NTNU

Norwegian University of Science and Technology

Assignment 1

UI, THE WEB & PERSISTENT STORAGE

Gent Rexha | ID: 998903 | Mobile Development Theory | 19.09.2016

Table of Contents

Description of the application 2
Behind the scenes 2
Use of stored data when restarting from shutdown 3
Difference between native Apps and web Apps 3
What's next? (Explanation of how the app could be extended) 3
Personal experience: 4
References: 4

Description of the application

The WhereAmI&WhatsTheWeather application is a simple Android application, which does exactly what it says. It allows the user to pinpoint his location and shows various information for that location, such us: temperature, altitude and the current place (user location) using the smart phones location sensor and the Google Maps API [1][2].

The application consists of 4 different activities, which interacts with each-other by passing variables to next activity (interface). The location sensor data is sent to the OpenWeather API [3] through the web and based on latitude and longitude returns the current weather for that position, the same process is applied for the altitude variable, which the application gets from the Google Maps Elevation API. All of the information is stored in the device which is shown to the user in a simple ListView.

Behind the scenes

The application consists of a main activity and 3 extra activities, which all are shown on button clicks. The main activity has a simple TextView widget which explains what the application does and with two buttons. When this activity is created the application checks if the user has location settings enabled and prompts the user to enable it [4], if that's the case and if the application can use the location settings of the device, since the app has been developed for Android Marshmallow application access isn't asked about before installation but every time the app needs a specific service it has to ask for that service and the user can allow or disallow it [5]. All of this could've been shown on the map activity and reduce the number of activity switches. After the map fragment is ready to show the map it enables the user to show his location and connects to the Google API through the Google API client. At first step, the app gets the users last location and moves the view there while it starts requesting location updates which are done in an interval of 10 seconds through the FusedLocationAPI [6], although if there is no internet access the app tries to pinpoint the user through the GPS sensor of the device which might take some time. If the current location is different from its last location it will move the view to current location and re-initialize the passing variables for the information activity. In case the user pauses the app and the app is still connected location updates are paused. If the user requests more information for his current location, altitude and longitude are passed to a bundle [7] for the next activity, for which the user has to be connected to the internet [8], in which task based on those parameters, two different asynchronous task are started [9]. One for retrieving the weather information and one for the current altitude, since the altitude taken from the location sensor wasn't really working. Both queries are sent through the web and are returned in JavaScript Object Notation (JSON) format which then are sent to a JSON object which on the respective PostExecute method is parsed for its weather and altitude [10][11][12].

USE OF STORED DATA WHEN RESTARTING FROM SHUTDOWN

Each time a user checks for more information for his current location after the new activity fetches the weather and elevation through the web and API services, at the same time the application inserts that information and the current time in a local database which is created when the activity is created. This database and therefore that information will stay on the device independent if the app is running or not. This information is shown in a simple ListView in the history activity where a query is attached to a cursor and then attached to the ListView through a SimpleCursorAdapter object [13].

Difference between native Apps and web Apps

The native app, the app one installs from app store, which is platform specific but can take advantages from phone features, making user racing game experience more fun by rotating his device and simulating a steering wheel or just applying a filter to a photo to make user look more tanned. On the other hand, there are web apps, which directly aren't apps just websites, but which are made to look and behave like apps all through the beauty of HTML5.

Web apps, in my opinion, are historical artifacts since they were based on the thin client and heavy CPU server architecture which at that time was meant to make mobile phones and PDA-s thinner and lighter for everyday usage since they were so big and heavy back then. But the technology evolved and processors got faster, More's Law [14], and the mobile phones got lighter and the need for a heavy CPU on the server side faded since a lot of that computing power came into our hands. The best example for this is Snapchat, looking at all the different face recognition filters it offers and how everything is processed in real-time by the phone processor, so we get to see the dog nose and ears on our face while we put our tongue out. But why are web apps still so popular? Mostly because they're so universal, if a company wants to show its presence on the mobile market it can either pay to develop two different applications for iOS and Android which basically have the same function, or just pay for one web app which then can be sent to both app stores without having to spend a big amount of money on developing.

The same could've been applied to our application since it doesn't make use of any other phone features except the location sensor which also can be accessed through the web and therefore reach a bigger audience.

What's next? (Explanation of how the app could be extended)

First of all, the application needs to be developed for a later android version other than Android Marshmallow, starting from Android KitKat would be a good idea since there is a

bigger market margin for the application to be popular [15]. The application needs to check on other settings which could disable the applications work process i.e. no internet connection, wrong location service settings, not allowing to use location service in which case the app crashes and implement workarounds in case the user decides to not allow access to certain features. One future extension to this application could be to change it into health/lifestyle application which would help the user in everyday activities like walking, running, etc. It could notify the user in case there is rainy weather forecasted for his location. More detailed weather reports and a 2-7-day forecast for his current location would also be a good idea.

Personal experience:

Although coding seemed hard at first since I didn't have any prior experience in mobile development, after you got into it, it got rather easy and the difficult part became in debugging and fixing problems. Especially trying to run the application in an Android Marshmallow VM with Google Play services since there isn't any support for this version, the only solution for this problem was to install the service package for Android Lollipop and then update it through another package, Because of that most of the testing was done on my personal Samsung Galaxy S7. I learned a lot about permissions and how to handle them, fragments, JSON and JSON parsing, OpenSQLite, API handling and asynchronous tasks.

References:

[1] Google Maps API, https://developers.google.com/maps/documentation/android-api/map
[2] Location Data , https://developers.google.com/maps/documentation/android-api/location
[3] OpenWeather API, http://openweathermap.org/current
[4] How to check if Location Services are enabled?, http://stackoverflow.com/a/27008913/3841083
[5] Requesting Permissions at Run Time, https://developer.android.com/training/permissions/requesting.html
[6] Getting the Last Known Location, https://developer.android.com/training/location/retrieve-current.html
[7] How to pass a value from one Activity to another in Android?, http://stackoverflow.com/questions/3510649/how-to-pass-a-value-from-one-activity-to-another-in-android
[8] How to check internet access on Android? InetAddress never times out, http://stackoverflow.com/a/4239019/3841083
[9] What is the most efficient way on Android to call HTTP Web API calls that return a JSON response?, http://stackoverflow.com/questions/19050294/what-is-the-most-efficient-way-on-android-to-call-http-web-api-calls-that-return

[10] How to use a web API from your Android app, http://www.androidauthority.com/use-remote-web-api-within-android-app-617869/
[11] Google Maps Elevation API, https://developers.google.com/maps/documentation/elevation/intro
[12] Android - JSON Parser Tutorial, http://www.tutorialspoint.com/android/android_json_parser.htm
[13] How to use SQLite to store data for your Android app, http://www.androidauthority.com/use-sqlite-store-data-app-599743/
[14] Moore's Law, https://en.wikipedia.org/wiki/Moore%27s_law
[15] Platform versions, https://developer.android.com/about/dashboards/index.html