



POZNAN UNIVERSITY OF TECHNOLOGY

Patryk Dąbrowki 100584
Aleksander Kędzierski 98875
Paweł Lampe 99277
Mateusz Sikora 99615

Platforma zarządzania zdarzeniami na urządzeniach mobilnych if{y}

Bachelor's Thesis

Supervisor: dr inż. Jerzy Błaszczyński

Poznań, 2014

Spis treści

1	Wstęp	5
1.1	Opis problemu i koncepcja jego rozwiązania	5
1.2	Cel i zakres pracy	5
1.3	Omówienie pracy	5
2	Rozdziały teoretyczne	7
2.1	Konkurencyjne rozwiązania	7
2.1.1	On X	7
2.1.2	Tasker	7
3	Projekt rozwiązania	9
3.1	Definicja pojęć	9
3.2	Przypadki użycia	9
3.3	Moduły systemu.	9
4	Opis implementacji	11
4.1	Użyte technologie	11
4.1.1	Android	11
4.1.2	Apache Commons	11
4.1.3	Apache HTTP Server	11
4.1.4	Git	11
4.1.5	HTML 5	11
4.1.6	Hibernate	11
4.1.7	JSON	12
4.1.8	Java 6	12
4.1.9	JavaScript	12
4.1.10	Apache Maven	12
4.1.11	MySQL	12
4.1.12	PHP	12
4.1.13	RESTeasy	12
4.1.14	SpringFramework	12
4.1.15	Vaadin.	12

4.2	Użyte narzędzia	13
4.2.1	Apache Tomcat	13
4.2.2	Eclipse with Android developer tools.	13
4.2.3	String Tool Suite.	13
4.2.4	Emacs	13
4.2.5	Git bash for windows	13
4.2.6	Git for linux	13
4.2.7	Github.	13
4.2.8	Latex	13
4.2.9	Linux	13
4.2.10	Notepad++.	14
4.2.11	Przeglądarki internetowe	14
4.2.12	Windows.	14
4.3	Użyty sprzęt	14
4.3.1	Komputery klasy PC	14
4.3.2	LG Swift GT540 -	14
4.3.3	Media-Droid IMPERIUS EN3RGY MT7013.	14
4.3.4	Motorola Defy MB525	14
4.3.5	Sony LT18 Xperia Arc S	14
4.3.6	Samsung Galaxy Mini GT-S5570	14
4.4	Architektura klienta	15
4.4.1	Moduł obsługi recept	15
4.4.2	Moduły dostępu do systemu	15
4.5	Architektura serwera	15
4.5.1	Repozytorium recept	15
4.5.2	Serwer recept grupowych	15
4.6	Opis pakietów.	15
4.6.1	Pakiety Aplikacji	15
4.6.2	Pakiety Biblioteki	15
5	Zakończenie	17
A	Przewodnik użytkownika	19
A.1	Opis Podfunkcjonalności	19
A.1.1	Akcelerometr (YAccelerometerFeature.java)	19
A.1.2	Battery (YBatteryFeature.java)	19
A.1.3	SMS (YSMSFeature.java)	19
A.1.4	Wifi (YWifiFeature.java)	19
A.1.5	GPS (YGPSFeature.java)	19
A.1.6	Sound (YSoundFeature.java)	19
A.1.7	RawPlayer (YRawPlayerFeature.java)	19
A.1.8	Group (YGroupFeature.java)	19
A.1.9	Geocoder (YGeocoderFeature.java)	19
A.1.10	Time (YTimeFeature.java)	20
A.1.11	AudioManager (YAudioManager.java)	20
A.1.12	Text (YTextFeature.java)	20

A.1.13 Internet (YInternetFeature.java)	20
A.1.14 Calls (YCallsFeature.java)	20
A.1.15 Notification (YNotificationFeature.java)	20
Bibliografia	21

Wstęp

1.1 Opis problemu i koncepcja jego rozwiązania

Współczesne urządzenia mobilne dysponują ogromnym zbiorem możliwości. W przypadku chęci złożenia kilku możliwości w pewną usługę, trzeba stworzyć dedykowany kod. Prowadzi to do zalania rynku aplikacji mobilnych dużą ilością prostych oraz stosunkowo schematycznych implementacji.

Problem można oczywiście rozwiązać tworząc aplikację pozwalającą na preparowanie pewnych schematów używając prostego języka opisu problemu. Ujmując krócej – można napisać program pozwalający na tworzenie małych programów wykonujących konkretne akcje w kontekście zaistniałych zdarzeń. Obecnie istnieją takie rozwiązania, jednak są to produkty komercyjne o zamkniętym kodzie.

Koncepcją jaka została przyjęta w niniejszej pracy, to stworzenie otwartoźródłowej biblioteki uproszczające dostęp do podzespołów urządzenia. Celem zwiększenia atrakcyjności, postanowiono stworzyć także aplikację ukazującą jej możliwości.

1.2 Cel i zakres pracy

Celem niniejszej pracy, jest stworzenie ...

1.3 Omówienie pracy

Rozdziały teoretyczne

2.1 Konkurencyjne rozwiązania

2.1.1 On X

Aplikacja Microsoftu umożliwiającą kontrolowanie telefonu z Androidem używając kodu w JavaScriptcie. Umożliwia wysyłanie Zasad (Rules) na telefon poprzez stronę internetową. Dostęp do funkcjonalności Androida jest zapewniony przez api w postaci Wyzwalaczy (Triggers) i Akcji (Actions). Cały system jest niestety połączony z Facebookiem i wymaga posiadania tam konta. Na podstawie [1].

2.1.2 Tasker

Więcej informacji można znaleźć w książce [2].

Projekt rozwiązania

3.1 Definicja pojęć

- Podfunkcjonalność (ang. Feature) – Część biblioteki zapewniająca Receptom dostęp do pozdbioru funkcjonalności Androida.
- Zdarzenie (ang. Event) – Zmiana stanu systemu, która powoduje uruchomienie kodu Recepty.
- Recepta (ang. Recipe) – Napisany przez użytkownika fragment kodu opisujący, co ma się zdarzyć po spełnieniu pewnych warunków.
- Targowisko (ang. Market) – Aplikacja internetowa pozwalająca tworzyć i pobierać Recepty.
- Aplikacja – Aplikacja androidowa wykorzystująca bibliotekę if{Y}.
- Serwer Grup – Komputer z działającą aplikacją, która zarządza grupami użytkowników i Zdarzeniami Grupowymi.
- Zdarzenie Grupowe – Zdarzenie związane z Grupą, wysyłane lub odbierane przez Aplikację z Serwera Grup.
- Grupa – Zbiór użytkowników identyfikowalny przez nazwę zdefiniowany na Serwerze Grup.

3.2 Przypadki użycia

3.3 Moduły systemu

System składa się z biblioteki, przykładowej Aplikacji appIFY oraz aplikacji działających na serwerze - Serwera Grup oraz Targowiska. Aplikacja korzysta z biblioteki oraz komunikuje się z serwerem. Oprócz tego Serwer Grup oraz Targowisko udostępniają z poziomu przeglądarki takie funkcje jak rejestracja użytkowników czy tworzenie recept.

Opis implementacji

4.1 Użyte technologie

W tej części zaprezentowano opis technologii użytych bezpośrednio w implementacji składowych platformy.

4.1.1 Android

4.1.2 Apache Commons

4.1.3 Apache HTTP Server

Otwartoźródłowy serwer HTTP. Najpopularniejsze narzędzie tego typu na świecie. Jego wielką zaletą jest mnogość informacji na jego temat dostępnych w internecie oraz dostępność na większość znaczących systemów operacyjnych.

4.1.4 Git

Rozproszony oraz wieloplatformowy system kontroli wersji będący wolnym oprogramowaniem. Preferowane narzędzie programistów związanych z otwartym oprogramowaniem.

4.1.5 HTML 5

Język programowania służący do tworzenia współczesnych stron internetowych. Jest rozwinięciem oraz uproszczeniem języka HTML 4.

4.1.6 Hibernate

Narzędzie odwzorowań obiektowo-relacyjnych (ang. object-relation mapping, ORM) rozwijany na zasadzie wolnego oprogramowania. Umożliwia odwzorowania obiektowo-relacyjne, pamięć podręczną, leniwe (ang. Lazy loading), chciwe pobieranie oraz rozproszoną pamięć podręczną.

4.1.7 JSON

JavaScript Object Notation, lekki format danych wywodzący się z języka JavaScript(jest jego podzbiorem).

4.1.8 Java 6

4.1.9 JavaScript

4.1.10 Apache Maven

Narzędzie automatycznego budowania oprogramowania dla języka JAVA. Głównymi problemami jakie rozwiązuje Maven przy budowaniu aplikacji są: zarządzanie zależnościami, możliwość wieloma modułami, wsparcie dla testów.

4.1.11 MySQL

System zarządzania relacyjnymi bazami danych. Jest to wolne oprogramowanie szczególnie upodobane przez twórców aplikacji internetowych. Bardzo dobrze współpracuje z językami takimi jak PHP czy Java

4.1.12 PHP

Obiektowy język programowania dedykowany generowaniu stron internetowych w czasie rzeczywistym. Szczególnie użyteczny w przypadku tworzenia prototypów tudzież niewielkich projektów wymagających stosunkowo niskiego poziomu abstrakcji.

4.1.13 RESTeasy

Framework oprogramowania służący do tworzenia aplikacji rozproszonych, oparty na wzorcu architektury oprogramowania Representational State Transfer(REST).

4.1.14 SpringFramework

Framework(Szkielet) tworzenia aplikacji w języku Java a w szczególności JavaEE. Do najważniejszych funkcji Springa zalicza się wstrzykiwanie zależności (ang. dependency injection, DI) oraz programowanie aspektowe (ang. aspect-oriented programming, AOP).

4.1.15 Vaadin

Framework sieciowy służący do tworzenia aplikacji sieciowych w szczególności interfejsu użytkownika w oparciu o Google Web Toolkit (GWT) w języku JAVA.

4.2 Użyte narzędzia

4.2.1 Apache Tomcat

Kontener aplikacji sieciowych.

4.2.2 Eclipse with Android developer tools

4.2.3 String Tool Suite

Zintegrowane środowisko programistyczne oparte o Eclipse dostosowany do SpringFramework.

4.2.4 Emacs

Popularny, w pełni rozszerzalny edytor tekstowy spotykany głównie w systemach operacyjnych z rodziny Unix. Używany przez wysokiej klasy programistów oraz naukowców na całym świecie.

4.2.5 Git bash for windows

4.2.6 Git for linux

4.2.7 Github

Serwis internetowy gromadzący społeczność programistów z całego świata. Służy jako hosting dla otwartoźródłowych projektów zarządzanych za pomocą systemu Git.

4.2.8 Latex

4.2.9 Linux

Rodzina systemów operacyjnych będących wolnym oprogramowaniem oraz używających jądra Linux.

4.2.10 Notepad++

4.2.11 Przeglądarki internetowe

4.2.12 Windows

4.3 Użyty sprzęt

4.3.1 Komputery klasy PC

4.3.2 LG Swift GT540 -

Procesor: Qualcomm MSM7227 600 MHz Pamięć RAM: 256 MB System operacyjny: Android 4.0.1 (Cyanogen mod)

4.3.3 Media-Droid IMPERIUS EN3RGY MT7013

Procesor: dwurdzeniowy, 1GHz ARM7 MTK6577 Pamięć RAM: 256 MB System operacyjny: Android 4.1.2

4.3.4 Motorola Defy MB525

Procesor: TI OMAP3610 800 MHz Pamięć RAM: 512 MB System operacyjny: Android 4.3.1 (Cyanogen mod)

4.3.5 Sony LT18 Xperia Arc S

Procesor: Qualcomm MSM8255T 1,40 GHz Pamięć RAM: 512 MB System operacyjny: Android 4.0.4

4.3.6 Samsung Galaxy Mini GT-S5570

Procesor: Qualcomm MSM7227 600 MHz Pamięć RAM: 384 MB System operacyjny: Android 2.2

4.4 Architektura klienta

4.4.1 Moduł obsługi recept

4.4.2 Moduły dostępu do systemu

4.5 Architektura serwera

4.5.1 Repozytorium recept

4.5.2 Serwer recept grupowych

4.6 Opis pakietów

4.6.1 Pakiety Aplikacji

pl.poznan.put.cs.ify.app - główny pakiet Aplikacji. pl.poznan.put.cs.ify.jars - pakiet odpowiedzialny za zarządzanie plikami .jar zawierającymi recepty pobrane z Targowiska. pl.poznan.put.cs.ify.core - pakiet odpowiedzialny za zarządzanie dostępnymi i aktywowanymi Receptami. pl.poznan.put.cs.ify.appify.receipts - pakiet zawierający Recepty wbudowane w Aplikację. pl.poznan.put.cs.ify.app.ui - pakiet zawierający kontrolki interfejsu użytkownika. pl.poznan.put.cs.ify.app.ui.params - pakiet zawierający kontrolki interfejsu użytkownika wykorzystywane do wprowadzania parametrów przy inicjalizacji Recepty. pl.poznan.put.cs.ify.app.market - pakiet odpowiedzialny za pobieranie danych z Targowiska i wyświetlanie ich. pl.poznan.put.cs.ify.app.fragments - pakiet zawierający widoki ekranów aplikacji.

4.6.2 Pakiety Biblioteki

pl.poznan.put.cs.ify.api - pakiet główny Biblioteki. pl.poznan.put.cs.ify.api.exceptions - pakiet zawierający wyjątki, które mogą być rzucane przez metody z Biblioteki. pl.poznan.put.cs.ify.api.features - pakiet zawierający Podfunkcjonalności i Zdarzenia. pl.poznan.put.cs.ify.api.group - pakiet odpowiedzialny za obsługę Recept Grupowych. pl.poznan.put.cs.ify.api.log - pakiet odpowiedzialny za obsługę logowania i domyślny widok logów. pl.poznan.put.cs.ify.api.params - pakiet zawierający typy parametrów wykorzystywanych przez Recepty. pl.poznan.put.cs.ify.api.security - pakiet odpowiedzialny za moduł uprawnień Biblioteki. pl.poznan.put.cs.ify.api.types - pakiet zawierający typy danych wykorzystywanych przez Bibliotekę.

Zakończenie

Przewodnik użytkownika

A.1 Opis Podfunkcjonalności

A.1.1 Akcelerometr (YAccelerometerFeature.java)

Umożliwia reagowanie na odczyty akcelerometru wbudowanego w urządzenie.

A.1.2 Battery (YBatteryFeature.java)

Umożliwia reagowanie na zmiany poziomu baterii urządzenia.

A.1.3 SMS (YSMSFeature.java)

Umożliwia wysyłanie wiadomości SMS oraz reagowanie na wiadomości przychodzące.

A.1.4 Wifi (YWifiFeature.java)

Umożliwia włączanie i wyłączanie modułu WiFi urządzenia.

A.1.5 GPS (YGPSFeature.java)

Umożliwia śledzenie pozycji urządzenia za pomocą modułu GPS.

A.1.6 Sound (YSoundFeature.java)

A.1.7 RawPlayer (YRawPlayerFeature.java)

A.1.8 Group (YGroupFeature.java)

A.1.9 Geocoder (YGeocoderFeature.java)

Umożliwia pobranie adresu związanego z podaną długością i szerokością geograficzną.

A.1.10 Time (YTimeFeature.java)**A.1.11 AudioManager (YAudioManager.java)****A.1.12 Text (YTextFeature.java)****A.1.13 Internet (YInternetFeature.java)**

Umożliwia wysyłanie i pobieranie danych z podanego adresu.

A.1.14 Calls (YCallsFeature.java)

Umożliwia reagowanie na połączenia przychodzące i inicjowanie połączeń wychodzących.

A.1.15 Notification (YNotificationFeature.java)

Umożliwia wyświetlanie powiadomień w interfejsie graficznym urządzenia.

Bibliografia

- [1] Projekt on{X} <http://www.onx.ms/#!findOutMorePage>. Ostatnio odwiedzone 6/02/13.
- [2] A. Tanenbaum. *Operating Systems Design and Implementation*. Prentice Hall, 2006.
- [3] C. Walls. *Spring in action, 3rd edition*. Manning Publication Co, 2011.
- [4] Vaadin <https://vaadin.com/book/vaadin6/-/page/preface.html>
- [5] E. Gamma. *Design Patterns, First edition*. Person Education, Inc, 1995.