

**Софийски университет „Св. Климент Охридски“**

**Факултет по математика и информатика**

**Курсов проект по мобилни приложения**

**SmartGarden app**

**Автор:** Христо Загоров **Ръководител:**

**Специалност:** Вградени системи ас. Христо Христов

**ОКС:** Магистър

**Фак. номер:** 25637

София, 30.06.2018 г.

Въведение

Целта на проекта е създаването на мобилно приложение за Android платформата, което да служи за следене на набор от параметри на околната среда, спомагащо отглеждането на растения в домашни условия. Параметрите, които ще визуализира приложението са влажността на въздуха, температурата и осветеността. Освен следене на климатичните условия, важен аспект при отглеждането на растения е тяхното поливане. За целта е предвиден сензор, следящ влажността на почвата, като при нужда от поливане, системата изпраща push notification към приложението и мобилното устройство.

Реализация

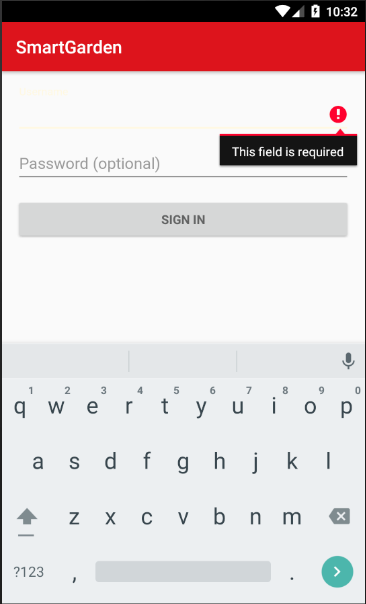
Проектът се състои от две части – сървърна и клиентска. Сървърната част се грижи за измерването на четирите величини и изпращането на данните към клиентското приложение, а клиентската (приложението) прави заявки към сървъра и визуализира получените резултати.

**Сървърна част**

Състои се от Raspberry Pi 3 микрокомпютър, ползващ LAMP стек за конфигурирането му като Apache уеб сървър с MySQL база, който събира данните от измервателните сензори за температура, влажност и осветеност, записва ги в базата, и ги изпраща под формата на JSON обект при заявка от клиента.

При достигане определен праг на влажността на почвата сървърът изпраща push notification към приложението, използвайки Firebase Cloud Messaging (наследник на deprecate-натия Google Cloud Messaging) услугата на Google, която се явява де факто стандарт за push нотификациите в Android. Firebase играе ролята на посредник между сървъра и клиентския ап, който препраща направената от сървъра заявка с информацията за известието и firebase token, уникален за всяко устройство на което е инсталирано приложението, с който се осъществява идентифицирането на приемника на известието.

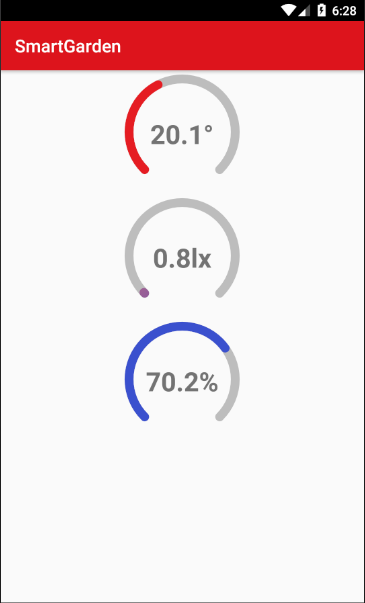
**Клиентска част**

Началният екран на приложението ще съдържа логин форма за въвеждане на username и парола. При валидно въвеждане на потребителските данни се показват стойностите на трите измерени величини – температура, влажност и осветеност. Връзката със сървъра се осъществява с get http заявка, на която сървърът връща JSON обект, съдържащ резултатите от измерванията. За създаването и изпращането на заявките се използва Volley библиотеката и Handler обект, с койтo се създава нова нишка, отговорна за изпращането на заявка през една минута. По този начин се цели извличането и обновяването на резултатите всяка минута.

Фиг.1 LoginActivity

Push notification функционалността е реализирана с помощта на FirebaseMessaging API и два background service-а – FirebaseInstanceIdService за генериране на firebase токен и MyFirebaseMessagingService с който се създава и визуализира известието с получените от Firebase данни – заглавие и текст.

Архитектура

Приложението съдържа две активитита - LoginActivity и MainActivity. LoginActivity е launcher activity, което се отваря при стартирането на приложението. Използван е стандартният Android LoginActivity template, съдържащ два елемента от тип EditView и Sign In бутон за вписване. При въвеждането на валидни име и парола се стартира AsyncTask, който проверява за съвпадение с записаните в SharedPreferences данни.

При коректно въведени данни се стартира MainActivity, съдържащо 3 TextView компонента с стойностите на получените от сървъра данни за величините, обгърнати от 3 CustomGauge елемента, които дават относителна представа доколко стойностите са в норма. Фиг.2 Екранът с резултатите

Background нишката, инстанциирана с Handler обект, праща всяка минута HTTP GET заявка към сървъра за получаване на резултатите от сензорите.

Push notifications

За работа с Firebase се интегрира firebase-messaging библиотеката в проекта. Преди това нужно в root-level build.gradle файлът да се добави google-services плъгин:

dependencies {  
        // ...  
        classpath 'com.google.gms:google-services:4.0.1' // google-services plugin }

После в add/build.gradle се слага :

// ADD THIS AT THE BOTTOM  
apply plugin: 'com.google.gms.google-services'

Следващата стъпка е прибавянето на двата сървиса в манифеста:

<service  
 android:name=".MyFirebaseMessagingService"  
 android:permission="com.google.android.c2dm.permission.SEND">  
 <intent-filter>  
 <action android:name="com.google.firebase.MESSAGING\_EVENT" />  
 </intent-filter>  
</service>  
<service android:name=".MyFirebaseInstanceIDService">  
 <intent-filter>  
 <action android:name="com.google.firebase.INSTANCE\_ID\_EVENT" />  
 </intent-filter>  
</service>

Сървисът, който наследява FirebaseInstanceIdService е отговорен за генерирането, обновяването и съхраняването на firebase токенът. Токенът е уникален за всяко устройство и се поставя в push notification заявката от сървъра. Токенът се генерира с FirebaseInstanceId.*getInstance*().getToken()

и се запазва в SharedPreferences.

Приемането и обработката на известието се извършва в MyFirebaseMessagingService, който наследява FirebaseMessaingService. Сървисът имплементира два метода – onMessageReceived, който приема като параметър полученото известие и се изпълнява **само** когато приложението е в преден фон. Всички notification съобщения, изпратени когато приложението е в на заден фон или затворено, биват получени в system tray на устройството.

При получено съобщение се инстанциира Notification.Builder, в който се добавят title и message полетата, съдържащи се в изпратеното от сървъра съобщение, които оповестяват потребителят, че растението има нужда от поливане.

notificationManager.notify(notificationId ,notificationBuilder.build());

създава известието и го изобразява на екрана на устройството.

Използвана литература

* Griffiths, Dawn and Griffiths, David (2017). Head First Android Development. A Brain-Friendly Guide. O’Reilly Media, Inc
* Android developers website: <https://developer.android.com/guide>
* Firebase documentation - https://firebase.google.com/docs/cloud-messaging/android/client