

актуализации субъективных желаний и позиции подростков, активным их включением в важную для развития личности деятельность и стимулировании личного выбора интересующих профессий. Большим подспорьем самоопределению подростка будет составление им предполагаемого карьерного пути, который приведёт к осознанному выбору профессии и осознанию факта, что он сам принял решение и готов нести за него ответственность, одновременно гордясь этим. Иначе, если такой выбор будет навязан сверху или сделан под влиянием сиюминутных обстоятельств, то не будет и успешного самоопределения наряду с решением проблемы профориентации. Профессиональное самоопределение это далеко не только факт выбора специальности, но и, в том числе, длительный динамический процесс формирования профессионала на протяжении всей жизни.

Список использованных источников

1. Саломатов А.В. Профориентация на базе Интернет- технологий: просто, доступ-но, продуктивно // Профессиональное образование и рынок труда. № 5. 2014.
2. Щербаева Л.А. Проблемы профориентации школьников и их решение // Труды IV Международной научно-практической конференции «Профнавигация молодё-жи», ФГБОУ ВО «КубГТУ», 2021. С. 490-496.
3. Панина С. В. Самоопределение и профессиональная ориентация учащихся. // Основы профессиональной ориентации. 2017. С39-82.

РАЗРАБОТКА КЛИЕНТ-СЕРВЕРНОГО ПРИЛОЖЕНИЯ ДЛЯ ОБМЕНА ТЕКСТОВЫМИ СООБЩЕНИЯМИ

Автор: Карпов Даниил Юрьевич, студент 3-го курса государственного университета «Дубна», филиал «Протвино».

Научный руководитель: Кульман Татьяна Николаевна, к.т.н., доцент кафедры Информационных технологий университета «Дубна», филиал «Протвино».

Аннотация

В данной статье рассматривается разработка программы для обмена сообщениями между пользователями.

Annotation

This article discusses the development of a program for instant messaging between users.

Ключевые слова: TCP, чат, мессенджер, клиент-сервер.

Keywords: TCP, Chat, Messenger, client-server.

Целью работы является создание программы «сервера» и программы «клиента», сервер должен осуществлять хранение и отправку сообщений между клиентами, подключенными к нему, так же сервер должен иметь возможность регистрации пользователей по логину и паролю.

Актуальность

В современном мире программы для обмена сообщениями, стали уже частью жизни. И разработка своей такой программы, является хорошим опытом для каждого программиста.

Постановка задачи

Создание клиент-серверной программы для обмена сообщениями с функционалом:

- регистрация новых пользователей,
- вход в уже созданный аккаунт,
- добавление в друзья,
- отправка сообщений.

Инструментарий

- Microsoft Visual Studio Community 2022
- PostgreSQL
- pgAdmin
- Notepad++
- Язык программирования C#

Основные понятия

Техника взаимодействия. Разработаны две программы, сервер и клиент. Клиенты могут обмениваться сообщениями через общий сервер.

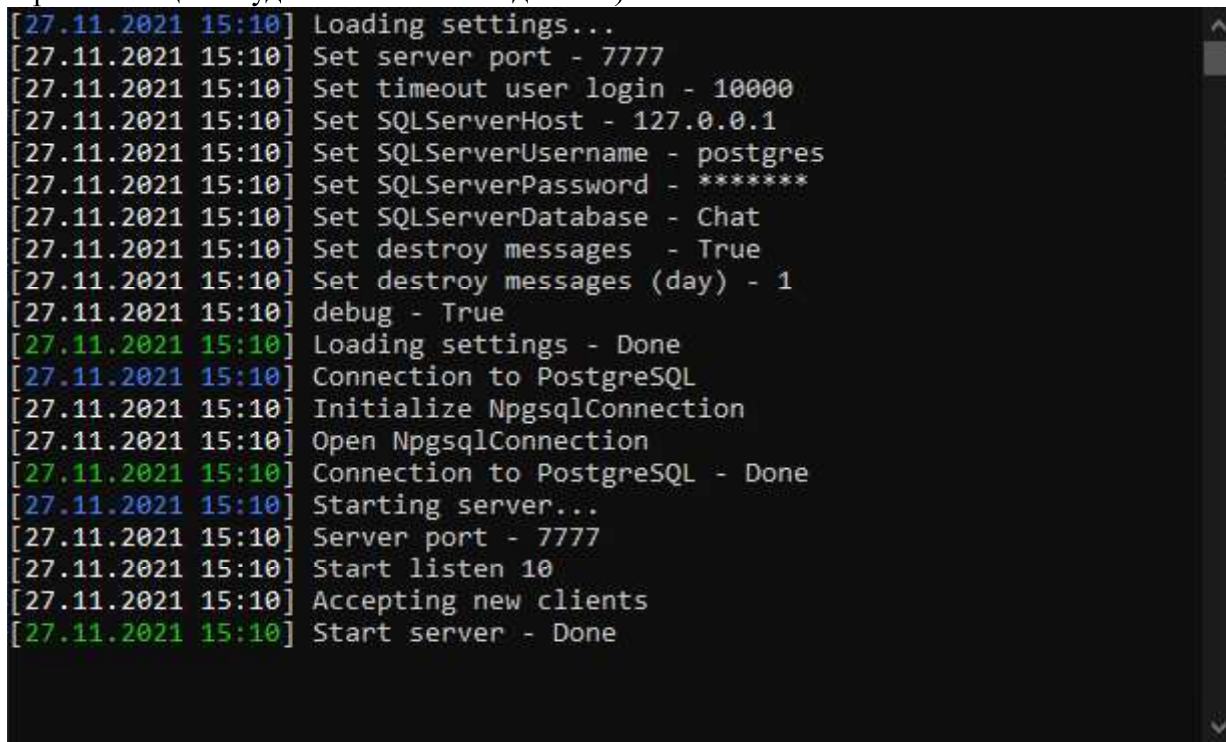
Сервер

Сервер сделан в форме консольного приложения (Рисунок 1), настройка которого происходит через конфигурационный файл settings.cfg (Листинг 1)

Листинг 1

```
server_port 7777
timeout_user_login 10000
sqlserver_host 127.0.0.1
sqlserver_username postgres
sqlserver_password root
sqlserver_database Chat
messages_destroy 1
destroy_messages_day 1
debug 1
```

В этом файле можно настроить порт сервера, время для входа пользователя (по истечении которого, если пользователь не войдет в учетную запись, то он будет отключен от сервера), данные для подключения к базе данных и количество дней хранения сообщений (старые сообщения удаляются из базы данных).



```
[27.11.2021 15:10] Loading settings...
[27.11.2021 15:10] Set server port - 7777
[27.11.2021 15:10] Set timeout user login - 10000
[27.11.2021 15:10] Set SQLServerHost - 127.0.0.1
[27.11.2021 15:10] Set SQLServerUsername - postgres
[27.11.2021 15:10] Set SQLServerPassword - *****
[27.11.2021 15:10] Set SQLServerDatabase - Chat
[27.11.2021 15:10] Set destroy messages - True
[27.11.2021 15:10] Set destroy messages (day) - 1
[27.11.2021 15:10] debug - True
[27.11.2021 15:10] Loading settings - Done
[27.11.2021 15:10] Connection to PostgreSQL
[27.11.2021 15:10] Initialize NpgsqlConnection
[27.11.2021 15:10] Open NpgsqlConnection
[27.11.2021 15:10] Connection to PostgreSQL - Done
[27.11.2021 15:10] Starting server...
[27.11.2021 15:10] Server port - 7777
[27.11.2021 15:10] Start listen 10
[27.11.2021 15:10] Accepting new clients
[27.11.2021 15:10] Start server - Done
```

Рисунок 1 – Сервер

Все данные о созданных аккаунтах пользователей хранятся в базе данных PostgreSQL.

Клиент

Для того чтобы начать пользоваться чатом, необходимо войти в свой аккаунт или создать новый (Рисунок 2).



Рисунок 2 – Форма входа

После входа в аккаунт или регистрации нового, для начала общения необходимо добавить собеседника в друзья, делается это через форму добавления в друзья (Рисунок 3). Необходимо ввести его логин в систему и после отправить запрос. Далее пользователь (которому отправили запрос) может принять или отклонить его.



Рисунок 3 – Форма добавления друзей

Когда пользователи добавили друг друга, они могут начинать общение в чате, который находится на главной форме клиента (Рисунок 4). Главная форма разделена на 2 столбца, слева находится список друзей и кнопка для поиска новых друзей, а справа чат, окно для ввода сообщения и информация о выбранном пользователе.

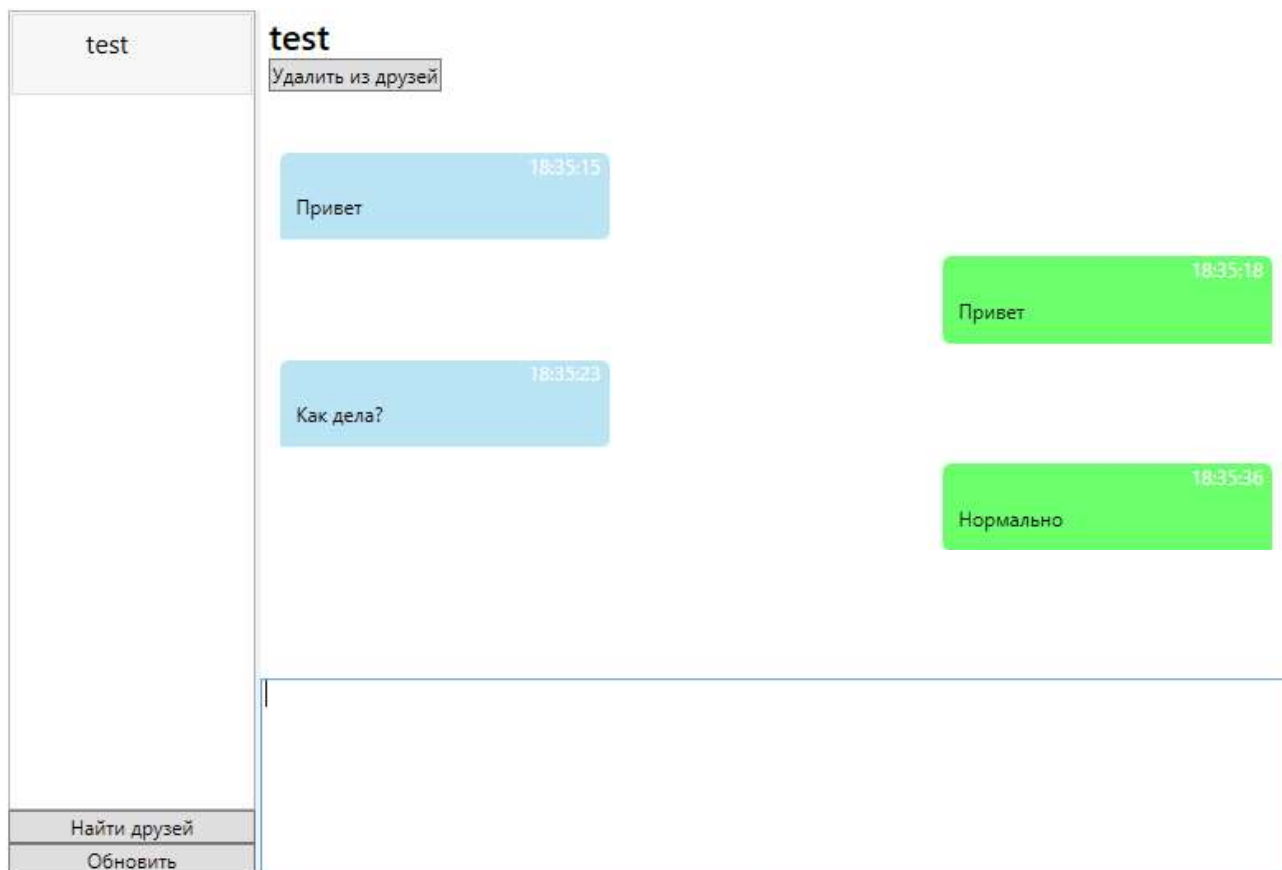


Рисунок 4 – Главная форма клиента

База данных

Для управления базой данных была использована СУБД PostgreSQL.

В базе данных существуют следующие таблицы:

- Users – служит для хранения зарегистрированных пользователей,
- Messages – необходима для записи сообщений,
- Friends – таблица списка друзей.

В таблице Users существует столбец AutKey, он предназначен для того, чтобы хранить ключ пользователя, который генерируется каждый раз, когда пользователь входит на сервер. Этот ключ сохраняется у пользователя в памяти приложения и каждое сообщение отправляется, чтобы сервер мог удостовериться в личности пользователя.

Таблица Messages осуществляет хранение всех сообщений пользователей, когда пользователь отправляет сообщение, сервер записывает его в данную таблицу и если собеседник (которому отправили сообщение) в сети, то сервер так же пересылает сообщение ему.

За хранение информации о всех запросах дружбы и ответах на них отвечает таблица Friends. Когда кто-либо отправляет запрос дружбы, в таблице Friends появляется запись с id отправителя, id того, кому отправляют, и статусом заявки (true – принята, false – не принята).

Выводы:

Была разработана клиент-серверная программа для обмена сообщениями.

Список использованных источников

1. Docs/.NET/Руководство по языку C#/Руководство по программированию — <https://docs.microsoft.com/ru-ru/dotnet/csharp/programming-guide/>
2. Сетевое программирование в C# и .NET - <https://metanit.com/sharp/net/>