

**Государственное бюджетное общеобразовательное учреждение города
Москвы "Школа № 1568 имени Пабло Неруды"**

**ПРИЛОЖЕНИЕ ДЛЯ ОТСЛЕЖИВАНИЯ
ПРИСУТСТВИЯ НА ДИСТАНЦИОННОМ УРОКЕ**

Выполнили:

Винокуров Кирилл Владимирович

Серёжин Илья Андреевич

ГБОУ школа №1568, 10-Ж

Руководитель:

Головин Александр Дмитриевич

ГБОУ школа №1568

Учитель информатики

Москва, 2022

Оглавление

Оглавление.....	2
Введение.....	3
Актуальность и похожие приложения.....	4
Цели работы и планы выполнения.....	5
Методика выполнения работы.....	6
Выбор языка программирования и среды разработки.....	6
Связь клиента с сервером.....	7
Графический интерфейс.....	8
Функционал приложения.....	10
Результаты обсуждения.....	11
Описание завершённого продукта.....	12
Список использованных источников.....	13

Введение

В настоящее время всё больше и больше людей проводят огромное количество времени в интернете. Появляются различные сайты, приложения и т.д. Это коснулось и обучения, а именно стали появляться клиенты для обучения дома, для проведения дистанционных уроков в виде онлайн-собраний. Предложение рождает спрос – сейчас нет такого учебного заведения, которое бы не проводило дистанционных уроков для своих обучающихся. Именно поэтому уже сейчас дистанционные уроки стали незаменимы в нашей жизни. Но всё же, как быть учителям, которые всё также должны обучать детей? Как проконтролировать ребёнка, как узнать, чем он занимается и присутствует ли он вообще во время дистанционного урока?

Мы видим возникшую проблему отсутствия реальной информации о присутствии учеников на уроке, к которой добавляются такие “побочные эффекты”, как необходимость тратить время на разыскивание учеников, если их надо спросить, отсутствие какой-либо информации о здоровье и состоянии ребёнка на уроке (если он маленький, а дома никого нет, и вдруг с ним что-то случилось; что делать, как об этом узнать?). Но как же решить эти проблемы?

Мы решили узнать, есть ли сейчас приложения или клиенты, которые смогут предоставить такую информацию. Изучив эту тему, мы пришли к выводу, что их просто нет как таковых, поэтому мы захотели сами сделать подобное приложение, которое действительно может стать очень нужно и незаменимо даже в ближайшем будущем.

Отсутствие, казалось бы, такого нужного инструмента для учителей и, возможно, очень актуального в будущем – прекрасный стимул для работы над созданием такого проекта. Ведь всегда приятно почувствовать себя первопроходцем в чём-то!

Но, разумеется, для успешного выполнения любого проекта нужно обозначить цели и задачи оною. Мы составили для себя план действий и обозначили цели, которые подробнее будут описаны ниже.

Актуальность и похожие приложения

Сейчас есть большое количество приложений, которые дают возможность проведения онлайн-собраний: Microsoft Teams, Zoom, Skype и т.д. Для обучения в школах обычно используются первые два, в то время как Skype больше подходит для индивидуальных занятий, например, с репетитором.

Все эти приложения обеспечивают только видеосвязь и аудиосвязь пользователей. Соответственно, они не могут дать достоверной информации о нахождении кого-либо на занятии. Именно благодаря этой проблеме, мы действительно можем создать новую и полезную утилиту, ведь главное в приложении – актуальность. По причине отсутствия аналогов наша работа может стать очень актуальна.

Взглянем на приложения, которые помогают что-то отслеживать. Ведь у нас имеются приложения для связи, но мы так и не можем знать о присутствии кого-то на уроке. Количество таких приложений огромно, но все они кардинально отличаются от нашей идеи. Это либо шпионские программы, отслеживающие все действия пользователя в сети или компьютере, либо приложения, использующие GPS для получения информации о человеке и его местонахождении, либо же приложения, собирающие информацию о рационе дне, времени сна, настроении в течении дня и т.д.

Цели работы и план их выполнения

Цель нашего проекта – приложение, которое будет выполнять его главную функцию: предоставление информации о присутствии учеников на уроке. При этом оно должно быть понятно и для детей, и для взрослых, чтобы не возникало проблем с ненужной тратой времени для того, чтобы разобраться в графическом интерфейсе приложения.

Для создания нашего проекта мы поставили себе задачи:

- 1) Отладить соединение двух компьютеров для дальнейшей работы.
- 2) Реализовать функционал приложения-мессенджера, передающего сообщения от ученика к учителю.
- 3) Создать простой, интуитивно-понятный для всех возрастов графический интерфейс приложения.

Мы составили план по работе:

1. Связь двух компьютеров в локальной сети (ноябрь 2021)
2. Создание функциональной части приложения (декабрь 2021)
3. Создание графического интерфейса приложения (январь 2022)
4. Итоговая разработка и связь графического интерфейса с функциональными частями (февраль 2022)

Методика выполнения работы

Выбор языка программирования и среды разработки

У нас имеются базовые знания в таких языках программирования, как:

- Python
- C++

Окончательный выбор пал именно на язык программирования Python по таким причинам, как:

- Понятность и простота синтаксиса языка
- Огромное количество библиотек и фреймворков для разнообразных целей и задач
- Возможность выбора среды создания графического интерфейса приложения

В создании своего проекта мы использовали среду разработки PyCharm. Для создания графического интерфейса на языке Python есть всего 2 ключевых варианта работы: Qt Designer и Tkinter. Нами было принято решение использовать Tkinter.

Qt Designer – это приложение, позволяющее создать графический интерфейс приложения, также для облегчения работы с Qt Designer в языке программирования Python есть библиотека, называемая PyQt5.

Tkinter – встроенная библиотека Python, позволяющая создать достаточно простой, не слишком нагруженный лишними действиями, графический интерфейс.

Решение об использовании Tkinter было принято именно потому что одной из наших главных задач является создание простого, всем понятного приложения.

PyCharm – одна из самых популярных сред разработки на языке Python. Она обладает удобным интерфейсом, а также отлично помогает в сборке всех файлов итогового проекта.

Связь клиента с сервером

Определив цели и задачи, стоявшие перед нами, мы решили использовать для их выполнения библиотеку Python под названием Socket. Данный выбор был обоснован возможностями, открывавшимися перед нами, такими как: высокая скорость отклика и производительность, высокая отказоустойчивость, а также гибкость в использовании любой ОС (операционной системой). После реализации возможности обмена данными между клиентом и сервером перед нами возникла проблема интеграции многих пользователей к одному серверу.

```
import multiprocessing as mp
from _thread import start_new_thread
import socket

sock123 = '1'

def callStudent(): # Функция обращению к клиенту
    command = b'aGVsb68='
    print(command)
    global sock123
    if sock123 != '1':
        sock123.send(command)

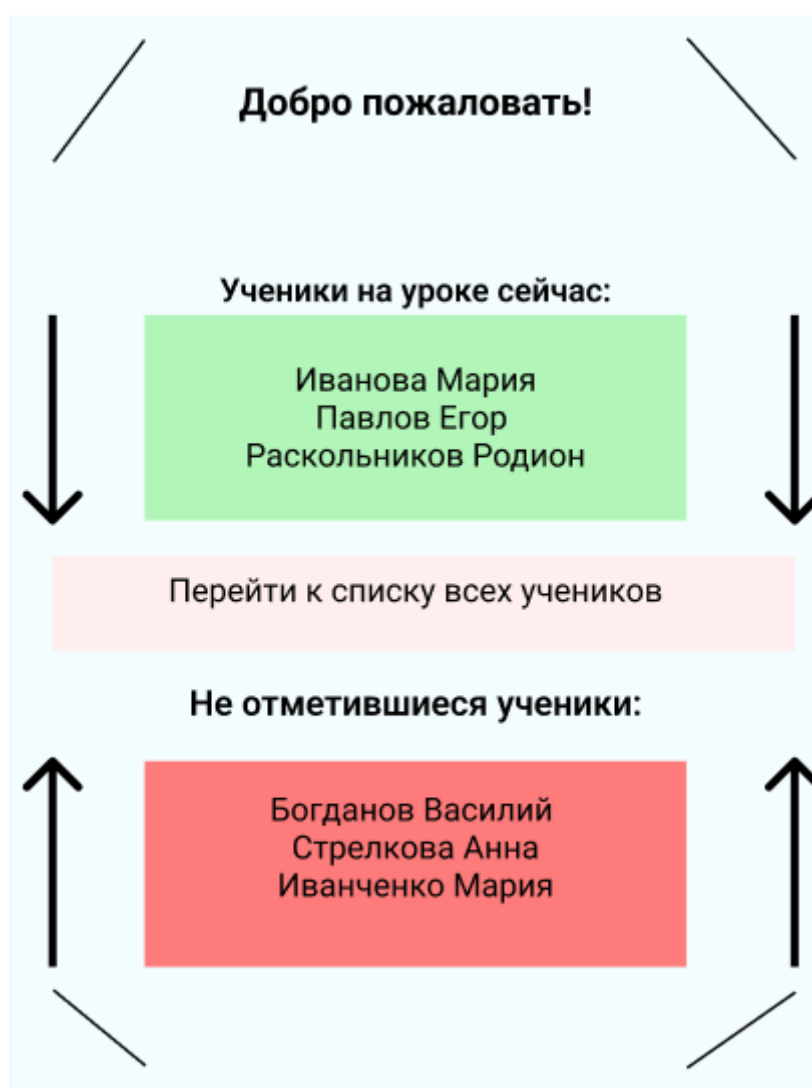
def serverStartUp(host, port): # Функция запуска сервера
    def threaded(connection): # Создание потоков для каждого отдельного клиен
        while True:
            data = connection.recv(1024)
            print(data.decode('utf-8'))
            if not data:
                print('Bye')
                break
            connection.send(data)
        connection.close()

    server = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    server.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
    server.bind((host, port))
    server.listen(5)
    print("Сервер запущен!")
```

Графический интерфейс

Следующим этапом выполнения работы стало создание графического интерфейса для ученика и учителя. Основопологающим элементом нашего приложения является интуитивно-понятный дизайн.

Нами было создано несколько вариантов визуализации, но для того, чтобы интерфейс был действительно простым, мы опросили некоторую группу людей, благодаря мнениям которых мы пришли к конечному варианту.



Неудачный вариант.

Как можно заметить, интерфейс совершенно не интуитивный, разобраться что к чему довольно тяжело. Учителю для открытия полного списка учеников требовалось бы тратить лишнее время.



Окончательный вид приложения.

Около 80% опрошенных выбрали данный дизайн.

Далее мы приступили к преобразованию рисунка в код. Несколько десятков часов было потрачено на изучение теоретической части и ещё более ста на реализацию. В ходе написания кода мы столкнулись с множеством проблем, например, связь объектов интерфейса с сетевым кодом. Каждый баг был

уникальным, для устранения их требовалось глубже и глубже погружаться в тонкость языка программирования.

Функционал приложения

Наша команда создала систему клиент-сервер, поддерживающие двустороннюю связь между учеником и учителем. Для упрощения взаимодействия с программой, мы отказались от таких идей как:

- Встроенный чат
- Голосовая связь
- Подключение дополнительных интеграций

В конечном виде программа получила возможность отмечать ученика, то есть при команде сервера, клиент воспроизводит звук на компьютере ученика и уведомляет его о необходимости ответить учителю. За это отвечает функция `callStudent`.

Также для минимизации действий учителя, наша команда добавили функцию проверки присутствия без участия учителя. Функция основана на таймере с произвольным временем, по истечению которого, воспроизводятся те же действия, что и при команде с сервера. За это отвечает функция `callTimer`.

Ещё одна функция `serverStartUp`, присутствует как в клиентском, так и в серверном приложении. Она отвечает за запуск сервера, то есть включает прослушивание трафика на определённых сетевых портах, что и позволяет функционировать двум частям проекта.

Результаты и обсуждение

В результате работы получилось приложение для организации учебного процесса во время дистанционных занятий.

Наша команда смогла воплотить в жизнь идею удобной и простой контрольной панели. При использовании учитель может быть на сто процентов уверен, что ученик действительно присутствует на уроке. А ученик без каких-либо трудностей может подтвердить своё присутствие.

В ходе выполнения проекта мы научились:

- Работать в коллективе
- Использовать язык Python
- Использовать различные библиотеки языка Python
- Объектно-ориентированному программированию
- Проектированию графического интерфейса
- Поиску информации на иностранных форумах

В будущем мы хотим:

1. Увеличить скорость обработки данных
2. Добавить интеграцию с платформами проведения видео конференций
3. Оптимизировать сетевой код
4. Добавить бесшовный переход между уроками
5. Создать веб-версию приложения

Описание завершённого продукта

Приложения предоставляющее возможности контроля присутствия детей на уроке. Учитель получает доступ к контрольной панель с возможность опроса учеников. Ученик, в свою очередь, решает проблему с ложными обвинениями о прогулах, посредством нашего приложения.

Список использованных источников

1. <https://devpractice.ru/python-lesson-22-concurrency-part-1/> - многопоточность и управление процессами Python
2. <https://pythonru.com/uroki/obuchenie-python-gui-uroki-po-tkinter> - фреймворк для GUI
3. <https://habr.com/ru/post/149077/> - сетевой код реализованный с помощью фреймворка Socket
4. <https://xn--90aeniddllys.xn--p1ai/ochen-prostoj-chatklient-server-na-python/> - сетевой код реализованный с помощью фреймворка Socket
5. <https://pythonworld.ru/moduli/modul-json.html> - фреймворк для парсинга файлов