

Техническое задание на разработку авторегистратора для VK

1. Общие сведения

Наименование проекта: Автоматизированная система регистрации аккаунтов ВКонтакте (VK AutoReg)

Платформа: ZennoPoster Pro (браузерная версия с Android-отпечатками)

Язык разработки: C# (кубики/экшены) с приоритетом сохранения траста и скорости

2. Цель проекта

Разработка высокоэффективного авторегистратора для создания аккаунтов ВКонтакте с максимальной эмуляцией действий реального пользователя Android-устройства, обеспечивающего высокий процент выживаемости аккаунтов (ликвида) и минимизацию блокировок.

3. Требования к эмуляции поведения

3.1. Эмуляция Android-устройства

Использование Android-отпечатков (User-Agent, WebGL, Canvas, AudioContext, разрешение экрана, модель устройства)

Эмуляция свайпов, кликов, скроллов с человеческой задержкой (рандомизация 100-500ms)

Имитация работы сенсоров (акселерометр, гироскоп) — по возможности

Эмуляция работы с клавиатурой (ввод текста с рандомной скоростью печати)

3.2. Требования к движку (C# / экшены)

Приоритет — сохранение траста (не нарушая поведенческие паттерны)

Оптимизация скорости выполнения без потери качества эмуляции

Использование C# для сложной логики, экшенов — для визуальных сценариев

Логирование всех действий с возможностью отладки

4. Функциональные требования

4.1. Процесс регистрации

Этап Описание Требования

1. Подготовка Загрузка прокси, получение номера, инициализация Android-отпечатка Прокси проверяется на валидность

2. Переход на сайт Открытие m.vk.com или vk.com с мобильным User-Agent Эмуляция мобильного браузера

3. Заполнение формы Ввод номера телефона, нажатие "Продолжить" Эмуляция ввода с паузами

4. Ожидание SMS Запрос кода из SMS-сервиса hero-sms.com Таймаут ожидания — до 120 секунд

5. Ввод SMS-кода Подтверждение номера, переход к заполнению профиля Обработка ошибок (неверный код, повторная отправка)

6. Заполнение профиля Имя, фамилия, дата рождения, пароль Рандомизация данных (генерация из словарей)

7. Капча Решение через cap.gugu (встроенный или внешний модуль) Автоматический повтор при ошибке

8. Завершение Сохранение cookies, токенов, статуса Закрытие сессии, запись в CSV

4.2. Интеграция со сторонними сервисами

SMS-сервис: hero-sms.com

Получение номера

Ожидание и получение SMS-кода

Отмена/отбраковка номера при ошибке

Капча-сервис: cap.guru

Отправка изображения/аудио капчи

Получение решённого ответа

Обработка ошибок решения (повтор)

4.3. Прогрев аккаунта (минимальный)

| Действие | Описание | Длительность |
|----------|----------|--------------|
|----------|----------|--------------|

| | | |
|-------------|---------------------------------------|----------|
| Авторизация | Вход в аккаунт с сохранёнными данными | 5-10 сек |
|-------------|---------------------------------------|----------|

| | | |
|----------------|------------------------------------|-----------|
| Просмотр ленты | Скролл ленты новостей (2-3 свайпа) | 10-15 сек |
|----------------|------------------------------------|-----------|

| | | |
|--------------------|---|----------|
| Добавление аватара | Загрузка случайного изображения (из базы) | 5-10 сек |
|--------------------|---|----------|

| | | |
|----------|---------------------------------|-----------|
| Подписки | Подписка на 3-5 публичных групп | 10-15 сек |
|----------|---------------------------------|-----------|

| | | |
|--------------------|--------------------------------------|----------|
| Обновление статуса | Размещение статуса (рандомный текст) | 5-10 сек |
|--------------------|--------------------------------------|----------|

Итого прогрев: ~60-90 секунд на аккаунт

5. Технические требования

5.1. Прокси

Поддержка HTTP/HTTPS/SOCKS5

Резиденские или мобильные

Автоматическая проверка прокси перед стартом

Смена прокси при ошибках (таймаут, бан, капча-луп)

5.2. Работа с CSV

| Поле | Обязательное | Описание |
|-------------|--------------|------------------------------------|
| phone_login | ✓ | Номер телефона |
| password | ✓ | Пароль аккаунта |
| api_token | ✓ | API-токен (если получен) |
| cookies | ✓ | Cookies-файл (JSON-строка) |
| status | ✓ | success / error / banned / pending |
| created_at | ✓ | Дата создания в формате ISO |
| updated_at | ✓ | Дата последнего изменения |
| source | ✓ | "hero-sms.com" |
| comment | ✗ | Текст ошибки / комментарий |

Пример CSV:

```
phone_login,password,api_token,cookies,status,created_at,updated_at,source,comment
79001234567,Passw0rd!,token123456,cookies_data.json,success,2026-06-27T14:30:00Z,2026-06-27T14:35:00Z,hero-sms.com,
```

5.3. Логирование

Логирование каждого шага регистрации

Сохранение скриншотов ошибок (опционально)

Ведение общей статистики (успешно/ошибки)

6. Требования к стабильности

Показатель Значение

Процент успешных регистраций $\geq 85\%$

Среднее время на аккаунт ≤ 3 минуты (включая прогрев)

Процент выживаемости (ликвида) $\geq 80\%$ через 24 часа

Поддержка многопоточности ≥ 4 потоков (ZennoPoster)

7. Обработка ошибок

Ситуация Действие

Прокси недоступен Смена прокси, повторный вход

Не приходит SMS Повторная отправка (до 3 раз) → отбраковка номера

Капча решена неверно Повтор до 5 раз → пауза 30 сек → следующий аккаунт

Ошибка валидации поля Генерация новых данных (имя, пароль)

Принудительный бан Сохранить статус "banned", завершить аккаунт

Превышение лимита запросов Пауза 60-120 секунд

8. Дополнительные требования

8.1. Безопасность

Хранение API-ключей в зашифрованном виде (конфиг-файл)

Маскирование чувствительных данных в логах (пароли, токены)

Автоматическая очистка cookies после сессии

8.2. Масштабируемость

Модульная структура (можно подключать другие SMS/капча-сервисы)

Возможность расширения количества потоков

8.3. Интерфейс (опционально)

Простой графический интерфейс для запуска/остановки

Отображение статуса потоков в реальном времени

9. План работ (ориентировочно)

Этап Описание Срок

1. Анализ ТЗ, проектирование архитектуры 1-2 дня

2. Настройка ZennoPoster, Android-отпечатки 2-3 дня

3. Разработка модуля регистрации (C# + экшены) 5-7 дней

4. Интеграция SMS-сервиса (hero-sms) 2 дня

5. Интеграция капча-сервиса (cap.gugu) 1-2 дня

6. Реализация прогрева и CSV-экспорта 2-3 дня

7. Тестирование, отладка, оптимизация 3-5 дней

8. Финализация, сдача проекта 1-2 дня

Итого: ~1-2 недели

10. Критерии приемки

Авторегистрация работает на 4+ потоках

Успешность $\geq 85\%$ (на 1000+ тестов)

Корректный экспорт в CSV со всеми полями

Прогрев занимает ≤ 90 секунд

Аккаунты выживают $\geq 80\%$ через 24 часа (ликвида)

Логирование всех действий

Интеграция с hero-sms.com и car.guru работает стабильно

11. Исполнитель

Исполнитель гарантирует:

Передачу исходного кода (C#-сборка + шаблон ZennoPoster)

Установку и настройку на сервере заказчика

Техническую поддержку в течение 14 дней после сдачи