ГБОУ «Школа № 444»

Проектная работа по информатике:

«Создание приложения для создания и обработки заявок на выход из школы»

Работу выполнили:        Карак Юлия Алексеевна

ученица 10 «И» класса

Мкртчян Алина Никитична

Ученица 10 «В» класса

Работу приняла:    Мякова Елена Георгиевна

учитель информатики

Москва

2023 год

**Тема проектной работы**

Создание приложения для создания и обработки заявок на выход из школы.

**Введение**

В настоящее время цифровизация процессов, встречающихся нам в жизни, является всё более важным аспектом глобального прогресса. С помощью всевозможных приложений и устройств мы можем всё более быстро и качественно осуществлять нужные операции.

Получение заявления на выход из школы – достаточно долгий, для столь несложной задачи, процесс, требующий, к тому же, согласования с классным руководителем. Поэтому, мы подумали, что создание приложения для подобного взаимодействия между учениками и классным руководителем решит данную проблему.

Таким образом, с помощью нашего приложения, появилась возможность снять дополнительную нагрузку и убрать избыточную коммуникацию из повседневных задач классного руководителя.

**Цель проекта**

Разработать приложение для создания заявок на выход из школы, их обработки и создания заявлений.

**Задачи проекта**

1. Изучить проблему долгого процесса выдачи заявлений на выход из школы
2. Изучить язык программирования Python и среду программирования PyCharm для создания приложения
3. Создать приложение

**Техническое задание**

Цель работы - создание приложение для создания заявок на выход из школы и последующего создания заявлений.

Приложение должно состоять из 2 основных окон со стороны ученика:

1. Основное меню с отправленными заявками
2. Создание заявки

И из 2 основных окон со стороны учителя:

1. Основное меню с полученными заявками
2. Просмотр заявки

**Практическая значимость**

Данное приложение будет полезным для учеников и их классных руководителей.

**Выбор среды разработки**

Для разработки приложения был выбран язык программирования Python. По состоянию на сентябрь 2022 года Python занимает первое место в рейтинге TIOBE популярности языков программирования.

**Стек технологий**

1. Python
2. PyQT5
3. PyCharm
4. SQLite
5. Git

**Структура программы**

При первом запуске приложения пользователю необходимо будет выбрать свою роль как пользователя: Учитель или Ученик

При выборе роли Учителя пользователю нужно будет ввести данные:

1. Фамилию
2. Имя
3. Отчество
4. Эл. Почту
5. Логин
6. Пароль

Далее пользователю нужно будет загрузить файл в формате .csv с данными учеников в формате Фамилия Имя. На каждого ученика будет выдан логин и пароль, которые учитель должен им выдать. Далее в программе к каждому учителю будут привязаны все его ученики и каждый ученик будет привязан к своему учителю.

При выборе роли Ученика пользователю нужно только авторизоваться.

**Структура базы данных**

1. Таблица поданных заявок – общая таблица для заявок, где содержится информация об учителе, ученике, заявке и её статусе.

Логин учителя

Логин ученика

Причина ухода

Время ухода

Дата ухода

Причина отказа (заполняется при отказе учителя в выдаче заявления на выход из школы)

Статус заявки

1. Таблица учителей – таблица, где содержится информация об учителях.

Фамилия

Имя

Отчество

Логин учителя

Пароль учителя

Адрес электронной почты

Аватар

1. Таблица учеников – таблица, где содержится информация об учениках.

Логин ученика

Пароль ученика

Имя

Фамилия

Адрес электронной почты

Логин учителя

Аватар

**Реализация проекта**

Для создания окон приложения использовалось приложение QT Designer. (Приложение 1)

Для отображения окон использовались функции библиотеки Python PyQT5.

Пример программы 1:

uic.loadUi('ui\_files/teacher\_entrance\_new.ui', self)

self.show()

Приложение использует локальные базы данных. Для их создания используется приложение SQLite и библиотеку Python sqlite3. Для создания базы данных и её таблиц использовался класс RoleWindow, функция \_\_init\_\_.

Пример программы 2:

self.users = sqlite3.connect("db/users.sqlite")  
self.cur1 = self.users.cursor()  
self.cur1.execute("""CREATE TABLE IF NOT EXISTS users(  
 pupillogin TEXT,  
 pupilpassword TEXT,  
 pupilname TEXT,  
 pupilsurname TEXT,  
 pupilemail TEXT,  
 teacherlogin TEXT,  
 avatarfile TEXT);  
""")  
self.users.commit()

Изменения базы данных происходили во многих классах и не использовали конкретных функций.

Пример программы 3:

self.users = sqlite3.connect('db/users.sqlite')  
self.cur1 = self.users.cursor()  
self.show()

cur1.execute(f"INSERT INTO teachers (teachersurname, teachername, teachername2, teacherlogin, teacherpassword, email, avatarfile) VALUES(?, ?, ?, ?, ?, ?, ?)", inp)  
users.commit()

Для проверки полей ввода логина и пароля при регистрации используется класс TeacherCheckin и его функция password. При введении некорректного логина или пароля (не удовлетворяющих хотя бы одно из условий:

1. Состоит из 8 – 20 символов
2. Содержит буквы латиницы любого регистра или символы «!#$%^}{[]()":\|.»
3. Содержит хотя бы одну цифру)

Выводится сообщение «Некорректный логин или пароль!»

Пример программы 4:

def password(self, x, y):  
 c1 = 0  
 if x != '':  
 if 8 <= len(x) <= 20:  
 if '123456' not in x:  
 if 'qwerty' not in x:  
 if 'password' not in x:  
 for i in x:  
 if i.lower() in 'qwertyuiopasdfghjklzxcvbnm1234567890!#$%^}{[]()":\|.':  
 c1 += 1  
  
 c2 = 0  
 if y != '':  
 if 8 <= len(y) <= 20:  
 if '123456' not in y:  
 if 'qwerty' not in y:  
 if 'password' not in y:  
 for i in y:  
 if i.lower() in 'qwertyuiopasdfghjklzxcvbnm1234567890!#$%^}{[]()":\|.':  
 c2 += 1  
  
 if c1 != len(x) or c2 != len(y):  
 msg = QMessageBox(QMessageBox.Information, '', 'Некорректный логин или пароль!', parent=self)  
 msg.show()  
 else:  
 return True

Для проверки полей ввода при авторизации в приложение частично используются классы RegWin и PupilEntrance и их функции clickBtn1.

Пример программы 5:

def clickBtn1(self):  
 users = sqlite3.connect('db/users.sqlite')  
 cur1 = users.cursor()  
 login = cur1.execute(f"SELECT \* from teachers where teacherlogin='{self.lineEdit\_11.text()}'").fetchone()  
 # Проверяем, есть ли такой учитель в базе данных  
 if login is None:  
 msg = QMessageBox(QMessageBox.Information, '', 'Логин не найден. \nПопробуйте ещё раз или \nсоздайте новый аккаунт', parent=self)  
 msg.show()  
 elif login[4] != self.lineEdit\_12.text().strip():  
 msg = QMessageBox(QMessageBox.Information, '',  
 'Неверный пароль. \nПопробуйте ещё раз или \nсоздайте новый аккаунт', parent=self)  
 msg.show()  
 elif login[4] == self.lineEdit\_12.text().strip():  
 self.teacher = str(login[3])  
 self.openTeacherEntrance()

def clickBtn1(self):  
 users = sqlite3.connect("db/users.sqlite")  
 cur1 = users.cursor()  
 login = cur1.execute(f"SELECT \* from users where pupillogin='{self.lineEdit\_4.text().strip()}'").fetchone()  
 if login is None:  
 self.openEntranceError()  
 elif login[1] != self.lineEdit\_7.text().strip():  
 self.openEntranceError()  
 elif login[1] == self.lineEdit\_7.text().strip():  
 self.user = self.lineEdit\_4.text().strip()  
 self.teacher = login[5]  
 self.openMainPupil()

**Заключение**

Как итог:

1. Мы успешно исследовали поставленную нами проблему длительности процесса выдачи заявлений на выход из школы.
2. Мы изучили язык программирования Python, среду программирования PyCharm и написали с их помощью код нашего приложения.
3. Нам удалось создать приложение для создания и обработки заявок на выход из школы, использующее локальную базу данных.
4. Учитель может просматривать заявки.

Планируется сделать анализ причин и количества пропусков у каждого ребенка.

В наших перспективах:

1. Добавить возможность пользователю в роли учителя добавлять учеников вне процесса регистрации.
2. Использовать не локальную базу данных
3. Добавить возможность редактирования заявок (до одобрения/отказа)
4. Добавить возможность редактирования причин отказа

**Список литературы**

<https://docs.python.org>

<https://ru.wikipedia.org>