

организовать локальное взаимодействие между устройствами. Полученный результат может быть использован в качестве основы для дальнейших исследований и разработок в области интеллектуальных и обучающих игр.

В. А. Калиниченко

*Науч. рук. Е. И. Сукач,
канд. техн. наук, доцент*

ОБ ИНТЕГРИРОВАННОЙ ВЕБ-ПАНЕЛИ TELEGRAM-БОТА ДЛЯ ОПЕРАТИВНОГО ИНФОРМИРОВАНИЯ О ПЕРЕБОЯХ В ЭЛЕКТРОСНАБЖЕНИИ

В докладе рассматривается разработка программного средства, предназначенного для оперативного информирования конечных потребителей о перебоях в электроснабжении. Акцент сделан на улучшении взаимодействия с системой для административного персонала за счет использования современных возможностей платформы Telegram.

Для решения поставленной задачи был разработан Telegram-бот [1]. Его серверная часть реализована на языке программирования Python с использованием библиотеки pyTelegramBotAPI (telebot) для взаимодействия с Telegram Bot API. Для хранения пользовательских данных и настроек использовался формат JSON. Функционал бота включает:

- регистрацию пользователей с разделением на две основные категории: граждан (с обязательным указанием района и микрорайона проживания для получения целевых уведомлений) и сотрудников (обладающих административными правами);
- автоматическую отправку уведомлений гражданам, проживающим в районах и микрорайонах, затронутых отключением электроэнергии;
- административный интерфейс, реализованный в виде Telegram Web App.

Последний представляет собой веб-панель, созданную с использованием HTML, CSS и JavaScript, которая позволяет сотрудникам через удобный графический интерфейс управлять ссылкой оповещений и добавлять новых сотрудников в систему.

Новизна данного подхода заключается в комбинации автоматизированного процесса информирования граждан с предоставлением сотрудникам интуитивно понятного веб-интерфейса для администрирования, интегрированного непосредственно в Telegram. Это позволяет минимизировать задержки в передаче информации, обеспечить ее целевую доставку и упростить управление системой для ответственного персонала. Использование Telegram как платформы обеспечивает широкую доступность сервиса без необходимости установки пользователями дополнительных приложений.

С точки зрения универсальности, разработанное решение характеризуется гибкостью и возможностью масштабирования. Серверная часть бота является кроссплатформенной. Интерфейс для граждан остается простым и понятным, а веб-панель для сотрудников значительно повышает удобство администрирования.

Практическая значимость программного обеспечения заключается в повышении уровня осведомленности населения, что позволяет гражданам своевременно реагировать на перебои с электроснабжением и минимизировать их негативные последствия. Для поставщиков электроэнергии данное решение предлагает эффективный инструмент для информирования абонентов, способный снизить нагрузку на службы поддержки.

Литература

1 Калиниченко, В. А. Telegram-бот для оперативного информирования о перебоях в электроснабжении / В. А. Калиниченко // Новые математические методы и компьютерные

технологии в проектировании, производстве и научных исследованиях : материалы XXVIII Республиканской научной конференции студентов и аспирантов (Гомель, 17–19 марта 2025 года) / М-во образования Республики Беларусь, Гомельский гос. ун-т им. Ф. Скорины ; редкол.: С. П. Жогаль (гл. ред.) [и др.]. – Гомель : ГГУ им. Ф. Скорины, 2025. – Ч. 1. – С. 168–169.

B. B. Козликовская

Науч. рук. **E. M. Березовская**,
канд. физ.-мат. наук, доцент

РЕАЛИЗАЦИЯ СИСТЕМЫ УВЕДОМЛЕНИЙ В КРОССПЛАТФОРМЕННОМ ПРИЛОЖЕНИИ FAIL: РАЗЛИЧИЯ РАБОТЫ С PUSH-УВЕДОМЛЕНИЯМИ НА ANDROID И IOS

В современном мобильном образовании push-уведомления играют ключевую роль в удержании пользователей и поддержании регулярности обучения. Приложение FAIL использует кроссплатформенный подход с Kotlin Multiplatform Mobile (KMM), что позволяет объединять бизнес-логику и адаптировать интерфейс для Android и iOS. Основная цель работы – исследовать особенности реализации push-уведомлений на разных платформах и выявить отличия, влияющие на эффективность взаимодействия с пользователем.

Push-уведомления делятся на локальные и серверные. Локальные уведомления формируются и отправляются самим приложением без подключения к внешним сервисам, тогда как серверные требуют взаимодействия с облачными платформами (Firebase Cloud Messaging для Android и Apple Push Notification Service для iOS). Кроссплатформенная разработка на KMM предполагает вынесение общей бизнес-логики уведомлений в общий модуль, а платформенные модули реализуют конкретные методы интеграции с FCM и APNS.

На Android FCM позволяет управлять приоритетом уведомлений, обрабатывать события в фоне и отправлять данные в формате JSON. Особенностью является необходимость запроса разрешений у пользователя, а также правильная работа при закрытом приложении или в фоне. На iOS APNS требует наличия сертификатов и токенов устройства, а также ограничивает частоту фоновых обновлений. Дополнительно iOS накладывает ограничения на кастомизацию уведомлений, например, звук и значок должны соответствовать требованиям системы.

Практическая реализация системы push-уведомлений в приложении FAIL включает создание общего модуля уведомлений, который отвечает за планирование, обработку, отправку и отображение уведомлений на обеих платформах. Этот модуль обеспечивает централизованное управление событиями, позволяет настраивать время отправки уведомлений, приоритет, тип уведомления и его содержимое, а также поддерживает возможность локальных и серверных уведомлений. Для корректной интеграции с платформенными сервисами на уровнях Android и iOS разработаны специальные адаптеры: на Android используется Firebase Cloud Messaging (FCM), обеспечивающий работу уведомлений в фоне, обработку данных в формате JSON, а также настройку приоритета и кастомизации визуального и звукового сопровождения; на iOS применён Apple Push Notification Service (APNS), который требует наличия сертификатов и токенов устройства, ограничивает частоту фоновых обновлений и накладывает ограничения на настройку внешнего вида уведомлений.

Кроме того, модуль уведомлений в приложении FAIL реализует обработку событий с учётом состояния приложения: уведомления корректно отображаются как при активном запуске приложения, так и при его нахождении в фоне или закрытом состоянии, обеспечивая непрерывность взаимодействия с пользователем. Таблица 1 демонстрирует основные различия в реализации ключевых функций уведомлений на Android и iOS, а также особенности их работы и ограничения, накладываемые платформами.