

УДК 004.584

ИСПОЛЬЗОВАНИЕ ЧАТ-БОТА ДЛЯ АВТОМАТИЗАЦИИ ПРОЦЕССА ОПОВЕЩЕНИЯ СОТРУДНИКОВ АДМИНИСТРАЦИИ Г.КЕМЕРОВО ОБ ОКОНЧАНИИ СРОКА ДЕЙСТВИЯ ИХ ЭЛЕКТРОННЫХ ПОДПИСЕЙ

Кадочигова А.Н., студентка гр. ИТб-171, IV курс
Сыркин И.С., доцент (к.н.) кафедры ИиАПС
Кузбасский государственный технический университет
имени Т. Ф. Горбачева
г. Кемерово

При помощи электронной цифровой подписи (ЭЦП) человек может установить свое авторство на электронный документ для того, чтобы предотвратить несанкционированное изменение данного документа другими лицами и подтвердить подлинность документа.

В настоящее время ЭЦП используют многие предприятия, организации, частные и физические лица, индивидуальные предприниматели и т.д.. В администрации г.Кемерово также используют электронные подписи (ЭП) для проверки подлинности документов и для определения подписавшего его лица. Но ЭЦП имеет ограниченный срок действия в размере 12 месяцев, после окончания которого, электронная подпись будет недействительна и ее пере-выпуск будет невозможен. Несвоевременная подача заявки на продление ЭП может повлечь за собой серьезные задержки всей работы предприятия.

В данный момент процесс оповещения сотрудников об окончании действия их ЭП в администрации г.Кемерово происходит вручную. Сотрудник отдела информационных технологий проверяет список всех сотрудников, владеющих ЭЦП, на наличии истекающих сроков их ЭП и лично информирует их по телефону.

Для автоматизации данного процесса разрабатывается система автоматического оповещения сотрудников с минимальным взаимодействием с человеком. Система имеет название «СОоЭП» - система оповещения об электронных подписях, и будет включать в себя следующие механизмы информирования:

- Информирование по адресу электронной почты;
- Информирование при помощи чат-бота;
- Информирование автозапускаемой программой, находящейся на персональном компьютере сотрудника.

В данной статье разберем, как можно внедрить в систему чат-бота для реализации информирования сотрудников.

Сам чат-бот будет работать в мессенджере Telegram, и написан на языке C# при помощи интегрированной среды разработки Visual Studio. Также в систему будет входить база данных (БД), к которой будет обращаться бот, созданная в СУБД Microsoft SQL Server, и хранящая контактные данные сотрудника и дату окончания действия его ЭЦП.

Предполагается, что программа, отвечающая за работу чат-бота, будет работать на сервере непрерывно, чтобы обрабатывать запросы от пользователей. Раз в сутки программа будет обращаться к БД с запросом на вывод всех сотрудников, чьи ЭЦП в скором времени будут недействительны. После этого, чат-бот разошлет всем сотрудникам, которых необходимо уведомить, сообщение с датой окончания действия ЭП и просьбой подойти в указанный кабинет для ее продления. Диаграмма деятельности процесса оповещения представлена на рисунке 1.

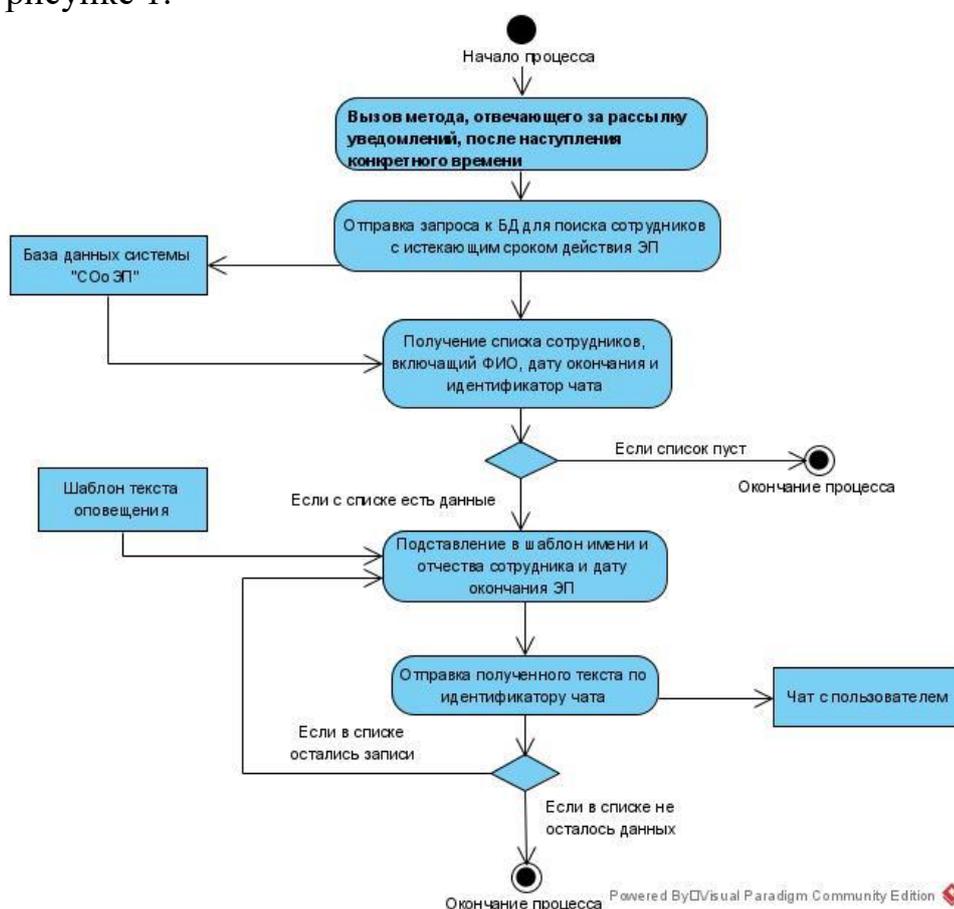


Рисунок 1 – Диаграмма деятельности процесса оповещения.

Чат-боты в Telegram не могут первыми начать переписку с пользователем, чтобы предотвратить распространение спама. Поэтому необходимо соотнести сотрудника, запись о котором уже есть в БД, с пользователем, который добавил чат-бота. Для этого в приветственном сообщении попросим пользователя отправить команду с номером телефона, который он указал при добавлении записи о нем в БД. После этого, чат-бот обратится к БД с запросом на поиск сотрудника с таким номером телефона и последующей записью идентификатора чата в атрибут кортежа данного сотрудника. Если данный

запрос завершился успехом, то в ответ пользователю должно прийти подтверждающее сообщение, обращающееся к пользователю по имени и отчеству. В противном случае, пользователю должно прийти сообщение с текстом, что сотрудник с таким номером телефона не найден. Диаграмма деятельности для идентификации пользователя представлена на рисунке 2.

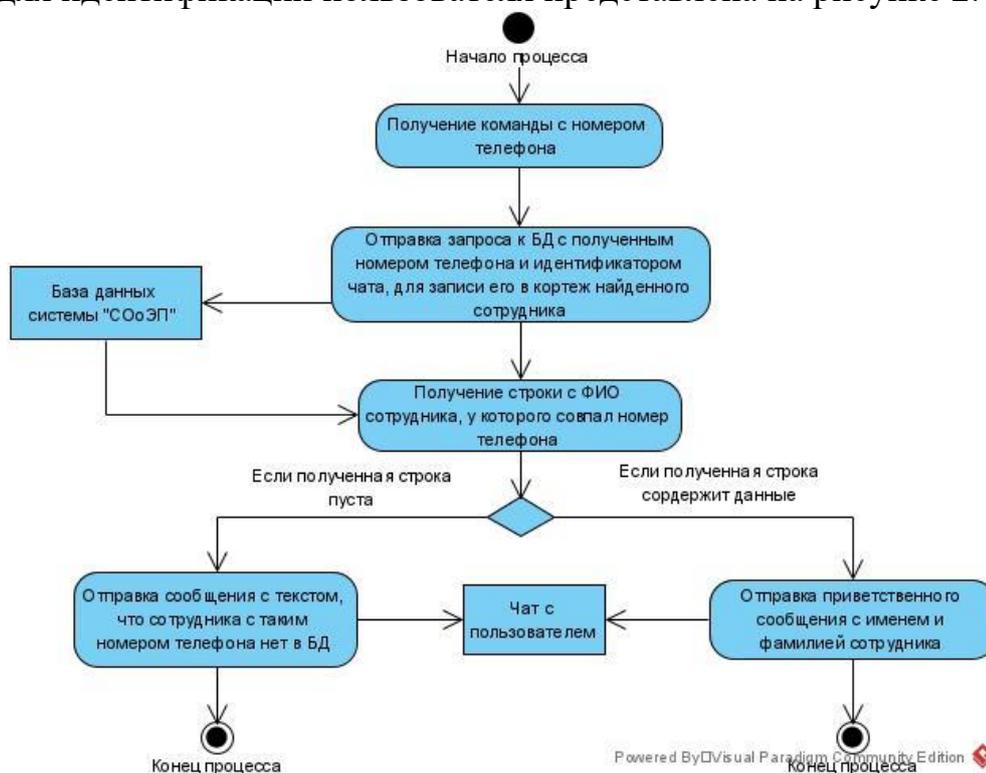


Рисунок 2 – Диаграмма деятельности процесса идентификации пользователя.

Для начала необходимо зарегистрировать чат-бота в telegram. Все создаваемые чат-боты проходят регистрацию через официального бота – BotFather. При помощи данного бота задается название бота, которое будет отображаться пользователям, и уникальное имя для поиска бота, которое обязательно должно оканчиваться на bot (например, @Chat_bot). Помимо настройки внешних параметров бота, можно настроить его функционал.

Общение с ботом BotFather, как и со всеми другими чат-ботами, происходит при помощи команд. Приведем некоторые команды доступные в BotFather:

- /newbot – создание нового чат-бота;
- /setname – изменение название чат-бота;
- /setcommands – указание списка доступных боту команд;
- /setuserpic – изменение изображения профиля чат-бота;
- /setabouttext – добавление информации о боте;
- /token – получение токена для авторизации бота;

После того, как боту будет задано название и уникальное имя, мы получим токен – последовательность символов, которая позволит управлять ботом

через написанное нами приложение, и которая должна содержаться в полной конфиденциальности.

Созданный бот представлен на рисунке 3.



Рисунок 3 – Созданный чат-бот в telegram.

Для написания команд бота и подключения к нему необходимо установить пакет `Telegram.Bot` в среде разработки `Visual Studio`, через систему управления пакетами `NuGet`.

Само приложение, работающее на сервере, может иметь вид консольного приложения с возможностью ввода полученного токена. Или можно создать простой интерфейс, содержащий поле ввода и кнопку для запуска чат-бота.

`Telegram` посылает боту массив обновлений, содержащий в себе идентификаторы чатов, текст сообщений, прикрепленные файлы и другую информацию. Существует два способа получения обновлений. Используя метод `GetUpdate` или прослушивая определенный порт.

В первом случае, метод `GetUpdate` будет находиться в бесконечном цикле, чтобы постоянно обращаться к серверу `Telegram` для получения обновлений. Этот метод реализуется легче, но нагружает сервера `Telegram`. Во втором случае, приложение будет прослушивать определенный порт и при наличии каких-либо обновлений, `Telegram` отправит их приложению. Но для использования данного метода необходим сертификат `SSL`, а так же наличие доменного имени. Чат-бот «СОоЭП» будет использовать первый метод.

Так как в задачи чат-бота входит обработка запросов от пользователей и информирование сотрудников, то метод, обрабатывающий запросы пользователей, нужно вынести в отдельный поток при помощи класса `BackgroundWorker`, чтобы он работал непрерывно в фоновом режиме.

Для начала необходимо подключиться к чат-боту из нашего приложения. Создадим экземпляр класса `TelegramBotClient` и передадим в него токен, полученный ранее, для подключения к боту. Соединение приложения и чат-бота показано на рисунке 4.

```

var SooopBot = new TelegramBotClient(token); //Подключение к боту
int counter = 0;
while (true)
{
    var updates = await SooopBot.GetUpdatesAsync(counter); //Получаем массив обновлений
    foreach (var update in updates)
    {
        var message = update.Message;
    }
}
    
```

Рисунок 4 – Подключение к чат-боту.

Определим, на какие команды будет отвечать чат-бот. В нашем случае будет всего 2 команды:

- /start – автоматически отправляется боту при добавлении пользователем чат-бота. В ответ бот должен отправить инструкцию для авторизации пользователя.
- /auth «номер телефона» - отправляется пользователем вместе с номером телефона для аутентификации. В ответ пользователю приходит приветственное сообщение, обращающееся к сотруднику по имени и отчеству.

Пример обработки команды /start показан на рисунке 5.

```

if (message.Text == "/start")
{
    await SooopBot.SendTextMessageAsync(message.Chat.Id, "Пожалуйста, введите номер телефона, " +
        "который вы указали как контактный, при " +
        "помощи команды: /auth 'Номер телефона'. " +
        "(Пример: /auth 8-123-456-7890)");
}
    
```

Рисунок 5 – Обработка команды /start.

Система «СОоЭП» должна знать, что сотрудник получил уведомление и был с ним ознакомлен. Можно было бы попросить пользователя в ответ на уведомление отправить команду, но лучше использовать кнопки. В Telegram можно сделать два вида кнопок - Inline кнопки, отображающиеся под отправленным сообщением чат-бота, и Reply кнопки, отображающиеся вместо клавиатуры у пользователя. Вывод Inline кнопки показан на рисунке 6.

```

keyboard = new InlineKeyboardMarkup(
    new InlineKeyboardButton[]
    {
        InlineKeyboardButton.WithCallbackData("Принято!", "Callback")
    }
);
    
```

Рисунок 6 – Вывод Inline кнопки.

При отправке уведомления созданная клавиатура передается в метод SendTextMessageAsync, как один из параметров.

Тестирование работы чат-бота «СОоЭП» показано на рисунке 7.



Рисунок 7 – Тестирование чат-бота «СОоЭП».

Благодаря внедрению чат-бота в работу отдела ИТ, исчезнет необходимость искать и лично уведомлять всех сотрудников, чьи ЭЦП необходимо перевыпустить. Это уменьшит рабочую нагрузку на сотрудников отдела информационных технологий и исключит человеческий фактор. Также бот может использоваться и для информирования о других событиях.

Список литературы:

1. Документация Telegram: Боты. [Электронный ресурс] URL: <https://tlgrm.ru/docs/bots> (дата обращения: 23.03.2021).
2. Справочник по Telegram Bot API. [Электронный ресурс] URL: <https://tlgrm.ru/docs/bots/api> (дата обращения: 23.03.2021).
3. Пишем бота telegram на C#. [Электронный ресурс] URL: <http://aftamat4ik.ru/pishem-bota-telegram-na-c/> (дата обращения: 24.03.2021).
4. Моделирование на языке UML в среде Visual Paradigm 14. [Электронный ресурс] – URL: <http://sp.cs.msu.ru/ooap/exer2017.html#exer34> (дата обращения: 26.03.2021).