**BC/NW 2021№ 1 (37):8.1**

**СИСТЕМА ОПОВЕЩЕНИЯ СТУДЕНТОВ**

**Журавлёв А.А., Борисова С.В.**

Система оповещения студентов является информационным агрегатором, который объединяет и отправляет студентам сообщения из различных источников с некоторой периодичностью. Такая система позволяет информировать студентов, при этом не отвлекая их потоком сообщений. Сообщения могут быть получены от преподавателей или из .xls выгрузки балльно-рейтинговой системы БАРС. Система имеет архитектуру, основанную на независимых узлах. Каждый узел охватывает свою область и может быть обновлен, развернут или масштабирован независимо. Такая архитектура предоставляет лучшую масштабируемость и гибкость в выборе технологий

**Функциональные особенности разработанной системы**

* Система оповещения студентов предназначена для отправки сообщений по почте, ВКонтакте, Telegram и SMS.
* Все сообщения отправляются в назначенное время с интервалом между отправкой.
* Пользователь может выбрать, куда он желает получать сообщения (на почту, SMS, ВКонтакте, Telegram).
* Перед отправкой сообщения, предназначенные одному адресату, агрегируются в одно сообщение.
* Система получает сообщения от пользователей и из .xls файлов.

Процесс добавления студента в систему можно разбить на два этапа:

1. Сотрудник университета добавляет студента в систему и указывает его данные.
2. Студент активирует свой аккаунт в системе. Для этого он меняет пароль и указывает свои контакты, куда желает получать сообщения (SMS, почта, ВКонтакте, Telegram).

После активации аккаунта студент может получать сообщения по указанным контактам.

Рассмотрим, чем эта система лучше других. Сравнивать будем с классическими средствами оповещения. Под классическими средствами оповещения будем понимать те средства, которые в настоящее время чаще всего используются для оповещения, то есть мессенджеры, социальные сети и почту.

1. Классические средства не имеют функции объединения нескольких сообщений в одно, что не позволяет отправлять большое количество сообщений, так как, вероятно, многие из них будут проигнорированные пользователем
2. Классические средства не предоставляют возможность выбора. Например, если был выбран Telegram, то сообщения будут приходить только в Telegram.
3. Классические средства не имеют функционала периодической отправки сообщений.
4. Мессенджеры и социальные сети не имеют удобных средств для отправки сообщений по группам, институтами и всему ВУЗу.

Все сообщения с момента их создания до момента их отправки хранятся на Google Drive в формате JSON (рис. 1.).



Рис. 1. Сообщение в формате JSON

JSON (JavaScript Object Notation) – это текстовый формат представления данных в нотации объекта JavaScript.

Есть два основных элемента объекта JSON: ключи и значения.

* Ключи должны быть строками. Они содержат последовательность символов, которые заключены в кавычки.
* Значения являются допустимым типом данных JSON. Они могут быть в форме массива, объекта, строки, логического значения, числа или значения null.

**Узлы информационной системы**

Система состоит из четырех узлов: узел взаимодействия с пользователем, узел отправки сообщений, узел преобразования .xls в JSON и узел-агрегатор. Оповещение происходит с помощью отправки студентам сообщений по почте, SMS, ВКонтакте и Telegram (рис. 2.).

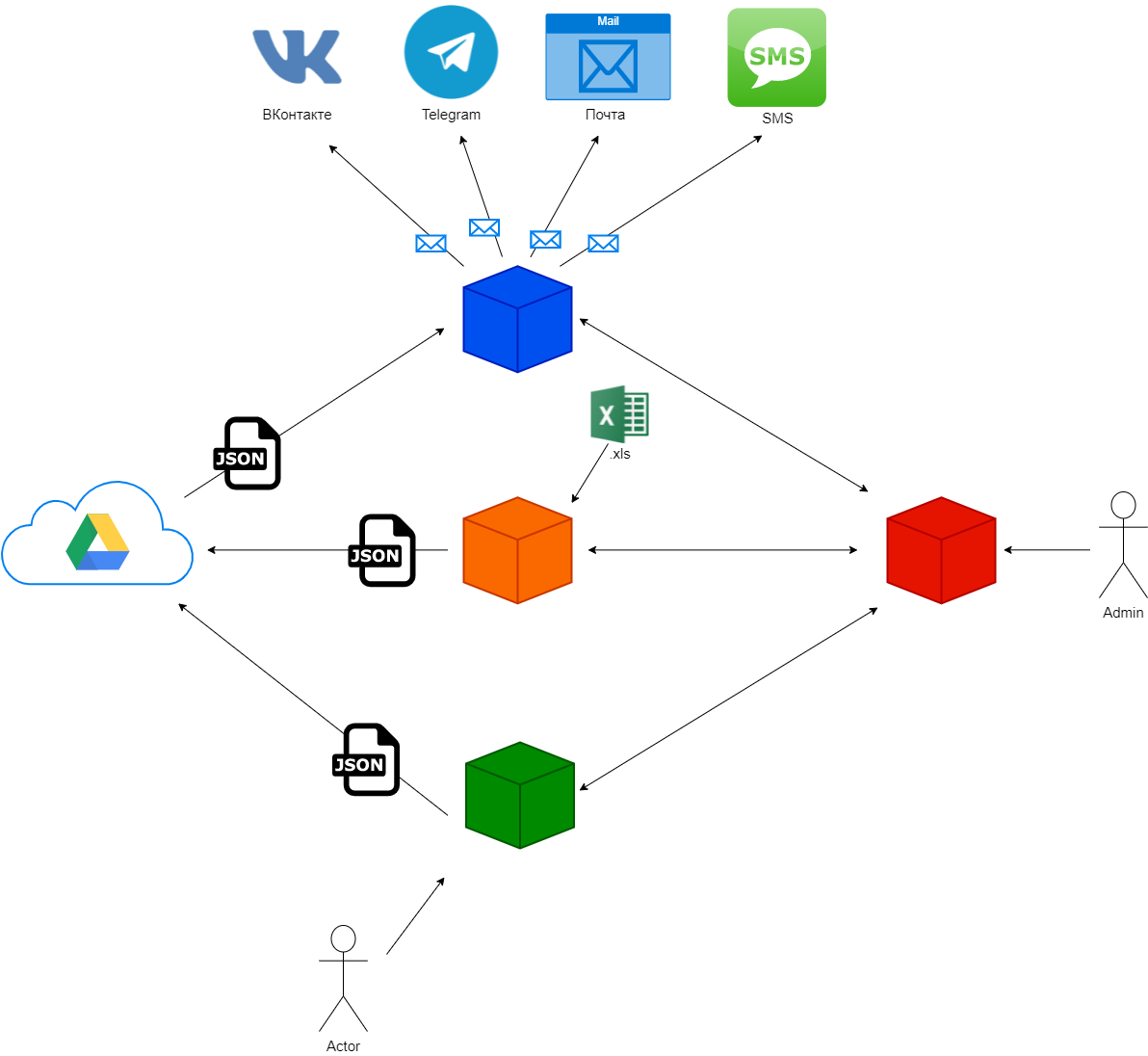


Рис. 2. Узлы информационной системы

* Узел взаимодействия с пользователем предназначен для работы с пользователем (зеленый узел).
* Узел преобразования .xls в JSON нужен для создания сообщений из .xls файлов (оранжевый узел).
* Google Drive используется для хранения сообщений.
* Узел отправки сообщений агрегирует сообщения и отправляет их по контактам студента (синий узел).
* Узел агрегатор отслеживает состояние системы и управляет ей (красный узел).

До момента отправки сообщения хранятся на Google Drive в формате JSON. Для их отправки требуются контакты студента, которые хранятся в базе данных. Узел взаимодействия с пользователем предоставляет пользовательский интерфейс для создания сообщения. Он также преобразует данные, введенные пользователем, в JSON-файл и загружает его на Google Drive.

**Информационные потоки, генерируемые узлами**.

Узлы взаимодействия с пользователем и преобразования .xls в JSON сохраняют на Google Drive сообщения в формате JSON. Узел отправки сообщений загружает из Google Drive эти сообщения и обрабатывает их (рис. 3.).

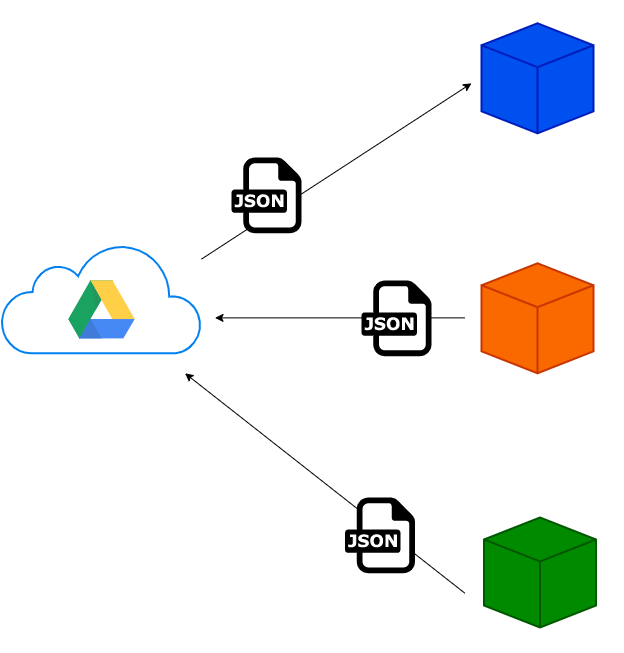
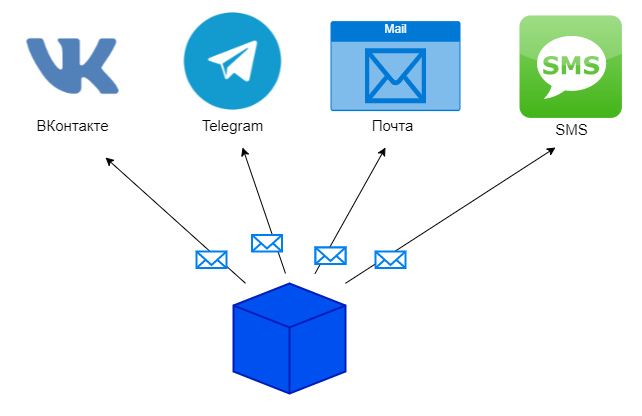


Рис. 3. Использование Google Drive узлами

Узел отправки сообщений отправляет сообщения в ВКонтакте, Telegram, по почте и SMS (риc. 4.).



Риc. 4. Отправка сообщений в ВКонтакте, Telegram, по почте и SMS

Узел агрегатор собирает логи со всех узлов и обменивается с ними данными по HTTPS (рис. 5.).

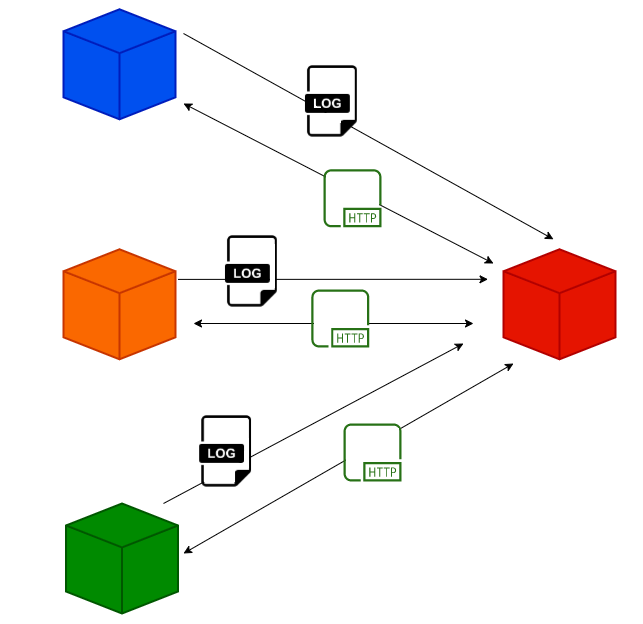


Рис. 5. Взаимодействие узла агрегатора с остальными узлами системы

**Работа с пользователем**

Для работы с пользователем было написано веб-приложение, на языке программирования Python с использованием фреймворка Django.

Функции веб-приложения:

* Создание пользовательских сообщений и загрузка их на Google Drive.
* Добавление новых студентов в систему.
* Указание контактов студента.

Имеются страницы для добавления студента, изменения настроек пользователя, отправки сообщения, активации аккаунта и входа в систему. Все страницы имеют адаптивный дизайн (рис. 6*.*).

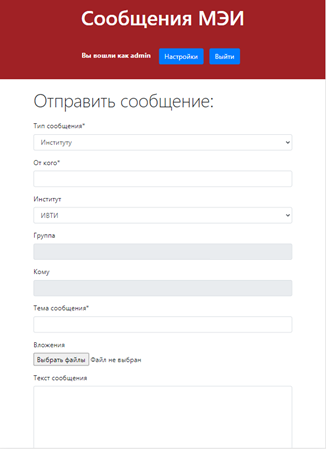


Рис. 6. Пример страницы веб-приложения

**Создание сообщений из .xls файлов**

Для создания сообщений из .xls файлов было написано приложение на языке программирования C#. Приложение извлекает данные их Excel таблицы (в данном случае рассматривается .xls выгрузка из БАРСа), создает из них сообщения (рис. 7*.*) и загружает эти сообщения на Google Drive. Приложение запускается периодически с помощью планировщика Windows. Для того, чтобы можно было получать информацию о состоянии приложения и управлять им, было написано Web API на языке программирования C# с использованием ASP NET Core.

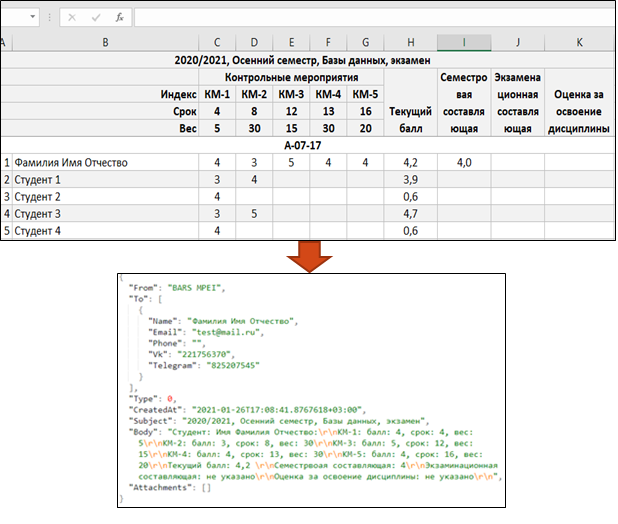


Рис. 7. Создание сообщения из .xls файла

**Отправка сообщений**

Отправка сообщений по почте, SMS, ВКонтакте, Telegram происходит с помощью приложения, написанного на языке программирования C# (рис. 8.). Приложение загружает с Google Drive сообщения, объединяет сообщения с общим адресатом и отправляет их по контактам, указанным в сообщении. Это приложение также запускается периодически с помощью планировщика Windows. Для взаимодействия с приложением и планировщиком Windows имеется Web API, написанный на языке программирования C# с использованием ASP NET Core.

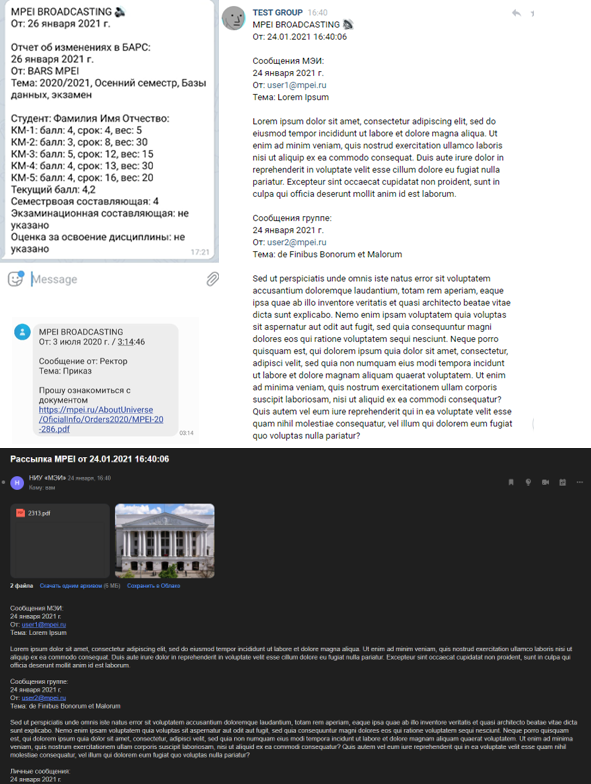


Рис. 8. Пример отправленных сообщений

**Управление и мониторинг состояния системы**

Для управления системой и отслеживания ее состояния было написано веб-приложение на языке программирования Python с использованием Django. Приложение может изменять настройки планировщиков Windows, которые запускают приложения отправки и создания сообщений из .xls файлов. Также приложение получает информацию о состоянии системы. Имеется возможность начать обработку .xls файлов или отправку сообщений немедленно, в обход времени, запланированного планировщиком. Также собираются логи с приложений и предоставляется интерфейс для работы с ними.

Для удобной работы с логами в данной системе нужно:

1. Привести логи к единому формату.
2. Хранить логи в одном месте.
3. Сделать централизованную фильтрацию и обработку логов.

Для решения этой задачи был использован стек ELK (рис. 9.). Стек состоит из трех технологий, которые выполняют следующие функции:

1. Logstash – сбор, преобразование и фильтрация логов.
2. Elasticsearch – хранение логов.
3. Kibana - визуализация данных из Elasticsearch.

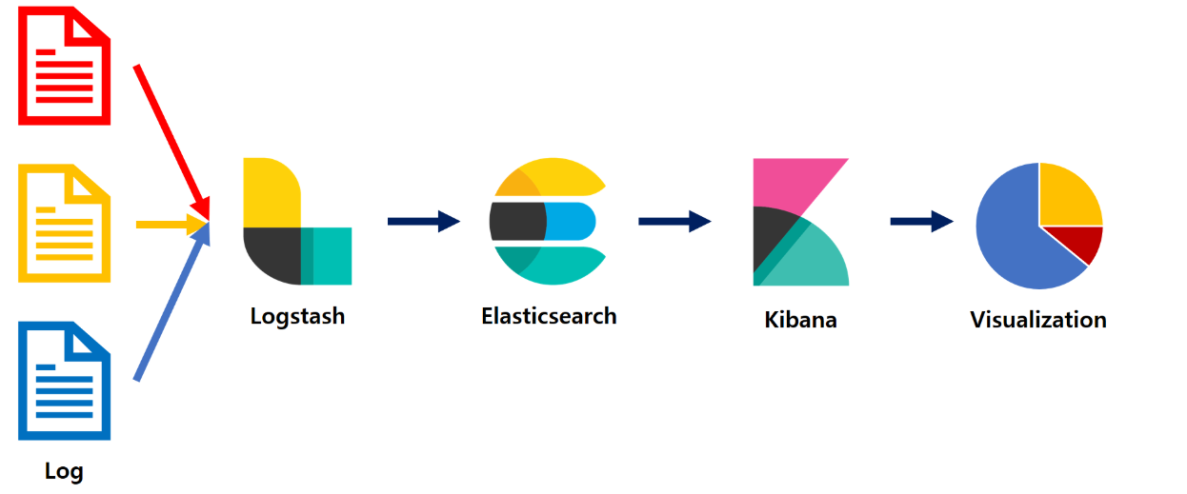


Рис. 9. ELK стек

Разработанная систем оповещения студентов внедрена в Национальном Исследовательском Университете «МЭИ», позволяет удобно и эффективно оповещать студентов и обладает возможностями масштабирования.

**Литература**

1. Рихтер Д. CLR via C#. — 4-е изд. — СПб.: Издательский Дом ПИТЕР, 2019. 896 с

2. Меле А. Django 2 в примерах — ДМК Пресс, 2019. 408 с.