Разработка мобильного приложения для свободной системы RunaWFE Free

Алиса Киселева
НИУ МИЭТ
Андрей Михеев
ООО «Процессные технологии»
RunaWFE Free

RunaWFE Free

RunaWFE Free - это система управления бизнес-процессами (СУБП) с открытым исходным кодом.

Бизнес-процесс отображается в виде графической схемы переходов между действиями или задачами.

Пользователи системы получают роли, описанные в схеме и по мере исполнения бизнес-процесса отмечают статус своих задач.

В чате бизнес-процесса пользователи могут уточнить информацию.

Мобильное приложение

Сейчас практически у каждого есть смартфон
В большинстве случаев — на ОС Android

Не всегда задания бизнес-процесса выполняют
люди с доступом к ПК
Например,
в поездках (водители, те, кто часто в разъездах)
или на производстве

Приложение для Android

Изначальное положение вещей:

- Язык программирования: Kotlin
- Приложение открывает указанную в настройках ссылку на сервер системы RunaWFE
- Основной режим работы: сайт в контейнере WebView

Приложение для Android

Проблема: новые задачи и сообщения в чате видны на сайте, но нет уведомлений в мобильной версии.

Уведомления избавят пользователя от ручной проверки, сэкономят время

- Пользователю приходится обновлять список задач вручную и проверять появление новых
- Для сообщений в чате есть оповещение на сайте, но оно видно только там
- Код существующих решений (нотификаторы на Java и на C++) было затруднительно адаптировать для Android

Реализация оповещений в Android

1. Клиент периодически опрашивает сервер о новых данных (polling)
2. Данные запрашиваются по REST-API (библиотека Retrofit, добавлен API-клиент)
3. Этим занимается отдельный сервис
4. Тем временем основное взаимодействие с приложением – в браузере WebView

Выбор решений

Почему Polling

- Это очень просто
- Надежно
- Независимость от сторонних сервисов

Выбор библиотеки для работы с REST

Это скорее выбор комбинации библиотек:

- Работы с HTTP (OkHttp, Volley, Ktor, Retrofit)

*Остальные библиотеки основаны (или могут использовать) OkHttp

- Сериализации (Gson, Jackson, Moshi, Kotlinx Serialization)

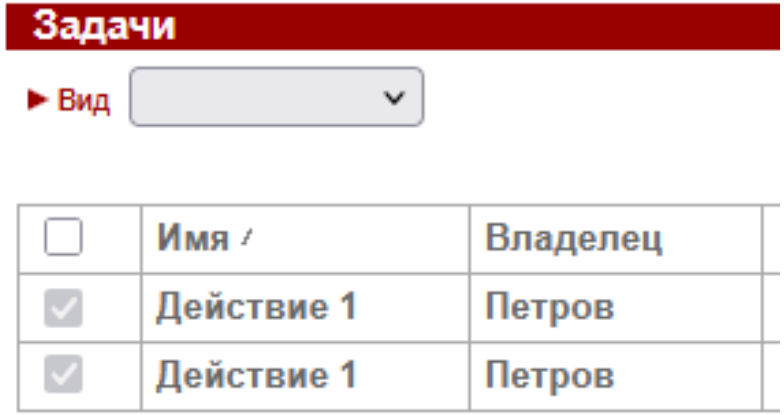
Выбор библиотеки для работы с REST

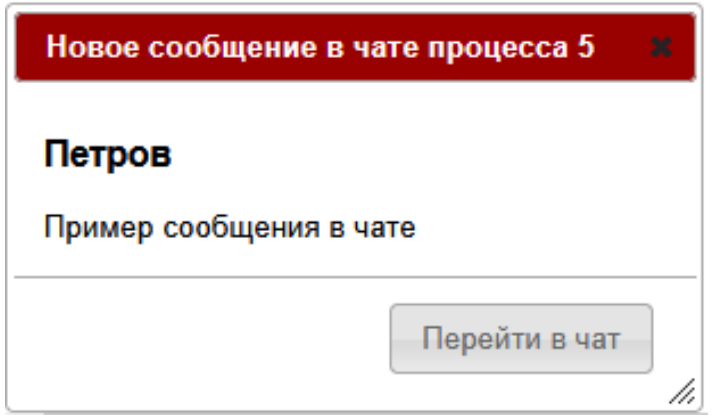
- Volley (Google) — последняя версия (1.2.1) вышла в 2021 году
- Ktor (JetBrains) — часть Kotlin Multiplatform, у которой отдельная система библиотек
- Retrofit (Square) — работает с любой библиотекой сериализации, интерфейс работы с REST в виде аннотаций

Была выбрана библиотека Retrofit

Два канала уведомлений

- Первый канал – оповещения о новых задачах
- Второй канал оповещений – новые сообщения в чатах





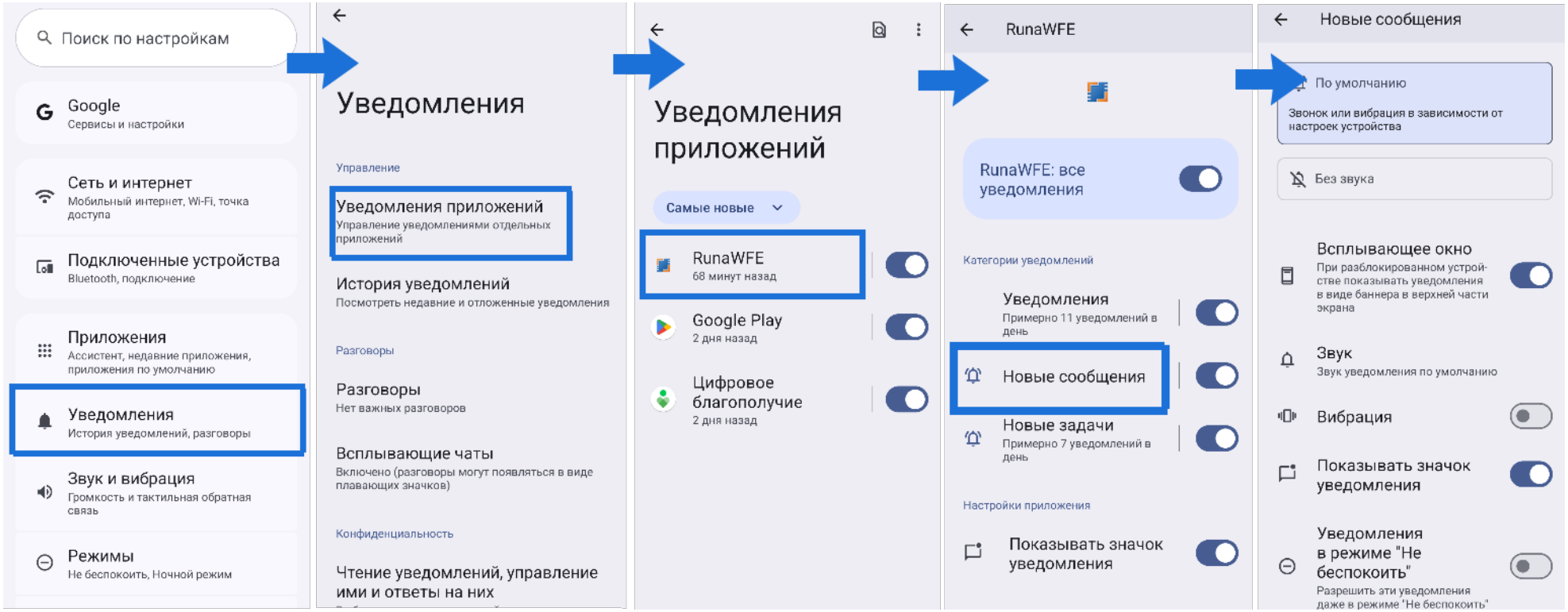
Особенности работы с каналами уведомлений Android

С Android 8.0 (API 26) все уведомления привязаны к каналу

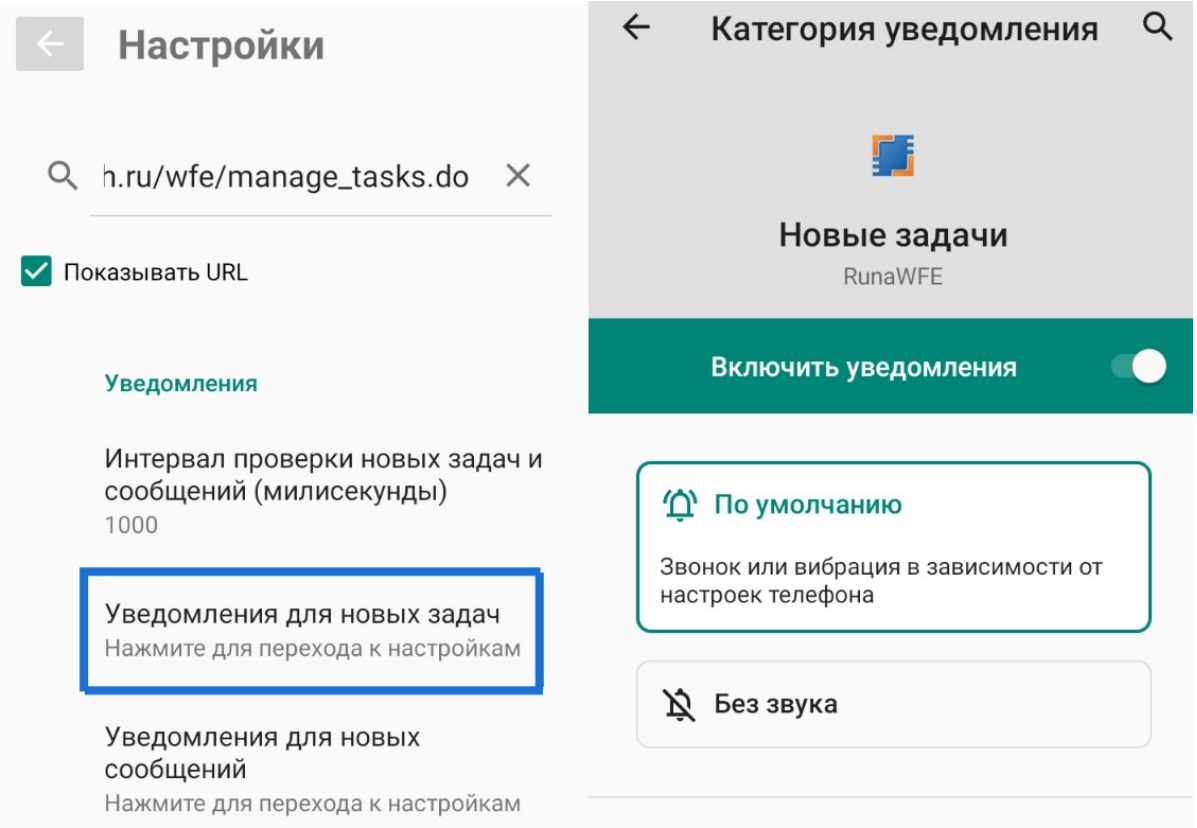
После создания канала можно изменить только его название и описание

Лишь пользователь может менять сигнал (звук, приоритет, вибрация)

Но сначала придется найти нужный раздел в настройках



Можно открыть точный раздел настроек, используя android.provider.Settings




```
intent.setAction(Settings.ACTION_CHANNEL_NOTIFICATION_SETTINGS)
intent.putExtra(Settings.EXTRA_CHANNEL_ID, channelId)
```

Генерация элементов настроек

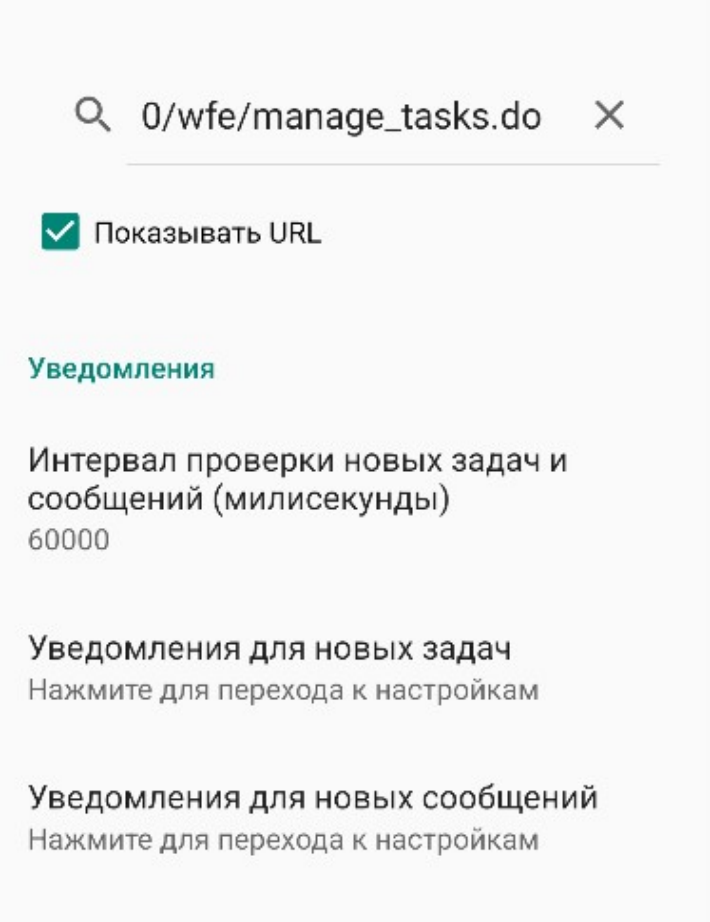
С помощью библиотеки androidx.preference

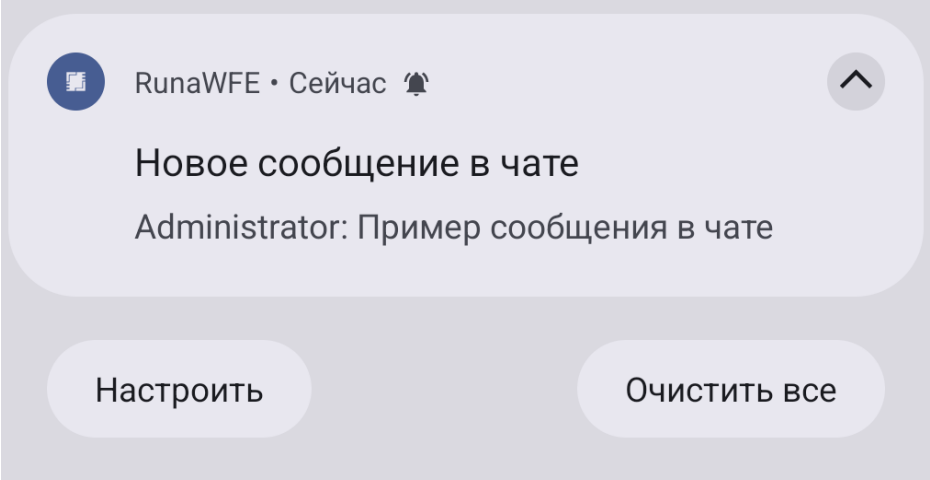
настройки приложения можно описать в формате .xml

Группа настроек уведомлений описана так

И добавлена как отдельный фрагмент: класс-наследник PreferenceFragmentCompat()

Результаты


0/wfe/manage_tasks.do
Показывать URL
Уведомления
Интервал проверки новых задач и сообщений (милисекунды)
60000
Уведомления для новых задач
Нажмите для перехода к настройкам
Уведомления для новых сообщений
Нажмите для перехода к настройкам


RunaWFE • Сейчас
Новое сообщение в чате
Administrator: Пример сообщения в чате
Настроить
Очистить все

Дальнейшие планы

- Тестирование, доработка и развитие имеющейся кодовой базы приложения
- Создание приложений для других мобильных ОС помимо Android

Простое приложение для ALT Mobile

```c
static void run(GApplication *app, gpointer *user_data) {
    gchar *website_url = read_url_from_config();
    if (website_url != NULL && *website_url != '\0') {
        open_url(website_url);
    }
    else {
        open_url(DEFAULT_URL);
    }
    g_free(website_url);
}

int main (int   argc, char *argv[]) {
    GtkApplication *app;
    int ret;

    app = gtk_application_new (APP_ID, G_APPLICATION_DEFAULT_FLAGS);
    g_signal_connect (app, "activate", G_CALLBACK (run), NULL);

    ret =  g_application_run (G_APPLICATION (app), argc, argv);
    g_object_unref (app);

    return ret;
}
```

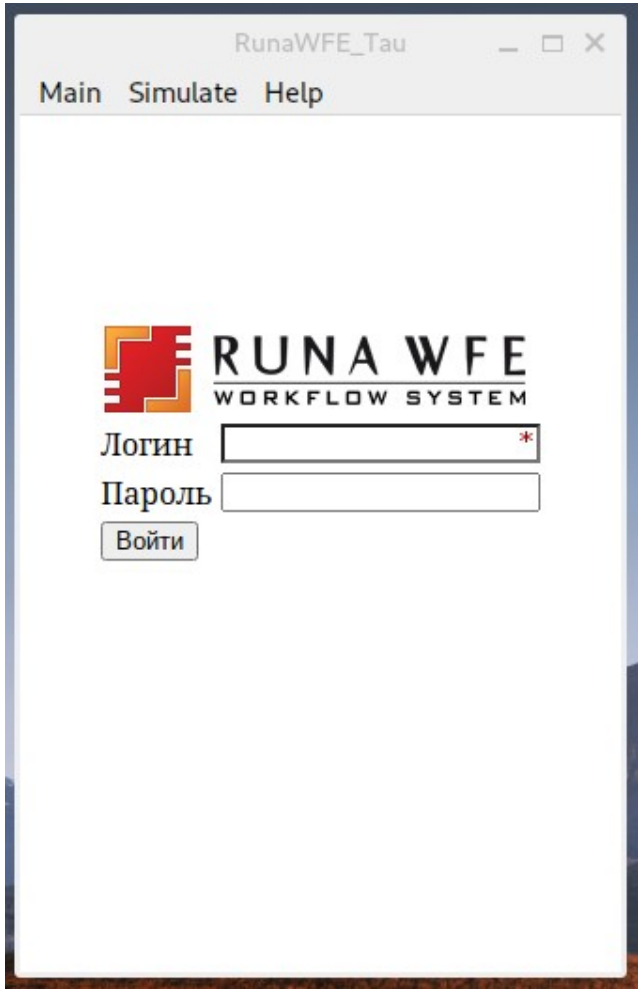
Приложение, которое открывает онлайн-версию сайта

Два варианта: на C и C++

(для сравнения)

Используются функции библиотек Glib и GTK 4

Приложение Tau-patform (фреймворк Rhodes)



Кросс-платформенный фреймворк для мобильных приложений (Ruby, JS)

https://tau-platform.com/ru/

Приложение простое: браузер WebView

Проверялась возможность разработки приложения на Rhodes

Зависимость: Qt 5

Заключение

Спасибо за внимание!

Исходный код приложений доступен на GitHub

RunaWFE Free Android:
https://github.com/processtech/runawfe-free-android-app/pull/1

RunaWFE Free ALT Mobile:
https://github.com/processtech/runawfe-free-alt-mobile/pull/1