSON`) вся структура документов будет приведена в формате `JSON`.

```

1  {
2      "_id" : ObjectId("5b9b2c496b90e718a1b803db"),
3      "email" : "nazemnov.g.a@gmail.com",
4      "ФИО" : "Наземнов Глеб Андреевич",
5      "год" : {
6          "2018" : {
7              "группа" : "311",
8              "руководитель" : "Блинков Ю.А.",
9              "тема" : "",
10             "ФГОС" : ObjectId("582f0682ca1acb217ceb5c82")
11         },
12         "2019" : {
13             "группа" : "411",
14             "руководитель" : ""
15         }
16     }
17 },
18 {

```

```

19     "_id" : ObjectId("5b9b2c496b90e718a1b803db"),
20     "email" : "nazemnov.g.a@gmail.com",
21     "ФИО" : "Наземнов Глеб Андреевич",
22     "год" : {
23         "2018" : {
24             "группа" : "311",
25             "руководитель" : "Блинков Ю.А.",
26             "тема" : "",
27             "ФГОС" : ObjectId("582f0682ca1acb217ceb5c82")
28         },
29         "2019" : {
30             "группа" : "411",
31             "руководитель" : ""
32     }
33 }
34 }

```

В данной выкладке находится вся нужная информация. Где "id" это внутренний номер объекта в базе данных.

На стартовой странице находится окошко со списком студентов. Выше на панели управления находятся функциональные кнопки, поиска, копирования запроса и перевода студентов на следующий курс. Интерфейс приложения соответствует рисунку 2.

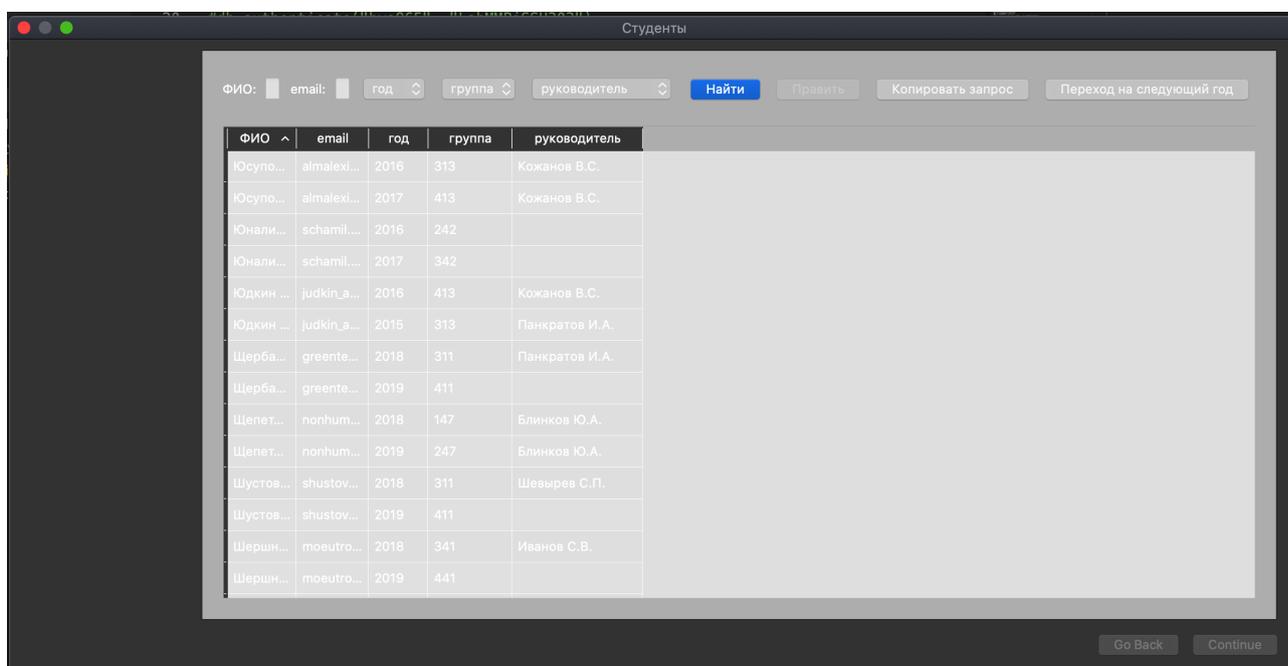


Рисунок 2 — Стартовая страница

Если указать нужные условия поиска и воспользоваться кнопкой найти, то список студентов изменится в соответствии с фильтрами:

Функция считывает фильтры установленные в приложении, и вставляет их в специальную форму заготовленную запроса заранее, запрос отправляется в СУБД.

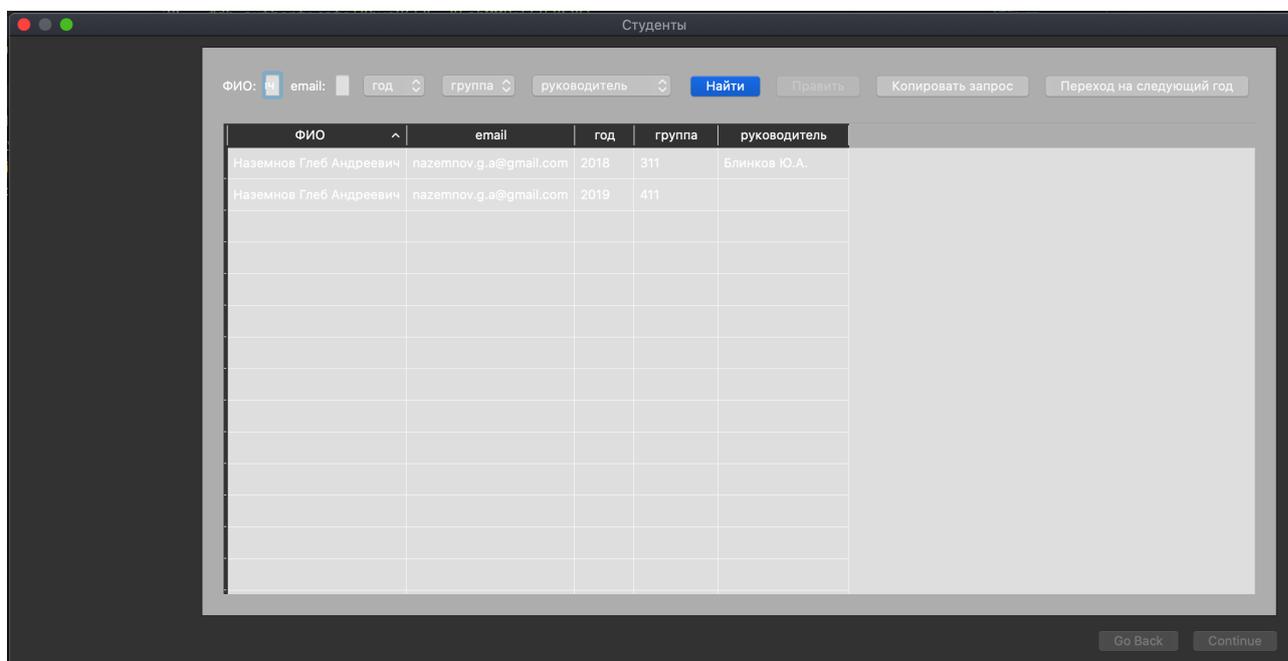


Рисунок 3 — Пример поиска

**Перевод студентов на следующий год.** Решение поставленной задачи. Для решения поставленной задачи, требовалась проанализировать написанный до этого код, и правильно интегрировать в него новые функции.

Перевод реализован с помощью нескольких функций.

Функция перевод одного студента, при вызове этой функции определяется последний год обучения студента, далее этот год сверяется со списком годов, которые позволяют перейти на следующий курс, если вхождение в список есть студент переводится на следующий курс и создается пустая работа:

```
1 def transition_student_to_next_year(id):  
2     years = get_work_of_students(id)  
3     max_y = 0  
4     for y in years.keys():  
5         if int(y) > max_y:  
6             max_y = int(y)  
7     last_year = years.get(str(max_y))
```

```

8     group = last_year.get('группа')
9     if int(group) in can_transition_group:
10         new_group = int(group)+100
11         new_year = {"группа": str(new_group), "руководитель": ""}
12         years[str(max_y+1)] = new_year
13         students.find_one_and_update({"_id": id}, {"$set": {"год": years}})
14

```

Так же реализована функция перевода всех студентов сразу, на следующий год, она использует функцию перевода одного студента в качестве основы, и просто переводит всех студентов сразу:

```

1 def all_to_next_year():
2     for student in students.find({}):
3         transition_student_to_next_year(student['_id'])

```

Теперь данные функции возможно использовать в пользовательском интерфейсе, путем вызова этих функций и привязки их к специальным кнопкам. Так же после вызова этих функций с помощью кнопок, пользователь должен увидеть предупреждающее окно, и подтвердить свое действие. Кнопка перевода одного конкретного студента, находящаяся в меню редактирования работы:

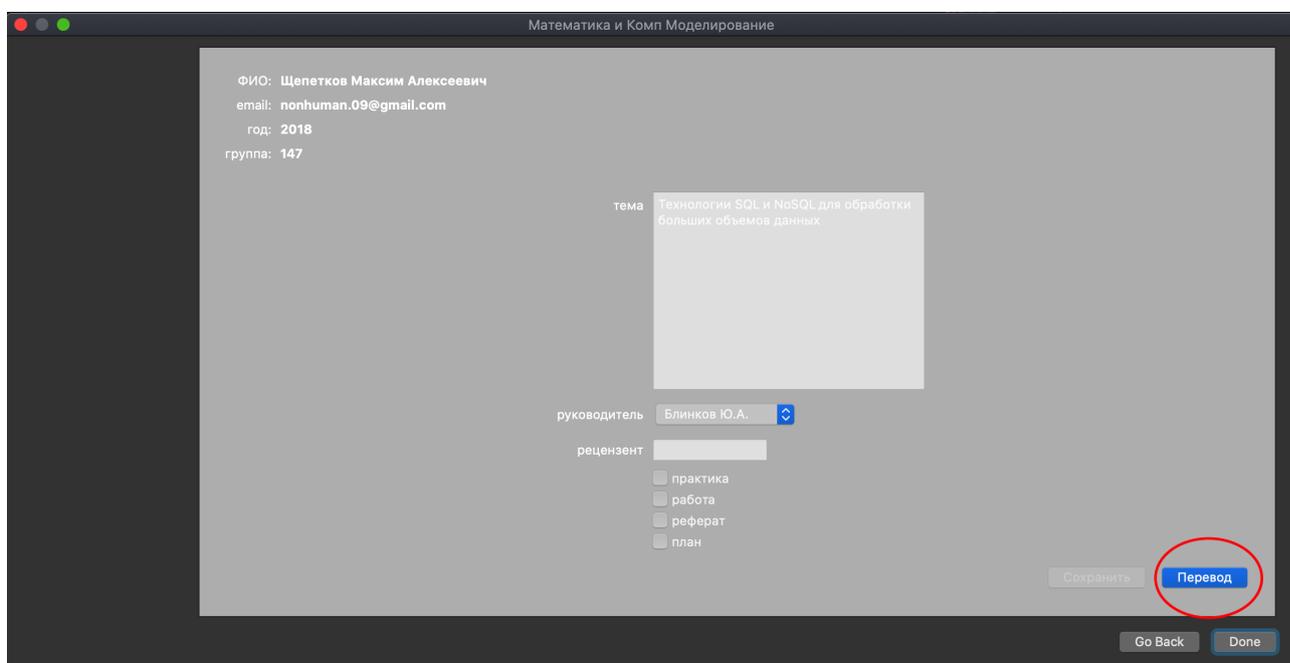


Рисунок 4 — Пример кнопки

Модальное окно с кнопками подтверждения:

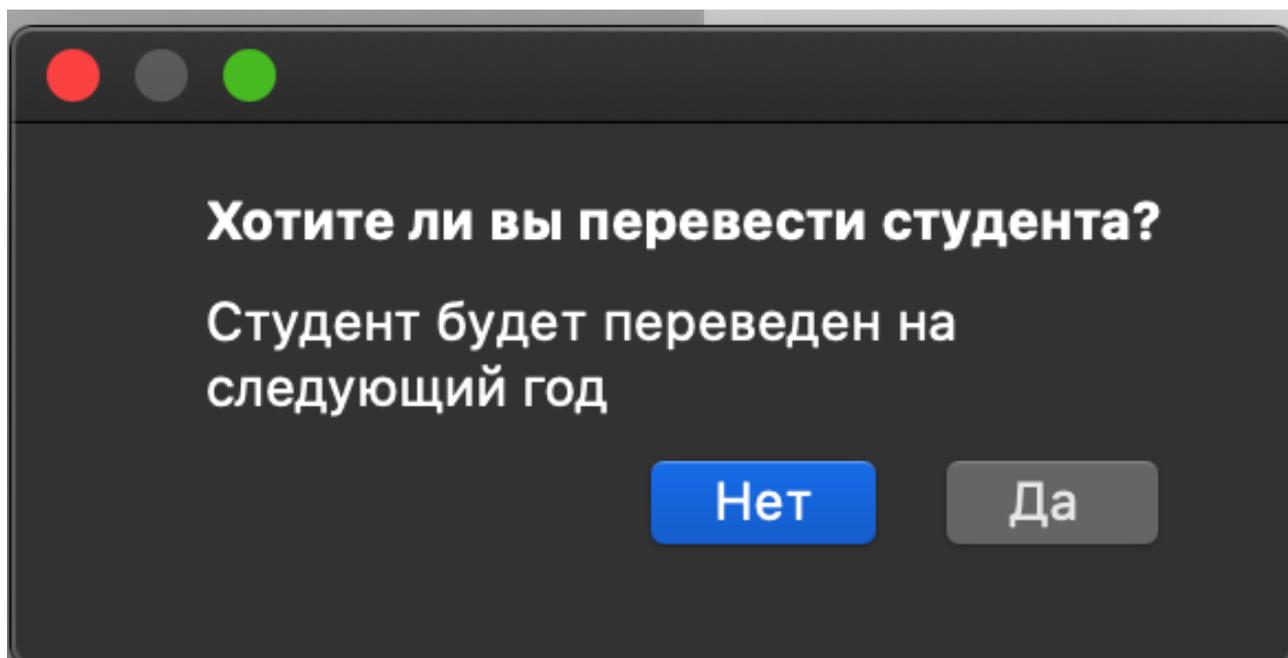


Рисунок 5 — Окно предупреждения

Так же была реализована кнопка перевода всех студентов на следующий курс:

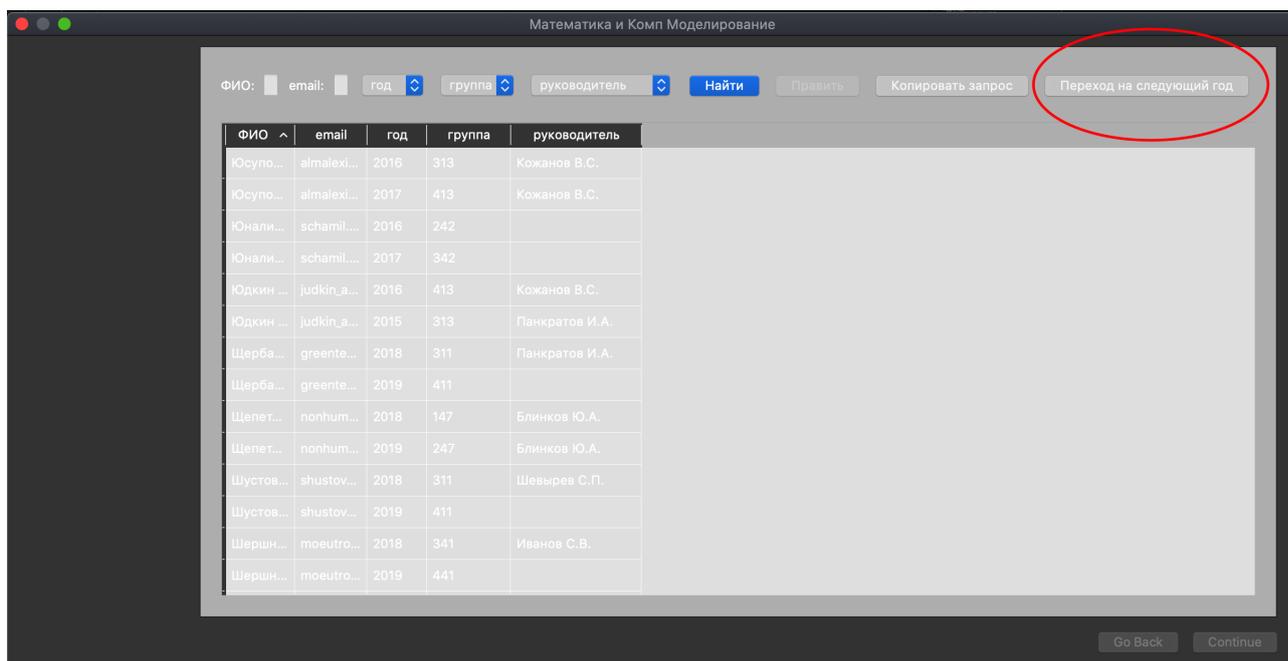


Рисунок 6 — Пример кнопки перевода всех студентов

Модальное окно с кнопками подтверждения:

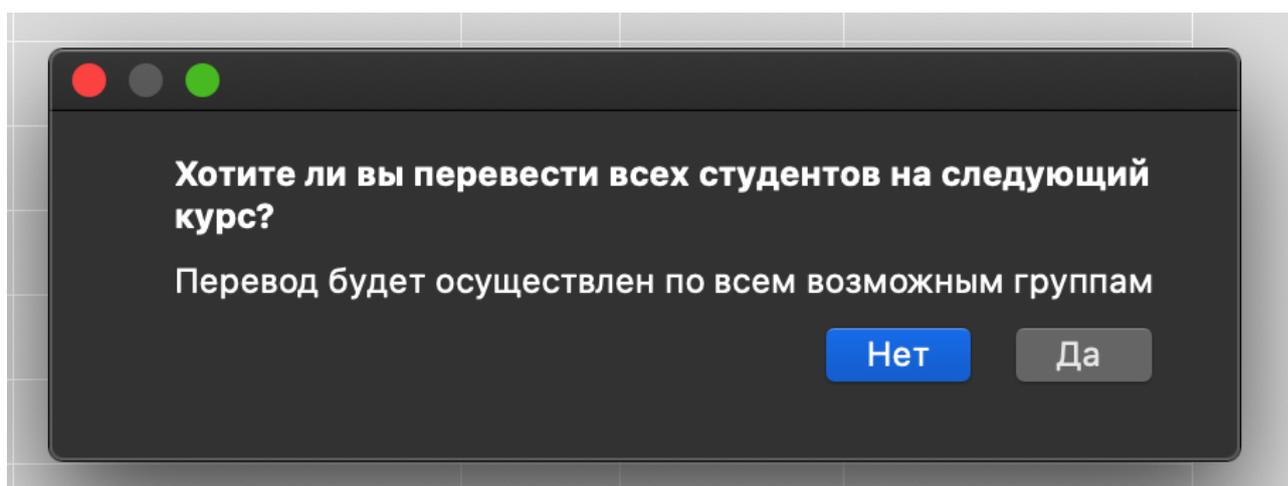


Рисунок 7 — Окно предупреждения

**Заключение.** На сегодняшний день, для осуществления деятельности любого предприятия, будь то ВУЗ или коммерческая организация требуется правильный подход к учету и ведению различных данных.

В результате выполнения данной выпускной квалификационной работы было произведено ознакомление с современными технологиями, используемыми при разработке информационных систем. Также, в соответствии с требованиями поставленными при написании дипломной работы, было разработано решение управления данными связанными с учетом студентов и их учебными работами(в частности способы перевода студентов на следующий курс).