NATIONAL AND KAPODISTRIAN UNIVERSITY OF ATHENS

SCHOOL OF SCIENCE
DEPARTMENT OF INFORMATICS AND TELECOMMUNICATION

BSc THESIS

Thesis Title

Dimitrios G. Soumis

Supervisor:	**Manolis Koubarakis, Professor**
Co-supervisor:	**George Stamoulis, Ph.D.** **Candidate UoA**

ATHENS

JANUARY 2020

ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Τίτλος εργασίας

Δημήτριος Γ. Σουμής

Επιβλέπων: **Μανόλης Κουμπαράκης, Καθηγητής**
Συνεπιβλέπων: **Γιώργος Σταμούλης, Υποψήφιος Διδάκτωρ ΕΚΠΑ**

ΑΘΗΝΑ

ΙΑΝΟΥΑΡΙΟΣ 2020

BSc THESIS

HotelPlus: Discovering Hotels and Areas of Interest using Linked Data

Dimitrios G. Soumis
S.N.: 1115201600157

SUPERVISOR:	**Manolis Koubarakis, Professor**
CO-SUPERVISOR:	**George Stamoulis, Ph.D. Candidate UoA**

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

HotelPlus: Ανακαλύπτοντας ξενοδοχεία και σημεία ενδιαφέροντος χρησιμοποιώντας Συνδεδεμένα Δεδομένα

Δημήτριος Γ. Σουμής
Α.Μ.: 1115201600157

ΕΠΙΒΛΕΠΩΝ:	**Μανόλης Κουμπαράκης, Καθηγητής**
ΣΥΝΕΠΙΒΛΕΠΩΝ:	**Γιώργος Σταμούλης, Υποψήφιος Διδάκτωρ ΕΚΠΑ**

ABSTRACT

In the last years there have been many attempts to develop applications that can offer a wealth of information in easily-understandable user interfaces, without the demand for knowledge of querying databases. This wealth of information is well acquired with the use of Linked Data.

The term Linked Data refers to a set of best practices for publishing and interlinking structured data on the Web. With Linked Data, a very pragmatic approach towards reaching Web's full potential, thus the Web to become a place where data can be shared, processed, and understood by automated tools as well as by people, has gained some traction recently.

The purpose of the project was to familiarize with Linked Data while developing an android app which makes use of databases DBpedia and OpenStreetMap, that host Linked Data, and provide results in a very simple way. This app presents the hotels of a user-selected area as well as points of interest (e.g museums, churches, parks) around a user-selected hotel.

To sum up, the application presented in this thesis, offers plenty information through simple tasks without requiring experience to query Linked Data knowledge bases.

The code that extends the system mentioned above can be found at the following link: https://github.com/dsoumis/HotelPlus

SUBJECT AREA: Android App Development with Linked Data

KEYWORDS: linked data, databases, user interface, knowledge graph, geographical information

ΠΕΡΙΛΗΨΗ

Τα τελευταία χρόνια έχουν γίνει πολλές προσπάθειες ανάπτυξης εφαρμογών που μπορούν να προσφέρουν πληθώρα πληροφοριών σε εύκολα κατανοητές διεπαφές χρηστών, χωρίς την απαίτηση γνώσεων για την αναζήτηση σε βάσεις δεδομένων. Αυτός ο πλούτος των πληροφοριών αποκτάται με τη χρήση των Συνδεδεμένων Δεδομένων-Linked Data.

Ο όρος Συνδεδεμένα Δεδομένα αναφέρεται σε ένα σύνολο βέλτιστων πρακτικών για τη δημοσίευση και τη διασύνδεση δομημένων δεδομένων στον Ιστό. Με τα Συνδεδεμένα Δεδομένα, μια πολύ ρεαλιστική προσέγγιση για την επίτευξη του πλήρους δυναμικού του Ιστού, δηλαδή ο Ιστός να γίνει ένας τόπος όπου τα δεδομένα μπορούν να μοιράζονται, να επεξεργάζονται και να κατανοούνται από τα αυτοματοποιημένα εργαλεία καθώς και από τους ανθρώπους, έχει λάβει πρόσφατα έδαφος.

Ο σκοπός του έργου ήταν η εξοικείωση με τα Linked Data, μέσω της ανάπτυξης μίας εφαρμογής Android που χρησιμοποιεί τις βάσεις δεδομένων DBpedia και OpenStreetMap, που φιλοξενούν Linked Data, και παρέχει αποτελέσματα με πολύ απλό τρόπο. Αυτή η εφαρμογή παρουσιάζει τα ξενοδοχεία μιας περιοχής που έχει επιλέξει κάποιος χρήστης καθώς και σημεία ενδιαφέροντος (π.χ. μουσεία, εκκλησίες, πάρκα) γύρω από ένα ξενοδοχείο που έχει επιλεγεί από τον χρήστη.

Συνοψίζοντας, η εφαρμογή που παρουσιάζεται σε αυτή την πτυχιακή εργασία, προσφέρει πολλές πληροφορίες μέσω απλών ενεργειών χωρίς να απαιτεί εξοικείωση με τις βάσεις γνώσεων των Linked Data.

Ο κώδικας που επεκτείνει το σύστημα που αναφέρθηκε βρίσκεται στον ακόλουθο σύνδεσμο: https://github.com/dsoumis/HotelPlus

ΘΕΜΑΤΙΚΗ ΠΕΡΙΟΧΗ: Ανάπτυξη Εφαρμογής Android με Linked Data

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: συνδεδεμένα δεδομένα, βάσεις δεδομένων, διεπαφή χρήστη, γράφος γνώσης, πληροφορίες γεωγραφικών τμημάτων

ACKNOWLEDGMENTS

I would like to acknowledge the support provided by my family and Giouli Pavlou during the preparation of this project.

I would also like to offer my special thanks to George Stamoulis for the advice given and his great help which contributed in completing the project.

Lastly, I am particularly grateful for the chance given by Professor Manolis Koubarakis to make my first steps in the android development world and build something stable with many future possible expansions.

CONTENTS

LIST OF FIGURES

LIST OF IMAGES

PREFACE

This thesis is a final work as partial fulfillment for the degree of Bachelor of Science in Faculty of Informatics and Telecommunications, National and Kapodistrian University of Athens.

This basis for this project originally stemmed from my passion for developing fully functional apps and my enthusiasm for Android. As the world moves further into the digital age, smartphones are becoming the main access points of media and information both for advanced and novice users. Thus, I felt the urge to create an android app which offers much information in little time through a user-friendly interface.

May this thesis be useful for readers and for future research about similar topic or any other related field.

1. INTRODUCTION

As the world moves further into the digital age, smartphones are becoming the main access points of media and information both for advanced and novice users. People need rapid and plenty results from simple searches in the shortest possible time by using quick access systems without having to boot up a computer in order to address their needs.

This thesis was completed by taking into account the needs referenced above and consists of an app which offers information for hotels and points of interest in the world.

It is based on a combination of public databases and technologies to make very specific information easily available to people who do not have a programming background and therefore are not able to use a querying language to retrieve results from a database.

1.1 Contributions of this thesis

To begin with, the application developed in this thesis provides information for hotels and for points of interest in a user-selected geographical region. Their position on an interactive map is also provided.

Secondly, there is the ability for one to get directions in order to approach a hotel or a point of interest from one's current or designated position.

Thirdly, users are capable of personalizing their search by using abbreviations in their searches or selecting which sites will show up on the map. For example, one can select to see churches and museums but no parks while searching in a geographical region.

Additionally, search fields are protected from typographical errors and abbreviations that users may write. For example, the search will recover and provide feasible solutions in case of "brlin" instead of "Berlin, Germany" or "gr" instead of "Greece".

Furthermore, the application developed in this thesis is lightweight with low demand of device's resources to function properly because while developing it, optimal and with lowest possible complexity techniques have been taken into consideration.

Moreover, by taking into consideration the cost of internet data consumption as well as the value of user's time, the application saves fetched data in local database in order to limit (at maximum level) internet data consumption and save a lot of time.

Last but not least, techniques used while developing this application are compatible with older Android versions so users with older android devices are capable of experiencing this app without problems and limitations.

2. BACKGROUND AND RELATED WORK

In this chapter the reader is provided with background information of technologies used in this thesis followed by an explanation of how they were used.

2.1 Linked Data, RDF, SPARQL

One of the biggest challenges in the area of intelligent information management is the exploitation of the Web as a platform for data and information integration as well as for search and querying. Just as we publish unstructured textual information on the Web as HTML pages and search such information by using keyword-based search engines, we are already able to easily publish structured information, reliably interlink this information with other data published on the Web and search the resulting data space by using more expressive querying beyond simple keyword searches. The Linked Data paradigm has evolved as a powerful enabler for the transition of the current document-oriented Web into a Web of interlinked Data and, ultimately, into the Semantic Web. The term Linked Data refers to a set of best practices for publishing and connecting structured data on the Web. These best practices have been adopted by an increasing number of data providers over the past three years, leading to the creation of a global data space that contains many billions of assertions – the Web of Linked Data.[3]

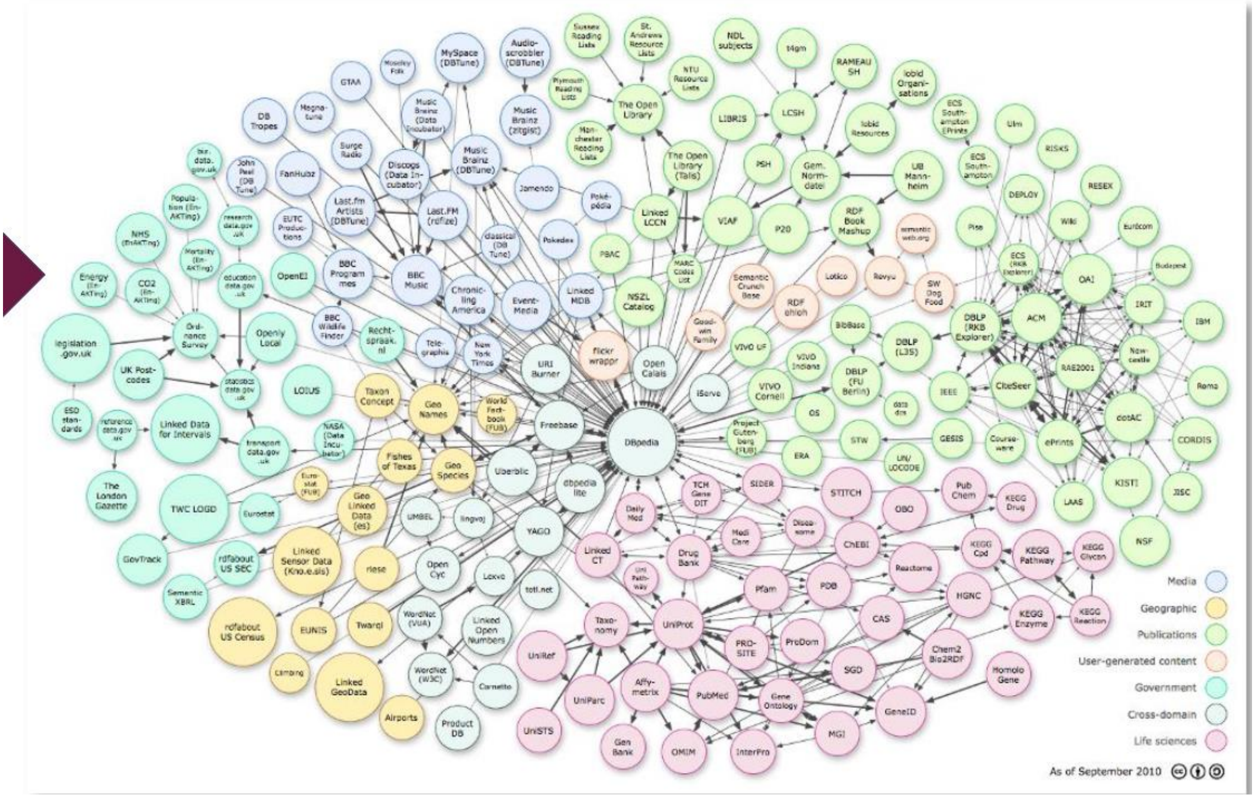


Figure 1: Linked Data knowledge bases and their interlinks available on the web

The four design principles of Linked Data:

1.Use Uniform Resource Identifiers (URIs) as names for things.

2.Use HTTP URIs so that people can look up those names.

3.When someone looks up a URI, provide useful information, using the standards (RDF*, SPARQL).

4.Include links to other URIs so that they can discover more things.

The Resource Description Framework (RDF) is a data model for representing information (especially metadata) about resources in the Web. RDF can also be used to represent information about things that can be identified on the Web, even when they cannot be directly retrieved on the Web (e.g., a book or a person). RDF is intended for situations in which information about Web resources needs to be processed by applications, rather than being only displayed to people. [4]

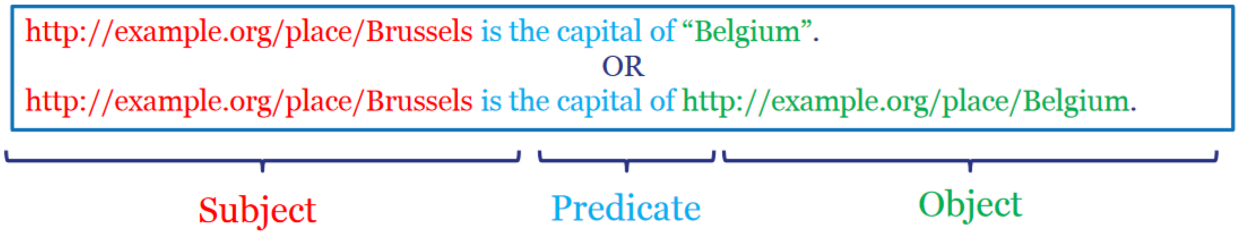
A Uniform Resource Identifier (URI) is a compact sequence of characters that identifies an abstract or physical resource.[5] Unlike URLs, URIs are not limited to identifying things that have network locations, or use other computer access mechanisms. A number of different URI schemes (URI forms) have been already been developed, and are being used, for various purposes for example http.[4]

RDF breaks every piece of information down in triples:

•Subject –a resource, which may be identified with a URI.

•Predicate –a URI-identified reused specification of the relationship.

•Object –a resource or literal to which the subject is related.[5]



http://example.org/place/Brussels is the capital of “Belgium”.

OR

http://example.org/place/Brussels is the capital of http://example.org/place/Belgium.

Figure 2: RDF example.

SPARQL is a standardized language for querying RDF data. SPARQL can be used to express queries across diverse data sources, whether the data is stored natively as RDF or viewed as RDF via middleware. SPARQL contains capabilities for querying required and optional graph patterns along with their conjunctions and disjunctions. SPARQL also supports extensible value testing and constraining queries by source RDF graph. The results of SPARQL queries can be results sets or RDF graphs.[6]

A SPARQL query comprises, in order:

- *Prefix declarations*, for abbreviating URIs
- *Dataset definition*, stating what RDF graph(s) are being queried
- A *result clause*, identifying what information to return from the query
- The *query pattern*, specifying what to query for in the underlying dataset
- *Query modifiers*, slicing, ordering, and otherwise rearranging query results

```
# prefix declarations
PREFIX foo: <http://example.com/resources/>
...
# dataset definition
FROM ...
# result clause
SELECT ...
# query pattern
WHERE {
    ...
}
# query modifiers
ORDER BY ...
```

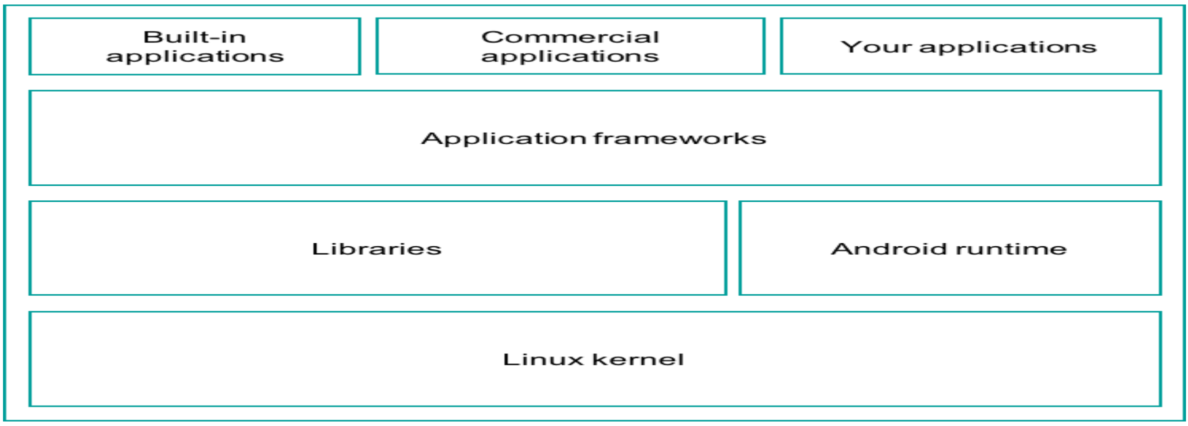
[7]

2.2 Android Development

Android is a mobile operating system based on a modified version of the Linux kernel and other open source software, designed primarily for touchscreen mobile devices such as smartphones and tablets. Android is developed by a consortium of developers known as the Open Handset Alliance, with the main contributor and commercial marketer being Google [8]. Looking at Android's breadth of capabilities, it would be easy to confuse it with a desktop operating system. Android is a layered environment, one that is built upon a foundation of the Linux kernel and includes rich functionality. The user interface subsystem includes everything you would expect from a mature operating system environment including windows, views, and widgets for displaying common elements like edit boxes, lists, or drop-down lists. The browser is both capable for general web browsing and available for embedding directly into your own application.[9]

Figure 1 displays a simplified view of the Android software layers:

- Applications: Built-in applications, such as phone, contacts, browser, and more. The specific applications vary by Android version and manufacturer. Commercial applications from marketplaces, such as Google Play, Amazon, and more. Side-loaded applications, including the ones you will build. You install these via USB cable or by downloading their .apk file from a server.
- Application frameworks, such as telephony manager, location manager, notification manager, content providers, windowing, resource manager, and more.
- Libraries, such as graphics libraries, media libraries, database libraries, sensors, and so on.
- The Android runtime is responsible for executing and managing applications as they run.
- Linux Kernel, including power, file system, drivers, process management, and more.



[9]

Figure 3: Android software layers

An important aspect of the Android application environment is that Android applications have historically been written in the Java™ programming language. However, one can also write them in a relatively new programming language from Google called Kotlin. This thesis focuses exclusively on Java.

There are some fundamentals to understand when starting out with Android development.

An Android application consists of one or more of the following four classifications:

- Activities: An application that has a visible user interface is implemented via an activity. When you select an application from the Home screen or application launcher, an activity is started.
- Services: You can use a service for any application that needs to persist for a long time, such as a network monitor or update-checking application.
- Content providers: The easiest way to think about content providers is to view them as a database server. A content provider's job is to manage access to persisted data, such as the contacts on a phone. If your application is very simple, you might not necessarily create a content provider, however if you are building a larger application or one which makes data available to multiple activities and/or applications, a content provider is the proscribed means of accessing your data.
- Broadcast receivers: You can launch an Android application to process a specific element of data or respond to an event, such as receiving a text message.

An Android application is deployed to a device along with a file named AndroidManifest.xml. This file is required for every Android application and is the guideline for the operating system in order to interact properly with one's application.

For example, if an application requires the geographical position of the user, it needs permission from the user to acquire it. This and other kinds of permission are explicitly stated in AndroidManifest.xml.

The easiest way to get started with developing Android applications is to use the Android Studio application suite. (https://developer.android.com/studio)

2.3 Datasets

In this thesis, a variety of datasets is used in order to be completed. These datasets are DBpedia, OpenStreetMap, Bing Maps, Google Maps and are analyzed in the following paragraphs.

DBpedia is a crowd-sourced community effort to extract structured content from the information created in various Wikimedia projects. This structured information resembles an open knowledge graph which is available for everyone on the Web. A knowledge graph is a special kind of database which stores knowledge in a machine-readable form and provides a means for information to be collected, organized, shared, searched and utilized.[10]

DBpedia data is served as Linked Data, which is revolutionizing the way applications interact with the Web. One can navigate this Web of facts with standard Web browsers, automated crawlers or pose complex queries with SQL-like query languages e.g. SPARQL through its SPARQL endpoint.[10]

A SPARQL Endpoint is a Point of Presence on an HTTP network that’s capable of receiving and processing SPARQL Protocol requests. It is identified by a URL commonly referred to as a SPARQL Endpoint URL. The SPARQL Protocol is an HTTP-based protocol for performing SPARQL operations against data via SPARQL Endpoints. Subject to the kind of operation being performed, HTTP payloads are dispatched using GET, POST, or PATCH methods.

A SPARQL endpoint offer several unique benefits via an API:

- Declarative data interaction (manipulation and definition) may be integrated via HTTP — targeting data represented as fine-grained RDF sentence/statement collections
- HTTP document URLs bring extensive flexibility to data queries, i.e., each component of a URL is a slot for parameterized alteration of the specifics of a query that extends from its target endpoint to the nature of the query solution
- A wide variety of content types are supported for query solution documents — HTML, JSON, CSV, RDF-Turtle, RDF-N-Triples, RDF-XML, and others
- Content types of all query solution documents are negotiable — courtesy of HTTP content-negotiation
- Endpoints may be accessed with any HTTP-compliant user agent or service [11]

OpenStreetMap primarily is a database of worldwide geographic data. The data is geographic base data, e.g. streets, roads, railways, shops, hotels etc. In general, everything that is observable on the ground can be added to OpenStreetMap. Personal data is never added to the database: It is forbidden in OpenStreetMap to copy names from bell plates to the database.

The Overpass API keeps a copy of the main database and provides it for querying.

Overpass API is designed to answer queries from other software over the internet. [12]

OpenStreetMap foremost contains three kinds of data:

- Geometries, more precisely coordinates and references to the coordinates, locate the objects on Earth's surface.
- Short bits of text give each object a semantical meaning.
- Meta data facilitates to attribute the sources to the data.

All selection criteria of the query language deal with properties of these data structures. [13]

The Bing Maps-Geospatial Endpoint Service is a REST service that provides information about Geospatial Platform services for the language and geographical region one specifies. The service information includes available service endpoints and language support for these endpoints.

It is important to note that the endpoints returned in the response are not typically URLs that you can execute. They give the base URL to which you can add parameters, such as a quad key for an image, route waypoints or an address to geocode. [14]

Google Maps is a Web-based service that provides detailed information about geographical regions and sites around the world.

Google Maps offers several services as part of the larger Web application, as follows.

- A route planner offers directions for drivers, bikers, walkers, and users of public transportation who want to take a trip from one specific location to another.
- The Google Maps application program interface (API) makes it possible for Website administrators to embed Google Maps into a proprietary site such as a real estate guide or community service page.
- Google Maps for Mobile offers a location service for motorists that utilizes the Global Positioning System (GPS) location of the mobile device (if available) along with data from wireless and cellular networks.
- Google Street View enables users to view and navigate through horizontal and vertical panoramic street level images of various cities around the world. [15]

2.4 Contribution of background to thesis

In this thesis, Android Studio was used to create and test the application (written in Java), SPARQL endpoint of DBpedia and Overpass API of OpenStreetMap were used to gather data/information, Geospatial Endpoint Service of Bing Maps was used to fetch correct geographical coordinates and Google Maps were used to present the results.

3. HOTELPLUS

HotelPlus is an android application created in Android Studio, written in Java and tested on 8.0 Oreo (API level 26) android version which focuses on providing hotels and points of interest in a user defined region while swapping screens through activities with the use of Intents.

The application consists of 4 activities which tamper with an SQLite database and device’s file system. Each activity has its own Java file and its corresponding XML file that depicts the user interface. The application is based on querying linked data from knowledge databases.

Additionally, many technologies have been used for the final product to function properly. These are Overpass API which provides data fetched from OpenStreetMap database, DBpedia’s SPARQL endpoint which provides data fetched from DBpedia, Bing Maps-Geospatial Endpoint Service for geographical locations querying and Google Maps API in order to display results.

DBpedia’s SPARQL endpoint was accessed with an http request consisting of the SPARQL query like “select distinct ?Concept where {[] a ?Concept} LIMIT 100”.

Map of Google Maps API is customized with the use of a json file (/res/raw/style_json.json) which was created at https://mapstyle.withgoogle.com/. Custom markers depict icons downloaded from https://icons8.com/.

3.1 Basic Architecture

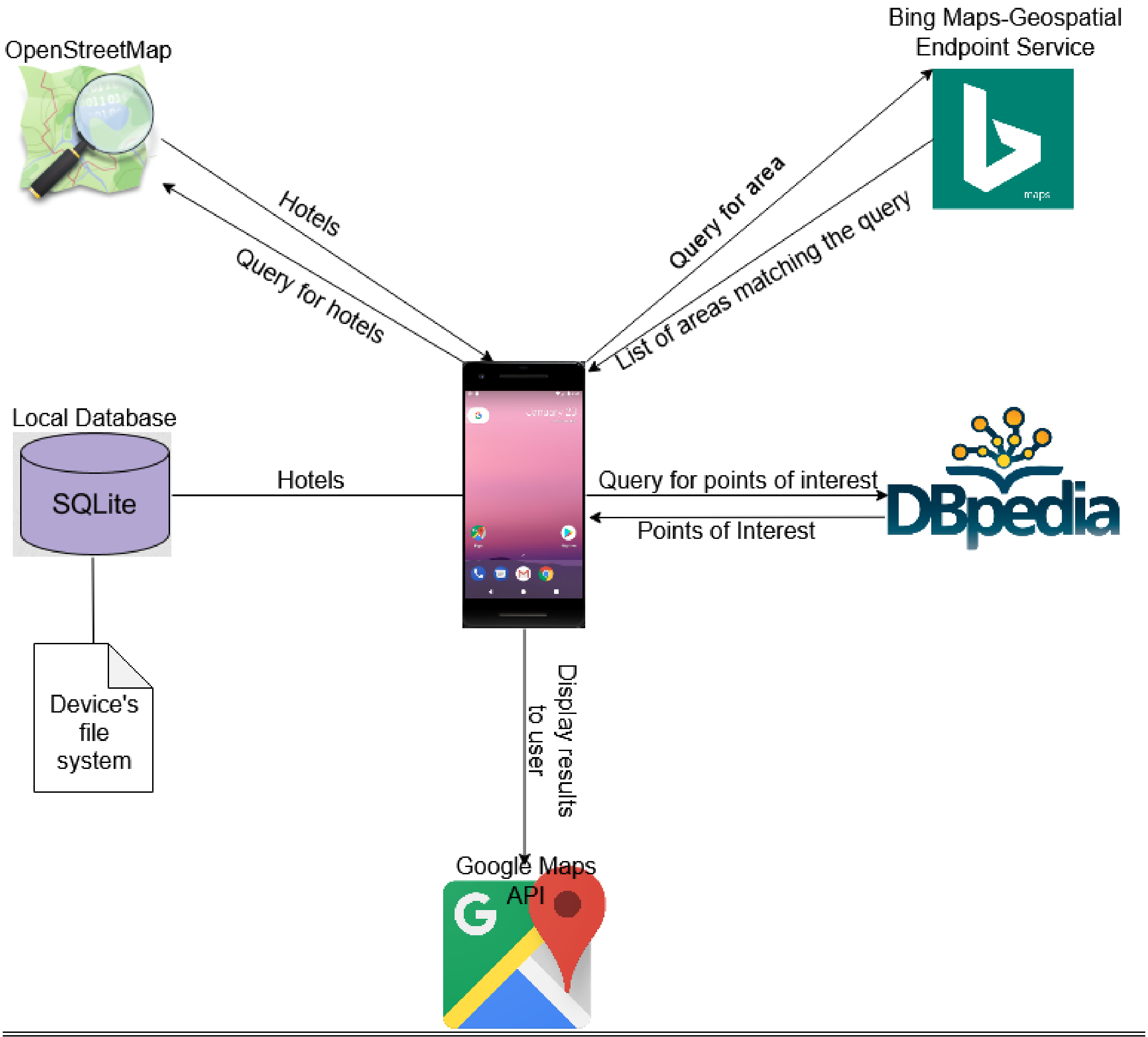


Figure 4: HotePlus Basic Architecture

HotelPlus sends an HTTP request to OpenStreetMap's Overpass API in order to fetch hotels of specific areas. The hotels from OpenStreetMap are saved into an SQLite local database which also tampers with the device's file system. Then_HotePlus sends HTTP requests to Bing Maps Geospatial Endpoint Service to get areas/locations information. Additionally, the application sends HTTP requests to DBpedia to get information about points of interest. Lastly, HotePlus uses the Google Maps API in order to provide user with a friendly and easily understandable interface which depicts user-query's results.

3.2 Fast results, less internet data consumption and Overpass API

To begin with, Overpass API was used to gather the information of all hotels in a specific geographical bounding box which represents a country. This was done by an http “GET” request to this link http://overpass-api.de/api/interpreter?data=[out:json];(node[tourism=hotel] (+bbox+););out body; where +bbox+ is a geographical bounding box like "47.2701114,5.8663153,55.099161,15.0419319" . The process in order to get results from this request is time consuming and costs internet data as the response is almost 6MB.

The main idea while developing this app was to build a stable and fast application which simultaneously requires the least possible resources to function properly. In order to achieve this, device’s file system is used. More specifically, the response from the Overpass API request is saved so the repetition of fetching data is avoided. Technically, when the response is delivered from Overpass API request, a file named as +bbox+ is created in device’s file system and its content consists of the response which is converted to a (huge) string. In addition to this, there exists a database with a table consisting of 5 columns, the 4 coordinates of the bounding box and the name of the file which holds the hotels’ information. So, once the file named as +bbox+ is created, there is also a new entry in the database for later access. To sum up, a request to the Overpass API is done only the first time a user runs the app and there is no need to consume internet data again for the same request. It is also noteworthy that the complexity of searching the (saved) response/file in a local database has O(n) complexity at worst case scenario where n is the number of records in the table, a lot faster than the http request.

In the code, the ability to save prefetched data delivered with application’s package is also implemented so there is no need for http requests to Overpass API not even once. The use of an already created database with entries is also implemented and delivered with application’s package. These parts of code are currently commented out for experimental purposes with the http request, but one can use them by just uncommenting lines 126-131 in MainActivity.java file and save a lot of time and internet data.

3.3 Invulnerable search and Bing Maps-Geospatial Endpoint Service

Another idea which held by, while developing this app, was to maintain a search which didn’t succumb in most common typing errors. Users nowadays are used to typing words very fast without regard for the result or are used to typing abbreviations, due to the fact that they are familiar with applications which recover from such incorrect queries. In HotelPlus this was achieved by using the Geospatial Endpoint Service of Bing maps with an http “GET” request to this link http://dev.virtualearth.net/REST/v1/Locations?q= +location+ &o=json&key= +bingapikey+ where +location+ is the search term and +bingapikey+ is the bing api key that one should create in order to get response from the above request. The response from this request is a list of locations which match most with the search term, regardless of the correctness of the term. For example, if a user searches for “brlin” instead of “Berlin, Germany” the application will provide a list which also includes Berlin in Germany. Additionally, if a user searches for “gr” instead of “Greece” the application will provide a list which also includes Greece.

It should also be noted that the name of the area where the user is currently located when using the app, is tracked by Geospatial Endpoint Service of Bing maps with the use of an http “GET” request to this link http://dev.virtualearth.net/REST/v1/Locations/" +latitude+ , +longtitude+ ?o=json&key= +bingapikey+ where +latitude+ and +longtitude+ are the geographical coordinates fetched by the device’s GPS.

3.4 Presentation

When opening the app, user is presented with app's splash screen which depicts the logo of HotelPlus.

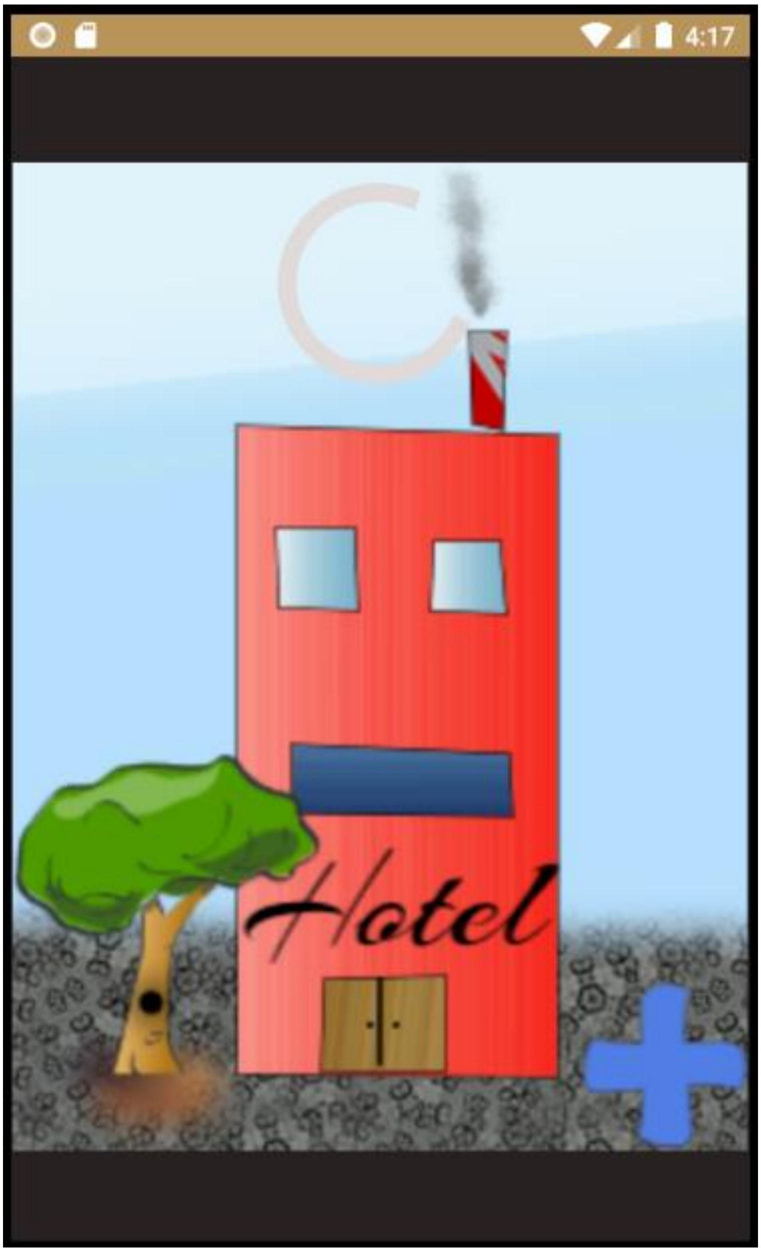


Image 1: Splash screen of HotelPlus

Afterwards, user is immediately redirected to a new screen where is prompted to fill a search field and press the "SEARCH" button in order to get hotels and points of interest of a wanted area. The search field is prefilled with user's current location if access to location is allowed. It is noteworthy that at this screen exist alternating background images of hotels which fade in and out every few seconds and are implemented as an animated list in background_animation_main.xml file. Last but not least, there is an ABOUT button which when pressed, it redirects the user to a new screen which provides general info and license of application as shown in Image 4 below.

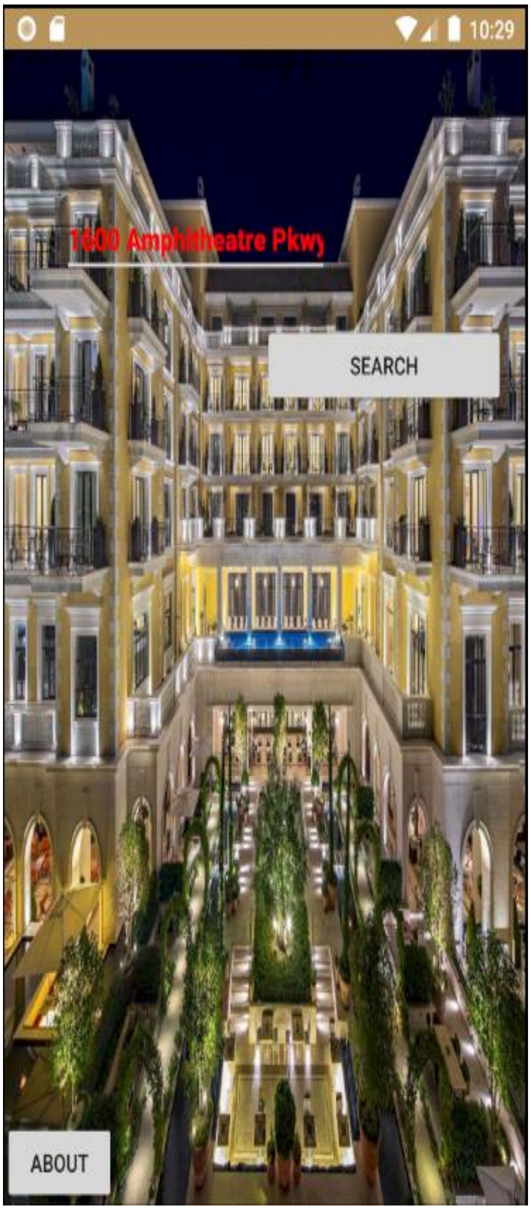


Image 2: Prefilled Search Field

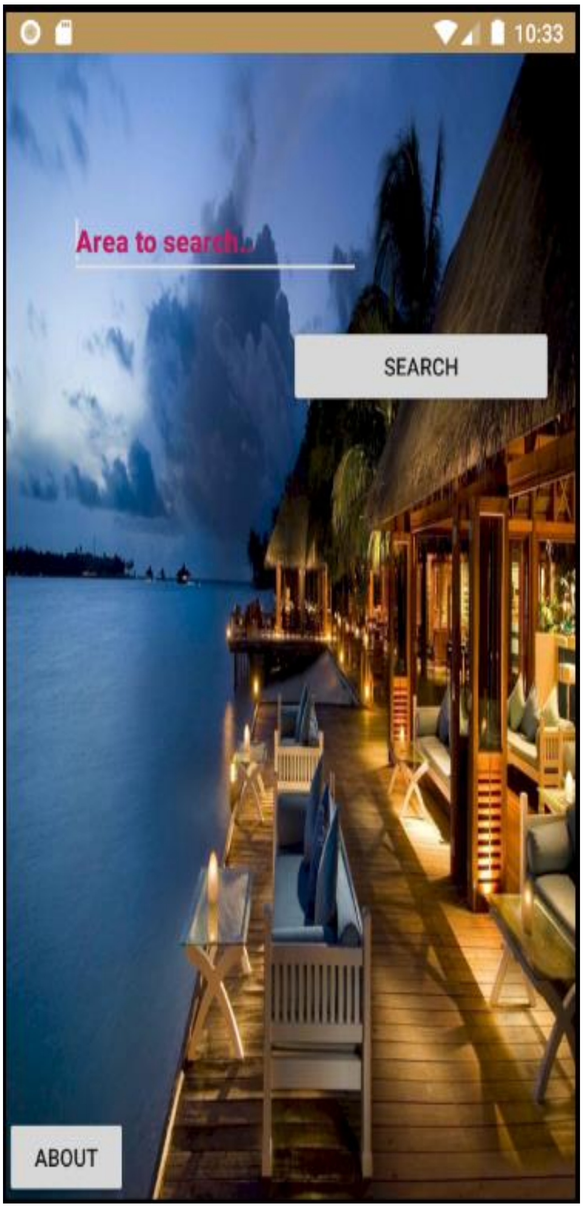
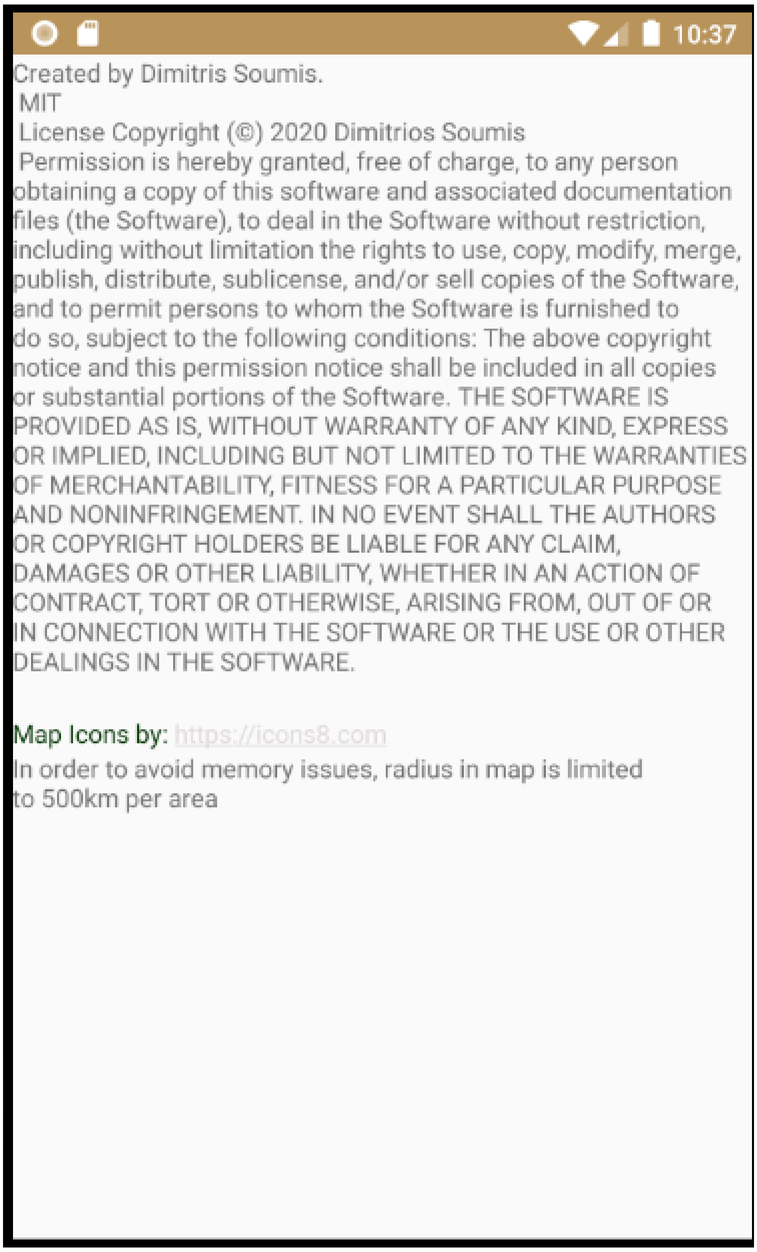


Image 3: Empty Search Field



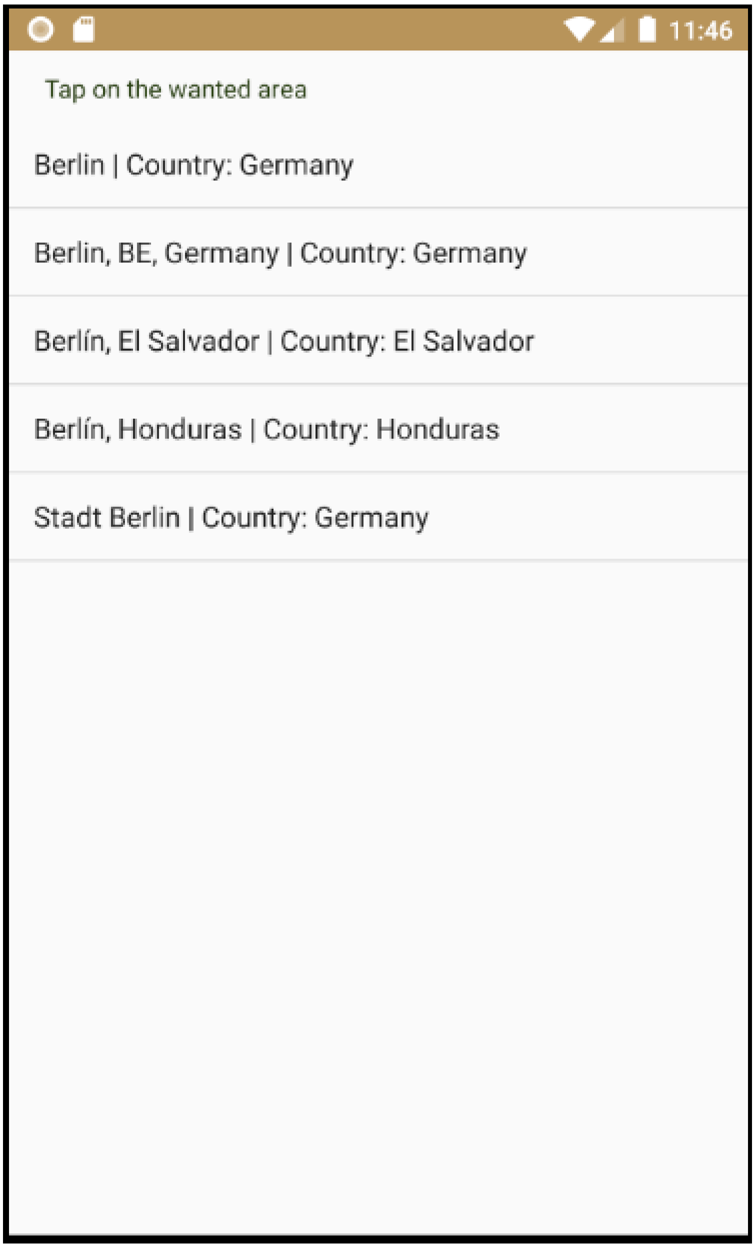
Created by Dimitris Soumis.
MIT
License Copyright (©) 2020 Dimitrios Soumis
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the Software), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions: The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software. THE SOFTWARE IS PROVIDED AS IS, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Map Icons by: https://icons8.com
In order to avoid memory issues, radius in map is limited to 500km per area

Image 4: ABOUT Screen

Note: User should type a region of Greece or Germany in the search field while app at current state [last modified: 8/1/2020] supports only these two countries.

After pressing SEARCH button, user is redirected to a screen which displays a list of options of geographical regions which match with the value of the previous search field. Next, user should just tap on an option which clarifies the wanted area in order to move to the next screen which consists of the map.



11:46

Tap on the wanted area

Berlin | Country: Germany

Berlin, BE, Germany | Country: Germany

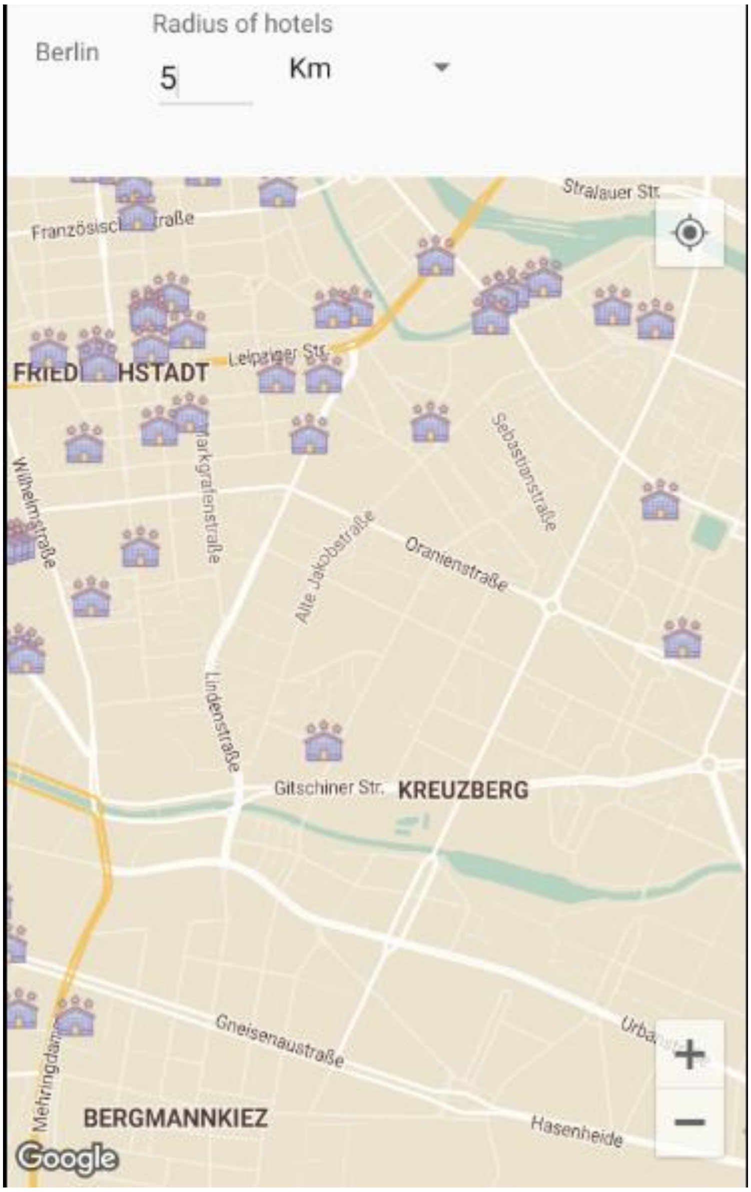
Berlín, El Salvador | Country: El Salvador

Berlín, Honduras | Country: Honduras

Stadt Berlin | Country: Germany

Image 5: List of options of geographical regions with search field value “berlin”

While at the screen which consists of the map, one can see the hotels which exist on the selected area and navigate freely on the map. One can see the name of the selected area on the upper left corner of the screen. Additionally, there is an input field next to the name of the selected area where the user can adjust the radius of selected area’s center for the hotels that are shown on the map(default value is 5, minimum value is 0 and maximum value is 500) and also a dropdown menu option where the user can change the distance measuring unit to the preferred one. Furthermore, at the upper right corner of the map there is a cyclic icon which when tapped it shows user’s current position on map (if access to location is allowed) and at the bottom right corner of the map two buttons + and – which zoom in and out of the map equivalently.



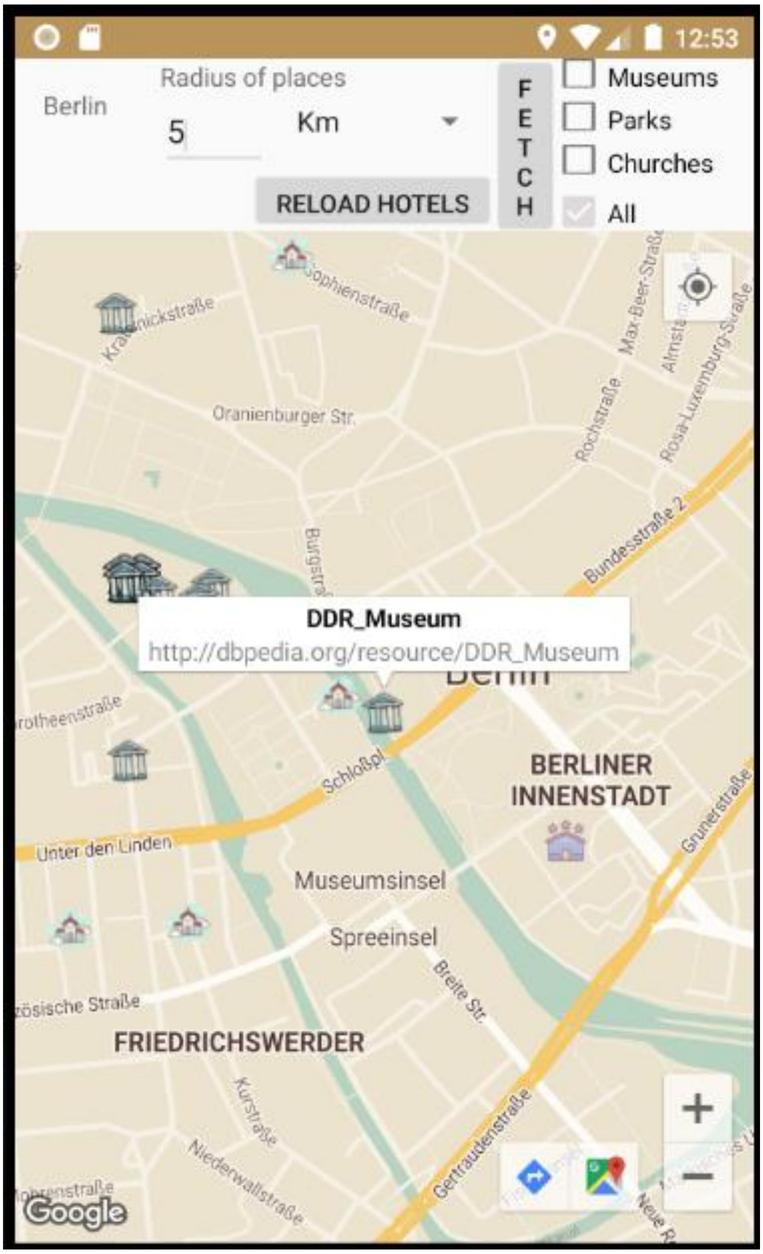
Berlin

Radius of hotels

5 Km

Image 6: Map of Berlin

When the user clicks on a hotel, the map focuses on this hotel and the rest of the hotels in the area become invinsible. In addition to this, more options are added on and above the map. Firstly, the input field for radius is currently measuring the maximum distance of points of interest/places around the hotel. One can select which points of interest wants to see on the map by checking the proper checkbox and pressing the FETCH button. “All” checkbox includes also skyscrapers at current state [last modified: 8/1/2020]. Moreover, the user can click on a point of interest or a hotel for an info window to show up above it which provides the name of the place or of the hotel equivalently and its website (if it is available) if it is a hotel or a link to DBpedia if it is a point of interest for further information about the site. The user can redirect either to a hotel’s website either to a DBpedia’s link by tapping the info window above the marker(=hotel or point of interest). Last but not least, if Google Maps application is installed to user’s device, the option of getting directions to this place is provided by the press of either of the two buttons, which show up at the bottom right corner, left of the zoom in and out symbols.



Berlin

Radius of places 5 Km

☐ Museums
☐ Parks
☐ Churches
☑ All

DDR_Museum
http://dbpedia.org/resource/DDR_Museum

Image 7: Example of clicked point of interest (museum) within the radius of selected hotel

4. CONCLUSIONS AND FUTURE WORK

This thesis gave me the opportunity to acquire some of the android development knowledge, expand my thought about application development, learn new techniques and also create an application which stands for itself and can be used by public. Additionally, in order to fully understand the technologies used in this thesis like the DBpedia SPARQL endpoint, I studied the fundamentals of Semantic Web, Linked Data and practiced on the dominant querying language for semantic knowledge bases called SPARQL. Furthermore, I read a lot of documentation about the APIs used and obtained a lot of knowledge about android user interfaces, permissions, maps and interactions with device’s system like internal and external storage.

As far as HotelPlus is concerned, I have in mind a lot of contributions that can be done in order to become an even better application. Firstly, I would like to add an autocomplete service in the region search field. Then, add some alerts in case of empty searches or unrecognizable characters. Secondly, at the map screen, I would like to replace “all” checkbox with more checkboxes in a dropdown menu and maybe abolish “fetch” button. Additionally, I would like one to have the ability to navigate on the map and see hotels by tapping on a specific area. Furthermore, I would like to make custom information windows above the markers(hotels and points of interest) which will include some extra information about them instead of just their name and website. Also, I want to beautify the buttons and fields used generally in user interface. Lastly, I would like to create a database which holds reviews for each site so users can add their review on the places they visit and obviously create a register/login system so users can write a review, save their favorite hotels and points of interest and personalize the whole experience.

ABBREVIATIONS - ACRONYMS

API	Application Program Interface
apk	Android Package Kit
URL	Uniform Resource Locator
HTTP	HyperText Transfer Protocol
SPARQL	Simple Protocol And Rdf Query Language
RDF	Resource Description Framework
HTML	HyperText Markup Language
JSON	Java Script Object Notation
CSV	Comma Separated Values
XML	Extensible Markup Language
SQL	Structured Query Language

REFERENCES

[1] Wikibooks, Semantic Web/The Vision; https://en.wikibooks.org/wiki/Semantic_Web/The_Vision. [Accessed 19/1/20]

[2] Axel-Cyrille Ngonga Ngomo, Sören Auer, Jens Lehmann, Amrapali Zaveri, Introduction to Linked Data and Its Lifecycle on the web

[3] M.Koubarakis, Reasoning Web 2014

[4] M.Koubarakis, Knowledge Technologies

[5] European Data Portal, Introduction to Linked Data

[6] W3, SPARQL; https://www.w3.org/2001/sw/wiki/SPARQL. [Accessed 19/1/20]

[7] W3, SPARQL By Example; https://www.w3.org/2009/Talks/0615-qbe/. [Accessed 19/1/20]

[8] Wikipedia, Android (operating system); https://en.wikipedia.org/wiki/Android_(operating_system). [Accessed 7/1/20]

[9] Frank Ableson, Introduction to Android development, Discover the depth and breadth of the android platform; https://developer.ibm.com/articles/os-android-devel/ .[Accessed 7/1/20]

[10] Learn about DBpedia; https://wiki.dbpedia.org/about.[Accessed 7/1/20]

[11] Kingsley Uyi Idehen, What is a SPARQL Endpoint, and why is it important?; https://medium.com/virtuoso-blog/what-is-a-sparql-endpoint-and-why-is-it-important-b3c9e6a20a8b.[Accessed 7/1/20]

[12] OpenStreetMap and the Overpass API;https://dev.overpass-api.de/overpass-doc/en/preface/preface.html.[Accessed 7/1/20]

[13] The Data Model of OpenStreetMap;https://dev.overpass-api.de/overpass-doc/en/preface/osm_data_model.html.[Accessed 7/1/20]

[14] Geospatial Endpoint Service;https://docs.microsoft.com/en-us/bingmaps/articles/geospatial-endpoint-service.[Accessed 7/1/20]

[15] Margaret Rouse, Google Maps; https://whatis.techtarget.com/definition/Google-Maps.[Accessed 7/1/20]