### Анализ задачи:

\*Основные компоненты приложения:\*

1. \*Клиентская часть (C/C++):\*

- \*Автоматический запуск:\* Клиент должен автоматически запускаться при входе пользователя в систему. Это можно реализовать путем добавления соответствующей записи в реестр Windows или создания задачи в планировщике.

- \*Фоновая работа:\* Клиент должен функционировать в фоновом режиме, не привлекая внимания пользователя.

- \*Коммуникация с сервером:\* Необходимо выбрать протокол (например, TCP или UDP) для обмена данными с сервером.

- \*Сбор данных:\* Необходимо собрать информацию о всех подключенных клиентах, включая домен, машину, IP-адрес и пользователя.

- \*Отображение времени последней активности клиента.\*

- \*Возможность получения скриншота рабочего стола клиента на сервере.\*

2. \*Серверная часть:\*

- \*Интерфейс:\* Можно использовать веб-технологии (например, HTML/CSS/JavaScript) или создать десктопное приложение на удобном языке.

- \*Список подключенных клиентов:\* Сервер должен отображать список всех подключенных клиентов с информацией о домене, машине, IP-адресе и пользователе.

- Необходимо создать структуру для хранения информации о клиенте.

- Хранить данные о каждом подключенном клиенте.

- Отображать информацию о клиентах при каждом новом подключении.

- \*Отображение времени последней активности:\* Сервер должен получать и отображать время последней активности каждого клиента.

- \*Скриншоты:\* Сервер должен иметь возможность запрашивать и получать скриншоты с клиентских машин.

### Примерный план действий:

1. Создать клиентское приложение для сбора данных и их отправки на сервер.

2. Реализовать серверное приложение для приема данных от клиентов и отображения их в интерфейсе.

3. Реализовать функционал для получения скриншотов.

### Поэтапное выполнение:

1. \*Автоматический запуск при входе пользователя:\*

- На данном этапе необходимо выбрать метод запуска через реестр или планировщик задач. После анализа обоих методов было решено использовать реестр, так как пользователю требуется лишь настроить запуск приложения от имени администратора.

- Реализован метод AddToStartup.

2. \*Фоновая работа:\*

- Вторым решением было создание текстового файла, который фиксировал бы время запуска программы. На этом этапе был разработан tasks.json, который отключал консоль, позволяя программе работать в фоновом режиме, оставаясь невидимой для пользователя.

3. \*Коммуникация с сервером:\*

- В результате анализа протоколов было установлено, что для приложения, отслеживающего работу пользователей, TCP будет более подходящим выбором из-за своей надежности и порядка доставки данных. UDP можно рассмотреть, если приложение требует высокой скорости и может допустить потерю данных.

- Лично я выбрал TCP, поскольку он обеспечивает надежность и последовательность, что важно для данного приложения.

4. \*Сбор данных:\*

- Реализован сбор информации о всех подключенных клиентах (домен, машина, IP, пользователь).