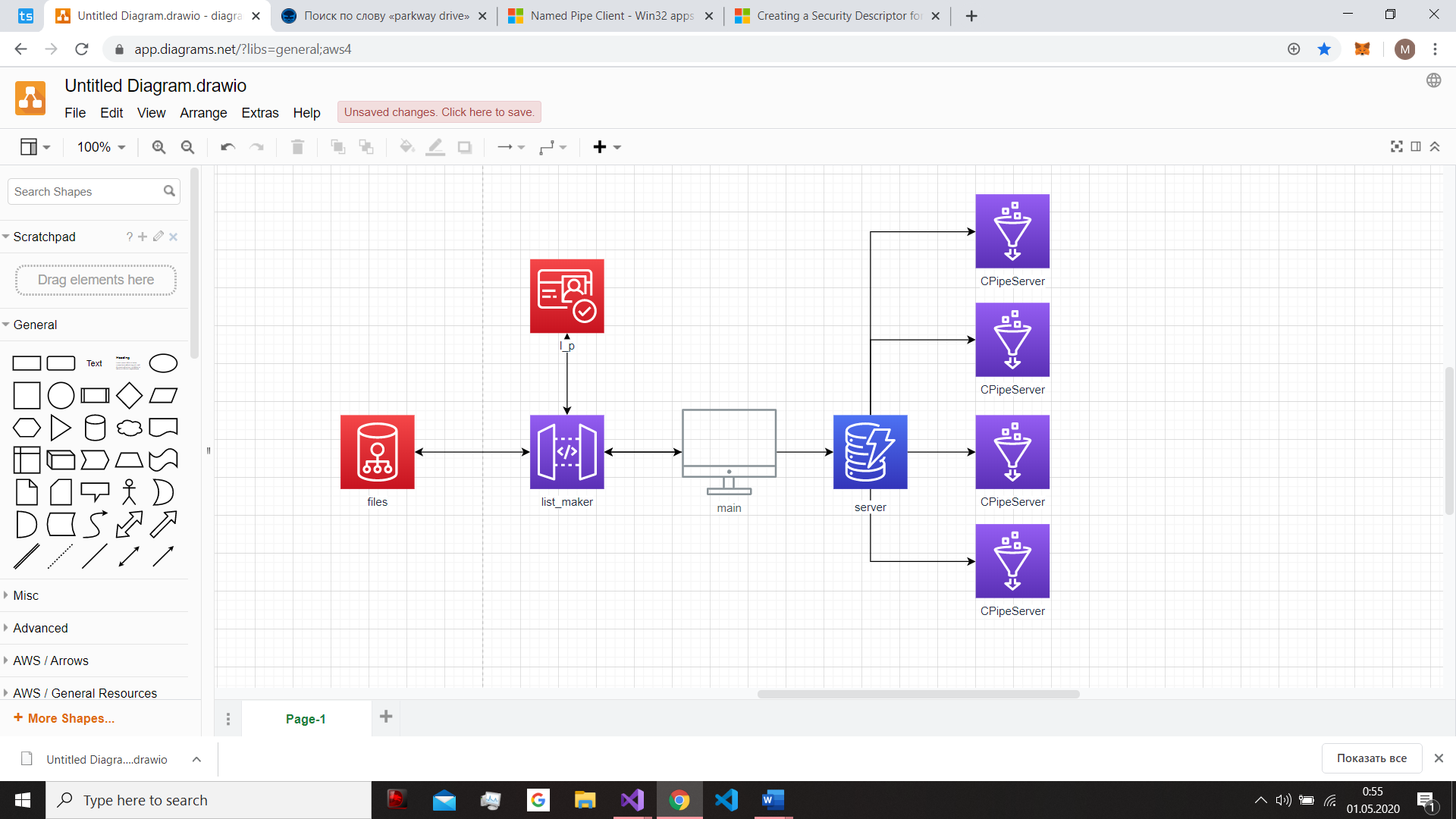
Содержание

[Server 2](#_Toc39436434)

[Struct task 2](#_Toc39436435)

[Class list\_maker 3](#_Toc39436436)

# Server



Сначала Main вызовет list\_maker и обработает файл с логинами и паролями.

Затем вызывается Server который занимается обслуживанием клиентов. Ему также передается список логинов и паролей созданный list\_maker.

Struct task{

Id; //индекс пайпа

Time;

}

В Server, после обработки состояния пайпа, клиент, если пароль неправильный, не будет отключен, а в task \_list будет добавлена задача. Проверка флажка, который определяет нужно ли ждать или можно отправить ответ будет добавлена в обработчик состояния пайпа. Обработка задач (проверка настало ли время отправки) будет вынесена в отдельный поток. У которого будет доступ к task \_list.

//занимается обработкой файла с логинами и паролями

Class list\_maker{

String path;

Unsigned max\_ password \_length;

Unsigned max\_login\_length;

list\_maker(string path){

this.path = path;

}

//получение списка с логинами и паролями, вызывается пользователем

Public: List <l\_p> Get\_list(){

String buf = read\_from\_file();

Get\_info(buf);

Make\_list(buf);

Return list;

}

//считываем файл

Private: String read\_from\_file(){

//считываем файл в строку и возвращаем данные

//throw если что пошло не так

}

//парсит одну строку

Private: l\_p get\_lp (string &str,){

//проверяем длину и корректность данных

//парсим строку, создаем l\_p, возвращаем

//throw если что пошло не так

}

//вычитывает и удаляет первую строку с данными о размере логина и пароля

Private: get\_info (string &buf){

//вычитываем первую строку

//заполняем поля max\_ password \_length, max\_login\_length

//удаляем первую строку

//throw если что пошло не так

}

//создание списка с l\_p;

Private: list<l\_p> Make\_list(string &buf){

list<l\_p> arr;

While{

//берем строки по очереди

//вызываем get\_lp();

//добавляем в arr

}

Return arr;

}

}

//логин + пароль

Struct l\_p{

List<string> Logins;

List<string> Passwords;

}