Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ: Информатика и системы управления

КАФЕДРА: Программное обеспечение ЭВМ и информационные технологии

ОТЧЕТ ПО УЧЕБНОМУ ПРАКТИКУМУ

Студент _____ Тонкоштан А.А.
подпись, дата

Студент _____ Ском Э.П.
подпись, дата

Студент _____ Терехин А.А.
подпись, дата

Студент _____ Исупов А.Д.
подпись, дата

Студент _____ Буланый К.С.
подпись, дата

Ментор _____ Романов А.В.
подпись, дата

Ментор _____ Кононенко С.
подпись, дата

Руководитель практики _____ Оленев А.А.
подпись, дата

Оценка _____

2020 г.

Оглавление

1. Введение

1.1. Задача

Создать бота, который бы пересылал текст новых писем из почты Pronto в мессенджер Telegram.

1.2. Роли

1. Тонкоштан Андрей – руководитель команды, дизайнер;
2. Буланый Константин – разработчик пользовательского интерфейса;
3. Исупов Андрей – разработчик «скрэпера»;
4. Ском Эрик – администратор баз данных;
5. Терехин Федор – системный администратор.

1.3. Менторы

1. Кононенко Сергей;
2. Романов Алексей.

2. Аналитический раздел

В силу необходимости часто проверять бауманскую почту и отсутствия у нее мобильного клиента возникла идея создания бота, пересылающего новые письма в Telegram.

Существующее решение, которое мы обозрели – GmailBot. Это полноценный почтовый клиент для почты Gmail, поэтому весь предоставленный в нем функционал реализовать в рамках практики не представляется возможным из-за объема работы, а также из-за отсутствия у нас API бауманской почты.

Необходимость нового решения проста – отсутствие мобильных клиентов для бауманской почты.

Итак, было решено создать бота, который путем «веб-скрэпинга» собирал бы информацию с сайта бауманской почты о количестве новых писем, о наличии в них файлов и пересылал бы эти данные, так же текст письма в Telegram.

3. Конструкторский раздел

Структурно задача состоит из 4 частей. 3 из них находятся в нашей зоне ответственности: пользовательский интерфейс в Telegram, база данных и администрирование серверной части, а одна – не в нашей: бауманская почта. Каждую компоненту решено отдать под ответственность одного разработчика для упрощения координации в команде.

Декомпозицию с помощью UML-диаграммы можно увидеть в приложении 1.

3.1. Структура компонент

1. Пользовательский интерфейс – это набор команд в Telegram, с помощью которых пользователь сможет авторизироваться и получать информацию о новых письмах.
2. Компонента веб-скрэпинга – интерфейс, с помощью которого бот связывается с почтой и собирает необходимые данные.
3. Компонента базы данных – интерфейс для работы с базой данных, в которой в зашифрованном виде хранятся данные о пользователях.
4. Серверная компонента – надстройка над основной программой для запуска бота на сервере.

3.2. Алгоритмы и структуры данных

Так как Python – это объектно-ориентированный язык, решено использовать классы Python для реализации веб-скрэпинга и работы с базой данных, потому что такой интерфейс независим и удобен.

При написании алгоритмов главный критерий для нас – читаемость. По возможности необходимо декомпозировать задачи

3.3. Тестирование

Тестирование бота в целом и пользовательского интерфейса в частности решено проводить вручную.

Для тестирования веб-скрэпера и базы данных решено написать скрипты.

4. Технологический раздел

4.1. Технические средства

Язык программирования – Python 3. Во-первых, потому что все члены команды знакомы с языком в достаточной степени для выполнения задач, а во-вторых, потому что для Python написано множество библиотек.

pyTelegramBotAPI – библиотека для работы с API Telegram. Основа пользовательского интерфейса.

MongoDB – база данных. Проста в использовании в силу того, что не требует описания схемы таблиц.

Pymongo – библиотека для работы с Mongo DB.

Selenium – библиотека для веб-скрэпинга. Эта библиотека наиболее удобна и проста в использовании, в ней есть все необходимые функции.

Flask – фреймворк для создания веб-приложений на Python. Самый простой на Python. Обладает всеми необходимыми для нас инструментами.

DigitalOcean – хостинг. Бесплатно и достаточно для нас, а также потому что сервера этого хостинга находятся там, где не заблокирован Telegram.

4.2. Модель разработки

Мы выбрали каскадную модель, так как она наиболее удобна для решения нашей задачи.

4.2. Реализация компонент

4.2.1. Пользовательский интерфейс

Разрабатывает Тонкоштан Андрей и Буланый Константин.

Сначала был выбран набор команд для пользователя, а затем, после изучения основ написания функций-команд для ботов Telegram, с помощью pyTelegramBotAPI был создан набор функций, благодаря которым пользователь может обращаться к боту. Также собрали все компоненты воедино.

4.2.2 Веб-скрэпинг

Компоненту веб-скрэпинга разрабатывает Исупов Андрей.

Нашел наиболее удобный браузер – Firefox. Изучил основы написания скрэперов, а затем написал скрэпер вместе с интерфейсом для его использования в программе. В его задачи входит: авторизация пользователя в бауманской почте, сбор количества новых писем, сбор текста писем.

4.2.3. База данных

Компоненту базы данных разрабатывает Ском Эрик.

Изучил работу с MongoDB и Pymongo. Осуществил подключение к базе данных с помощью внутренних функций с возможностью чтения и редактирования ее содержания. Несмотря на то, что cloud.mongodb имеет встроенные функции защиты данных, принял решение написать собственный алгоритм шифрования для более безопасного хранения данных пользователя. Написал интерфейс для использования базы данных в программе.

4.2.4. Серверная компонента

Разрабатывает Терехин Федор. Создал сервер в хостинге DigitalOcean, установил на нем все нужные библиотеки и приложения, настроил браузер для работы на сервере, получил сертификаты и настроил соединение с ботом. А также создал бота в Telegram.

4.3. Реализация тестирования

Для тестирования всего продукта был написан список тестовых случаев, который затем был пройдем вручную.

Для тестирования модулей были написаны скрипты на Python.

4.4. Развертывание продукта

Бот развернут в хостинге DigitalOcean Терехиным Федором.

5. Заключение

При проектировании мы ориентировали на некоторый минимум, который мы должны сделать, а также мы планировали дополнительные функции, которые добавим, если все реализуется успешно.

В целом, минимума мы достигли, решения, которые принимали правильные.

Однако в процессе реализации возникла проблема, которую решить быстро и хорошо не удалось. Это различие в представлении писем от разных сервисов в клиенте Pronto: не всегда удавалось собрать содержание письма. Здесь показал себя главный недостаток веб-скрэпинга: при малейшем изменении интерфейса сайта, с которого собирается информация, программа, собирающая эту информацию, перестает работать. Оптимальным решением стало вывод пользователю сообщения о неудаче считывания письма.

6. Список литературы

1. Документация pyTelegramBotAPI // github.com URL: https://github.com/eternnoir/pyTelegramBotAPI (дата обращения: 31.05.2020).
2. Документация Selenium // readthedocs.io URL: https://selenium-python.readthedocs.io (дата обращения: 25.05.2020).
3. Документация MongoDB // mongodb.com URL: https://docs.mongodb.com (дата обращения: 01.06.2020).
4. Документация Flask // flask.palletsprojects.com URL: https://flask.palletsprojects.com/en/1.1.x/ (дата обращения: 25.05.2020).

Приложение 1