DOPŘÍRODY! – GEOINFORMATION MOBILE APPLICATION

MIROSLAV ČÁBELKA*, MICHAL JAKL

Charles University, Faculty of Science, Department of Applied Geoinformatics and Cartography, Czech Republic
* Corresponding author: miroslav.cabelka@natur.cuni.cz

ABSTRACT

Many people in Prague keep seeking a refuge in green areas within the city. Such a desire has inspired us to develop an app for cell phones that would be able to find natural areas in Prague. Users are navigated to the green spaces in real time and space. This article deals with the design and development of such app which has been aptly named DoPřírody! (To the Nature!).

To create such an app requires a sound systemic design and also connection with a number of different components. Technically, it is a geoinformation community system composed of database, server, and mobile parts. First, it was necessary to define natural areas. Second, database of such areas containing exact position, description, type, photos, etc. was created. The server part includes application modules that secure users management, area search, routing, navigation, etc. The mobile part consists of user interface for cell phones. Special attention is given to the design and creation of algorithm that calculates the appropriate route within the road network which serves as a base for the navigation. DoPřírody! is also compared with competing projects like Googlemaps, Mapy.cz and OSM. Route length, time required, and computing time are the basic elements compared.

DoPřírody! has all the features of a community system, i. e. logged-in users can contribute to further improvements. Anyone can update information on natural areas, upload more detailed descriptions, photographs, and comments. Thus, the whole system remains "live". At the present time there are some 80 users that have downloaded the free installation app on Google Play.

Keywords: android; network analysis; server; database; nature; mobile app

Received 28 March 2017; Accepted 1 November 2017; Published online 15 November 2017

1. Introduction

1.1 Motivation and chief aims

Many Czech cities boast large green areas within municipal boundaries; their positive effects on the quality of life is indisputable (Alcock 2014; Pondělíček 2012). Also the capital city of Prague, chosen as a test area for this app, has an exceptionally high number of natural or semi-natural biotopes (i. e. areas where the negative effects of human activities are minimal so far) (Kočí 2016). Such areas provide necessary space for a number of animals and plants including some protected species (ENVIS 2016). The urban green areas include parks, historically important forests, lines of trees, various protected areas, and spaces along water bodies. All these green areas contribute to the exceptional character of the city and create an attractive and special atmosphere.

How to find the most suitable and best available green area in a big city? Such that would best meet our expectations? Which is the easiest "escape way" from all the hustle and bustle? These questions can be answered using a mobile app that would help to find one's way when looking for a green area. At the moment there are a few web portals (with varying degree of quality and updates) that show and describe the green areas of Prague. For example, the Green Map of Prague (http://zelenamapa.cz) or Prague in Czech Atlas (http://www.czechatlas.com/prague). So far, however, there has not been any free app that would be available "in the field".

The article focuses on the role of modern technologies (based on geoinformatic platform) while designing and creating applications for cell phones. The authors strived to build the system in such a way so that in future it could be enlarged and improved by the users. A special algorithm has been created to calculate the shortest route which could be compared with competing products, i. e. with Google Maps, Mapy.cz, and OSM.

Such app requires a sound systemic design as well as links to many various components. Consequently, the app forms a part of a community geoinformatic system that consist of three mutually interconnected components:

1) Database of natural objects (areas, entrance points);
2) Algorithms and modules for data search;
3) Application part:
   a) mobile app,
   b) website (with database administration).

The system has been named DoPřírody! (To the Nature!), thus reflecting its character. It is available to the public for free as an app that can be downloaded at Google Play and also as a website. The app works on cell phones, users can find their way to natural areas on the territory of Prague. As it works as a community system, it is easy to add, edit, and share natural areas using the app

https://doi.org/10.14712/23361980.2017.20
Čábelka, M. – Jakl, M. (2017): DoPřírody! – geoinformation mobile application
AUC Geographica

(or website). The system can be accessed at http://mjakl.cz/doprirody, the application at Google Play.

Such a system did not exist among mobile apps so far. The article describes its design, creation of individual components, and mutual interconnectivity of the whole system.

1.2 Why Android operating system?

Android operating system is widespread among users, support and tools are readily available for computer programmers – that is why Android was chosen as platform for the application. Android has become the market leader in the field of mobile devices, at the moment it has more than 80% of the market (iOS 18%, Windows Phone 1%) (Mikudík 2016). Most software developers have focused on Android; consequently, a number of tools enabling easy and intuitive development of new applications came to existence, often aided by a huge amount of instructions and advices on different message boards.

Developing different applications for Android using JavaScript is a recent, dynamic field, featuring fast and unorganized emergence of new trends and practices. It is, however, also an attractive and modern subject with a whole plethora of authors and innovators (Appfigures 2015; Grant 2013; Konečný 2017). Android provides complex solutions for manufacturers and software developers. Instruction manuals, message boards, as well as tools intended for developing new applications are in general free – Software Development Kit (SDK) (Android Developer 2017). As printed material dealing with the above topic quickly becomes outdated, information was chiefly obtained from electronic sources and instruction manuals.

They are mainly technical universities, that specialize in the production and development of Applications for Android. That is way the scientific works produced mainly on technical universities (Kalčík 2013; Šarata 2015; Krejčí 2015; Njunjic 2012) proved to be a valuable source of inspiration and information.

1.3 System architecture

The system requires an uninterrupted communication between client's application and remote server. In order to optimize the pressure, three-layer model "client – server – database" was adopted. Client's app sends queries to the remote server, which has a much higher capacity and provides all computations. Afterwards, results are sent back to the client's app and displayed to the user. All data are stored in the relational database and made accessible to the server.

Such a three-layer architecture is currently used by a high number of applications that deal with data, i. e. by a substantial portion of modern company applications. Compared to the now outdated two-layer client-server model, the three-layer systems allows a better performance distribution among clients' devices and server; thus, client's applications can work even on the cheapest devices (Güner 2005; Techopedia 2015). The author and users appreciated this feature many times during the creative process and testing. For example, any reported problem with the server or database could be centrally fixed without the necessity to create updated versions of the app.

Unlike the server, requirements for client's application are low and these usually work trouble-free also on low-performance devices. Both the web application and the application for Android utilize the same server core of the system. The graphics of the final "product", however, are different, depending on the design abilities and operating modes of different devices.

The server part of the system and the database must be carefully protected – fact that may be seen as a disadvantage. Server must be very stable plus quality internet connection between client's application and server must be secured during the whole period of use. Relatively frequent failures of the database server (caused by the web-hosting provider) proved to be a big problem. When such a failure occurs, the client's application and the whole system become unavailable.

2. Data preparation, database design

Precise localization of natural areas makes up the key part of the system. In order to create database of natural areas, first it was essential to define "natural area" as such including different types and qualities.

2.1 Data on natural areas

Scientific sources cite a number of ways how to define natural areas (Deslauriers et al. 2017; Udržitelný rozvoj 2016). As the system has been developed on a community base, it became necessary to produce a new definition understandable for casual users. The following formula indicating natural areas was chosen:

Natural area is such a type of public space (accessible for free or subject to charge) that can bear at least 10 to 15 mature trees on a continuous green space. Visitors can hear birds singing and feel (at least seasonally) the scent of flowers there.

Field research has revealed that the above definition includes also some rather extensive areas with scattered trees and plants among solid surfaces (tarred roads, cobblestones). On the other hand, some small, yet environmentally valuable parks, plus vegetation along water streams also fit into the definition. The database is further divided into five subtypes that can be (de)activated by users:

Park/garden – Areas significantly altered and maintained by humans on a regular basis. These are mostly city parks and gardens, including large green areas within city squares.

Tab. 1 Table stored in database (example).

ID	Name	Type	Play ground	WC	Dogs	Submitted _by	Entry	Bench	Food	Sport	Light	Tree	Wheel chair	Bike	Run	Nature_Trail	Description	Region	Prot.
1	Folimanka	Park	1	1	1	Majkl	0	1	1	1	1	0	1	1	1	0		Praha 2	0
2	Na Karlově	Park	0	0	0	Majkl	0	1	0	0	1	0	1	0	0	0		Praha 2	0
3	Ztracenka	Park	0	0	0	Majkl	0	1	0	0	1	0	0	0	0	0		Praha 2	0
4	Havlíčkovy sady	Park	1	1	1	Majkl	0	1	1	1	1	0	1	1	1	0		Praha 2	0
6	Albertovské stráně	Park	0	0	1	Majkl	0	1	0	0	0	0	0	0	0	0		Praha 2	0
7	Náměstí Míru	Park	0	1	1	Majkl	0	1	1	0	1	0	1	0	0	0		Praha 2	0
8	Karlovo náměstí	Park	0	1	0	Majkl	0	1	0	0	0	0	1	0	0	0		Praha 2	0
9	Botanická zahrada UK	Botz	0	1	0	Majkl	0	1	0	0	0	0	1	0	0	0		Praha 2	0

Forest – Areas covered by dense, fully grown trees intended for economic or recreational purposes. Forests often play a key role in the landscape biodiversity. In most cases, forests are found outside cities or at city margins.

Cemetery – A special transitional type between park and forest, with graves and memorial places. Dogs and activities related to sport are often not allowed. Some users may not be happy to visit such areas; deactivation is possible.

Botanical garden – A special transitional type between park and orchard with a broad array of (exotic) plants, often kept in greenhouses. Such areas are usually subject to entrance fee. Botanical gardens may be preferred by some users; however, it can be deactivated, too.

Other green areas – All other types of green areas like meadows, pastures, vegetation along water streams, roads, and railways, etc. Any other areas covered by plants.

Several criteria that can be clearly assessed were chosen (illumination, benches, playground, entrance fee etc.). These allow users to narrow the search for appropriate area.

2.2 Creating the database

Relational database MariaDB was chosen as data storage device for DoPřírody! system. It presents a more advanced branch of MySQL developed on a communal base. The features of GNU GPL free software licence are respected (Wikipedia 2016). This database is compatible with MySQL and can be implemented on the webzdarma.cz servers that host the whole system. The structure of data storage within the relational database includes a number of tables (Kolář 1998; Zajíc 2016; PHPMyAdmin Team 2017).

The system base consists of tables that include data on natural areas and entry points. The first one of this double table features data on natural areas with all details; geographical location, however, is not included (Table 1). The second table – database of entry points to respective areas – includes geographical coordinates only. Areas and entry points are linked together using primary key (ID) (Figure 1).

Road network data are divided into sub-networks and stored separately for each area. Each sub-network includes table of nodes and table of road network edges. These are linked together using IDs of respective nodes that feature geographical coordinates. Each road network edge bears only the information on starting and final node (these nodes can be switched when required). System of sub-networks allow to reduce the data volume used by algorithm when computing the shortest way towards selected object.

Users' feedbacks and reviews form a very important part of the system. These are also sorted into tables and used by different system modules. Administrators only can make use of the "log" table that trace the users' activities within the system: all editing, new entries, and new registrations are stored here (Figure 1).

3. Server part of the system

Server forms the key part of the whole system. Server provides most computations, it receives all the information from client's applications, communicates with the database, and sends feedbacks to the users. Components of the server application part are divided into several modules that maintain intensive mutual links for the sake of easier creation and management. Each module includes one (or more than one) entry, processing, and exit pages with algorithms using the PHP scripting. Depending on requirements, modules can access tables stored in the database.

Users' management module – Provides registrations of new users and management/editing of registered users. Certain procedures offered by the system (updates of natural areas and entry points, etc.) require registration

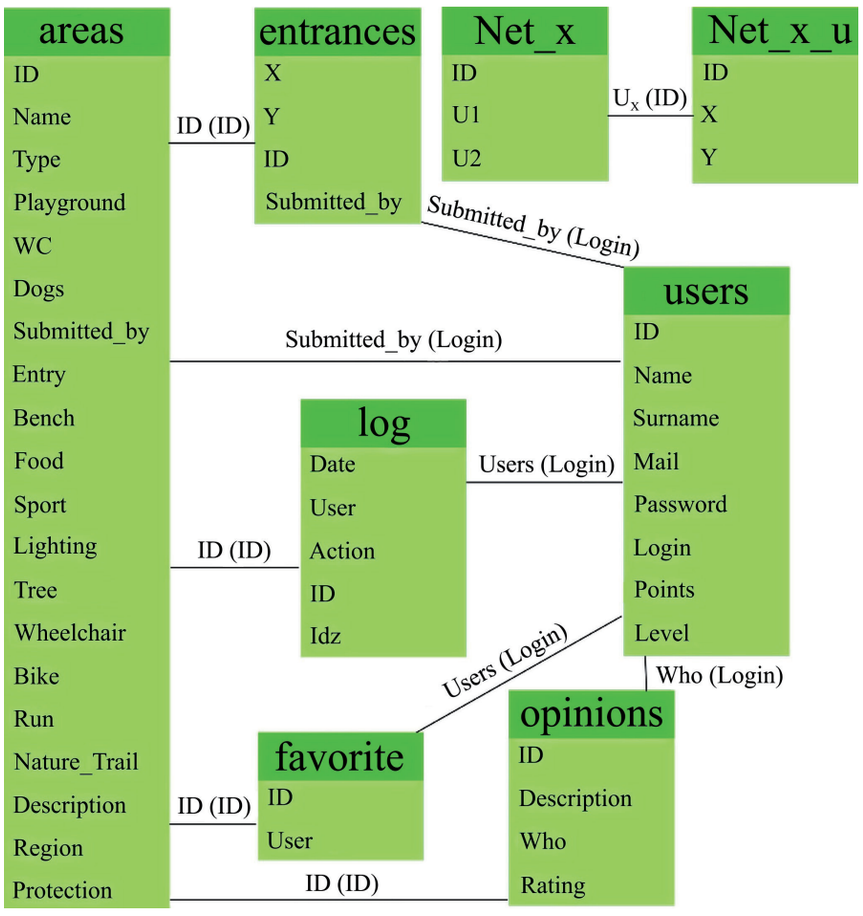

Fig. 1 Mutual links among tables ("x" indicates respective sub-networks within the database).

to secure safety and control. This module is firmly integrated and mutually linked with other modules. It provides users' authorization and records points that can be obtained by users for selected activities (thus motivating users to improve the system).

Area search module – One of the key modules. It receives users' requirements for the areas searched, creates SQL query, and consequently selects matching areas from the database. It also chooses the nearest (crow-flight) entry point for each of the selected areas. This is shown on map.

Route and navigation module – Secures the shortest way leading to selected object. When the routing can make use of the road network stored in the database, computation utilizes own algorithm (see below). The GPX file is subsequently visualized on map. When the routing includes sections that are not part of the database, users receive an aerial navigation supplemented by another route created by Graphhopper algorithm using Leaflet-routing-machine on the OSM network (Liedman 2016). This module also allows tracing in real time, and – if necessary – recalculation of a new, more suitable route.

Detailed module – Provides detailed list of all features of the selected natural area. This also includes pictures that are assigned to respective areas by areas ID stored on the server. The module allows registered users to input feedbacks and comments.

Update module – It is closely linked with the detailed module. It allows registered users to update their comments and to alter area descriptions.

Add module – Community based system requires that users could extend the database of natural areas and entry points. This module registers new data into the database of areas / entry points.

Favourite areas module – Registered users can add selected natural areas among favourites. Hence, client's application allows users to browse through his/her favourite areas plus areas that he/she reviewed, updated or added to the system in the past.

Help module – Fully integrated, essential module that makes the whole system coherent. Provides guidelines how to use other modules.

Information and statistics module – Fully integrated module with high internal heterogeneity. It collects and stores information on quality and speed of route planning, on the number and location of natural areas in the database. This module informs users about system updates and changes within the system core and client's applications. It also secures clients' feedbacks.

Administration and management module – It can only be used by the author and allows easier management of the system. The module includes different management tools, it also registers all users' activities within the system.

3.1 Maps

Maps form the initial or final parts of some modules. Projection of suggested routes into the map proved to be an uneasy task with a number of possible solutions. It was intended to use the same projection type for web client and for Android mobile app. That is why JavaScript library Leaflets, enhanced by plugin from Maxim Petazzoni for GPX files, was chosen (Petazzoni 2016). This solution works excellently in both client's applications.

3.2 Finding the shortest route

The algorithm responsible for finding the shortest route in the road network stored in the database presents the crucial and most complicated of all algorithms within the system. In theory, some of the existing algorithms may have been used (Berg et al. 2008; Algoritmy.net 2016; Friebelová 2011; Hliněný 2008). However, as the system has a number of special features (frequent transitions among different sub-networks, etc.), a new, tailor-made algorithm that would fully fit specific requirements of the system posed a challenge. The difference between our algorithm and the often used Dijkstra algorithm is that the Dijkstra algorithm scans all network points. Problems that arise include, for example, blind trips and possibly over-encryption. (Kolář 2004; Cormen et al. 2001). Creating a new algorithm also allowed a new innovative approach that can address problems of navigation in the road network.

The search algorithm is triggered immediately after user selects the object that should be reached; the search for best routes can start.

If the selected object was too far from the nearest part of the route network (more than 500 metres), the route is shown on crow-flight basis and supplemented by visualisation that utilizes Graphhopper algorithm the OSM data. If the user's position was near the road network stored in the database, the algorithm works as follows:

1. The surrounding is detected – Database network is not continuous, thus the unnecessary parts can be ruled out at the beginning which speeds up the computations.
2. Catching up with the road network – The nodes nearest to start and selected object are found. In this way algorithm is "attached" to respective nodes and can proceed to find the best.
3. Sub-network selection – A sort of a special "switch" allows that algorithm could pass through different sub-networks while conducting iterations. This file switches among different sub-networks depending on the node or edges names.
4. Movements within the sub-network system – This section of the algorithm forms the main iterating part. Operations are repeated till the node (identified as the target node) is reached. The algorithm is located at position S (Figure 2) at the beginning of this cycle. All road network edges containing this point (initial or final one) are found in the database.
   a) Linear movement – It follows if the S point was found in less than three road network edges. If S point was the closest point to the user' position, algorithm would move in a direction that brings it closer to the selected object (target). In all other cases the algorithm is already moving in a certain direction (from S – 1 point); consequently, it will continue in the same direction (towards S + 1 point) and not vice versa. The S + 1 point would further become a new "starting point" S and its coordinates are recorded into GPX file. If the S + 1 point was not yet the final desired node, the iteration continues.
   b) Intersection – If the S point was found in three or more road network edges, the algorithm has reached an intersection and continuation will be a bit more complicated. If the S point was found in four road network edges, three possible further directions exist (the S – 1 point is omitted). First, the algorithm identifies the number of edges that contain each of three $S_x + 1$ points. In case any of the $S_x + 1$ points would be identical with the target, the algorithm moves directly towards point 5).
      I. Dead end edge – If the $S_x + 1$ point was found in one edge only, it must be a dead end. Such point is omitted in further calculations and the next node is examined (in this case only two more candidates for an appropriate S point exist).
      II. Familiar node – Sometimes happens that in the past the algorithm already examined the selected $S_x + 1$ point. Such node is also excluded from further calculations to avoid unwanted circulations.
      III. Direction preference – If the $S_x + 1$ node fits the above criteria, the following parameters are calculated: distance from the object (target), distance towards the midway point between starting and final point, and the angle between two lines (connection S – 1 and target, connection $S_x + 1$ and target). Sum of these three variables makes up the final "score" of $S_x + 1$ node and it is compared with other possible candidates. The node featuring the lowest "score" (i. e. the one which is closest in the possibly straight direction) will be selected as a new S point. Its coordinates are recorded into the GPX file.

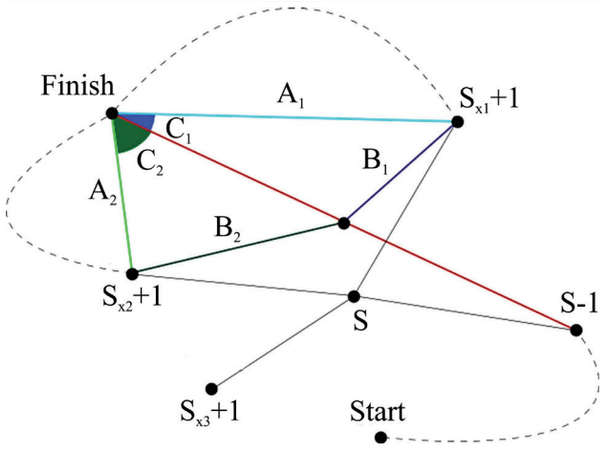

Fig. 2 One step of the navigation algorithm (diagram)

IV. Safe return – Sometimes algorithm selects a straight direction which may, however, later reach a dead end. It also may happen that no further continuation is possible beyond an intersection (usually due to the rules described under i and ii). As two movements through the same node are forbidden, the algorithm must return one step back. It goes back to the last intersection (the visited intersections are registered; this is the only exception when algorithm is allowed to enter again the already visited node) and moves in a different direction. If even such a movement was not be possible, the algorithm must further "reverse" until it finds a possible onward route.

"Reversing" means that the GPX file contains error points which creates a messy network. Thus, the unused nodes must be deleted from the GPX file. When the algorithm reaches the same intersection second time – let it be node No. 2751 –, all points registered after the first visit of the node 2751 are deleted from the system. Consequently, the final route consists only of "successful" steps of the algorithm.

5. Reaching the destination – After having reached the final node, user is redirected to the map page where the route is shown. Information on the route length and time required to reach the destination (by walking) is displayed, using the Leaflet, plugin GPX functions. Time needed to cover the distance is based on the average speed 4,500 metres per hour (75 metres per minute). In other words, one meter of distance equals to 0.8 seconds (or 0.013 minute).
6. System failure – Under certain circumstances, failure may be triggered purposely as a safety brake. When the algorithm "gets lost" in the network (i. e. needs to reverse a number of times and computation takes a rather long time), the process is terminated by an error message. System failure is recorded into the internal files.

4. Client's application

Mobile client's application uses the very same websites and scripts as client's application for PC. The size of display, however, is fundamentally different (20″ vs. 5″) and the same applies to connection speed (tens of kb per second when using GPRS/EDGE vs. fast connections of hundreds of MB per second). As a result, it became necessary to secure appropriate formatting for mobile application (Figure 3). Thus, the system includes two different CCS styles: one secures correct appearance of the web app, the second one that of the mobile app. Each URL sequence in the mobile app is followed by information that results optimized for cell phones are required.

Fig. 3 Optimizing the first page for web browser and cell phone (example).

WebView components that enable to display websites (Android Developer 2016) secure communication with the remote server. Adding GET strings behind URL pages or sending data by the POST method allow to create users' queries. WebView components are widely used in many activities and often are integrated into the activity core in such a way that users can not distinguish between parts generated by the website and those that remain part of the activity core.

Another advantage of the mobile app is that the user's log-in details can be stored in the system. Consequently, registered users enjoy immediately all privileges without the need to fill in the log-in details again and again. This works also in case that user would like to log in on a different device.

Different languages are easy to implement in the Android client's mobile app. All captions are displayed either in Czech (if Czech was set as the prime language) or in English (for all non-Czech languages). Information on the language used is being sent to the server with each URL address and texts in the same language are produced.

5. Community character of the system, distribution, safety features

The two modules that enable addition of natural areas including entry points and users' management are essential for the open character of the system. Registered users can rate the respective natural areas, leave comments, and

update information. New natural areas and entry points can be added either using the web application or directly in the open field with the client's app for cell phones. Users can communicate with the author, report imperfections, and share ideas that may lead towards further improvements.

The open character of the system which allows easy editing and adjustments by users, however, also brings a high risk of hacking. Thus, high level of data security consisting of several tiers is necessary. All users' passwords are encrypted. All operations when users input data for communication with the database are protected against SQL-injection attacks. There are special scripts integrated into the system that can immediately and permanently block IP address of anyone with irregular behaviour.

No web / system security, however, can be absolutely perfect. For the sake of safety, full backups of database and scripts are provided regularly.

6. Conclusions – comparison with competing systems, practical use

The ability to find the shortest route leading towards selected destination forms the most important feature of the DoPřírody! system. The best way how to test quality of the algorithm used would be to compare it with widely used web mapping services. Inside information on algorithms and computing processes used by services like Mapy.cz or Google Maps, however, are not available to the public; consequently, a detailed comparison is not possible.

To provide an approximate comparison at least, routes suggested by DoPřírody!, Google Maps, Mapy.cz, and OSM were tested on a number of examples – always between the same starting and end points. Some of these comparative analyses are shown in the appendix.

Comparisons show that the algorithm used by DoPřírody! provides results that are fully competitive with those offered by the big commercial services. However, due to the strictly geographical logics of DoPřírody!, in some cases the algorithm may suggest a slightly longer route.

The time needed for computations (which is longer compared to the big competitors) remains the only real imperfection of DoPřírody!. However, the speed can be increased in future by implementing of a few planned improvements. Though DoPřírody! can not secure 100% success as regards the final route so far, the last updates reduced the system failures to a minimum. At the moment, routes are displayed successfully in 92% of cases.

In the case of failures or excessively long routes, analyses mostly showed that it is the road network – not yet complete – that is responsible for imperfections. Still, under most circumstances the algorithm used by DoPřírody! brings results comparable with big mapping services.

At the very beginning, there were a number of "grey zones" and unclear solutions. These gradually disappeared as the work was advancing. With the growing amount of work also experience mounted and even some "dead ends" brought new ideas, modules, and better solutions that helped to enhance the already existing parts of the system.

Hundreds of hours spent on developing DoPřírody! brought invaluable experience regarding databases and Android-related web programming. Users' feedbacks reveal potential for future development. The work on DoPřírody! triggered developing of many other mobile apps, some with commercial character (geolocation game Real World RPG – www.realworldrpg.info; app developed for Turistické známky and TéTetka collectors; app for participants of Závod tří vrchů – ŠerLok, etc.). These are currently used by hundreds of people.

7. Appendix – comparison with competing services

The route displayed in the following figures (maps) connects Prague Castle (50.090 N, 14.400 E) with Vinohrady (50.074 N, 14.444 E), i. e. it crosses much of the territory Prague 1 and Prague 2. DoPřírody! was primarily developed for these city parts. The route was calculated by four services: Mapy.cz, Google Maps, Graphhopper.com, and DoPřírody! The time needed for calculation was measured three times each; average of the three is shown. Tables include comparison of other routes.

Mapy.cz – shortest route 4.4 kilometres, estimated walking time 75 minutes, computing time ca. 2 seconds.

Google Maps – shortest route 4.5 kilometres, estimated walking time 59 minutes, computing time ca. 3 seconds.

Graphhopper.com (OSM data) – shortest route 4.4 kilometres, estimated walking time 53 minutes, computing time ca. 3 seconds.

DoPřírody! – shortest route 4.5 kilometres, estimated walking time 59 minutes, computing time ca. 6.5 seconds.

Tab. 2 Start 50.075 N, 14.420 E, end 50.082 N, 14.424 E

Service	Distance (km)	Walking time (min)	Computing time (s)
Mapy.cz	0.876	0:13	1
Google maps	0.900	0:11	2
Graphhopper.com	0.895	0:11	1
DoPřírody!	1.063	0:14	2

Tab. 3 Start 50.069 N, 14.450 E, end 50.078 N, 14.431 E.

Service	Distance (km)	Walking time (min.)	Computing time (s)
Mapy.cz	1.800	0:34	2
Google maps	1.800	0:25	2
Graphhopper.com	1.860	0:22	1
DoPřírody!	2.102	0:27	3

Tab. 4 Start 50.073 N, 14.410 E, end 50.077 N, 14.439 E.

Service	Distance (km)	Walking time (min)	Computing time (s)
Mapy.cz	2.400	0:42	1
Google maps	2.400	0:33	2
Graphhopper.com	2.410	0:29	1
DoPřírody!	2.455	0:32	3

Tab. 5 Start 50.070 N, 14.430 E, end 50.090 N, 14.410 E

Service	Distance (km)	Walking time (min)	Computing time (s)
Mapy.cz	3.100	0:47	2
Google maps	3.100	0:37	2
Graphhopper.com	3.130	0:38	1
DoPřírody!	3.748	0:49	5

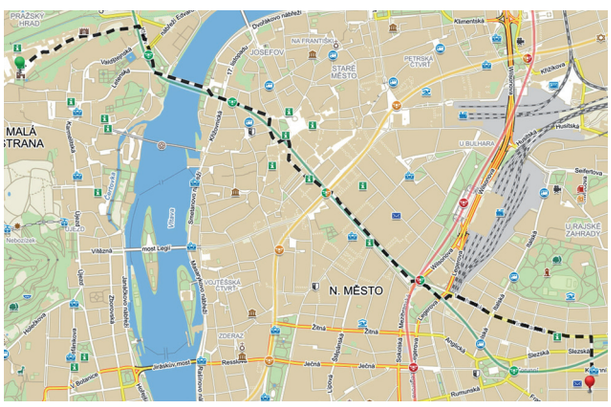

Fig. 4 Route produced by Mapy.cz.

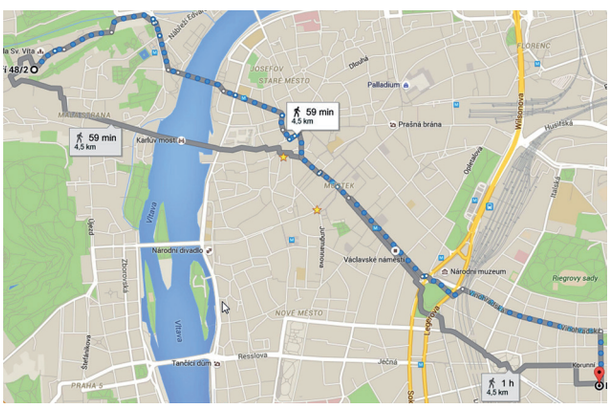

Fig. 5 Route produced by Google Maps.

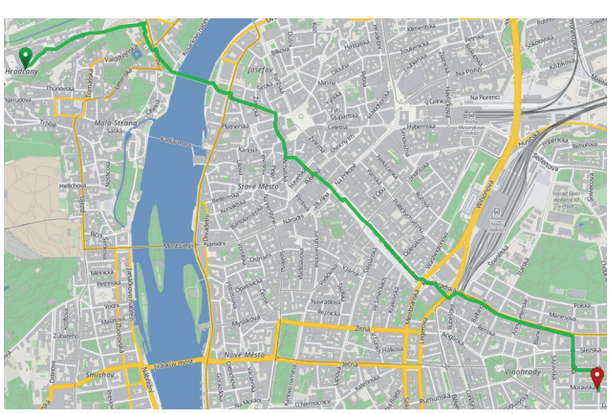

Fig. 6 Route produced by Graphhopper.com.

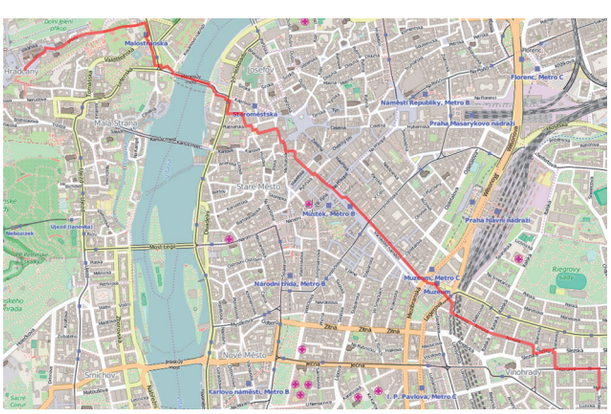

Fig. 7 Route produced by DoPřírody!

REFERENCES

ALCOCK , I., WHITE, M. P., WHEELER, B. W., FLEMING, L. E. (2014): Longitudinal Effects on Mental Health of Moving to Greener and Less Green Urban Areas. Environmental Science & Technology 48(2), 1247–1255. https://doi.org/10.1021/es403688w

ALGORITMY.NET: Problém nejkratší cesty [online]. [downloaded 8.1.2017]. https://www.algoritmy.net/article/36597/Nejkratsi-cesta

ANDROID DEVELOPER [online]. [downloaded 2.1.2017]. http://developer.android.com/tools/studio/index.html

ANDROID DEVELOPER: WebView [online]. [downloaded 2. 1. 2017]. https://developer.android.com/reference/android/webkit/WebView.html

ANDROID DEVELOPER: Android SDK [online]. [downloaded 2. 1. 2017]. https://developer.android.com/studio/releases/sdk-tools.html

APPFIGURES blog [online]. [downloaded 2.1.2017]. http://blog.appfigures.com/app-stores-growth-accelerates-in-2014/

BERG, M., CHEONG, O., KREVELD, M., OVERMARS, M. (2008): Computational Geometry – Algorithms and Applications. Springer.

CORMEN, T. H., LEISERSON, C. E., RIVEST, R. L., STEIN, C. (2001): "Section 24.3: Dijkstra's algorithm". Introduction to Algorithms (Second ed.). MIT Press and McGraw–Hill. pp. 595–601.

DESLAURIERS, M. R., ASGARY, A., NAZARNIA, N., JAEGER, J. A. (2017): Implementing the connectivity of natural areas in cities as an indicator in the City Biodiversity Index (CBI). Ecological Indicators, https://doi.org/10.1016/j.ecolind.2017.02.028.

ENVIS – Informační systém o životní prostředí v Praze [online]. [downloaded 8.1.2017]. http://envis.praha-mesto.cz

FRIEBELOVÁ, J. (2011): Síťová analýza, České Budějovice. [online]. [downloaded 2.1.2017]. http://www2.ef.jcu.cz/~jfrieb/rmp/data/teorie_oa/SITOVA%20ANALYZA.pdf

GRANT, A. (2013): Android 4: průvodce programováním mobilních aplikací. Brno: Computer Press.

GÜNER, S. (2005): Architectural Approaches, Concepts and Methodologies of Service Oriented Architecture, Master Thesis, 126 p., Technical University Hamburg, Harburg.

HLINĚNÝ, P. (2008): Teorie Grafů. Brno: MU, Fakulta informatiky.

JANOVSKÝ, D. (2016): Jak psát web: o tvorbě, údržbě a zlepšování internetových stránek, [online]. [downloaded 2.1.2017]. http://www.jakpsatweb.cz

KALČÍK, V. (2013): Implementace GIS nástroje pro mobilní počítačová zařízení. Diplomová práce, VUT Brno.

KOČÍ, P. a kolektiv: Nejzelenější česká města při pohledu z vesmíru [online]. [downloaded 2.1.2017]. http://www.rozhlas.cz/zpravy/data/_zprava/nejzelenejsi-ceska-mesta-pri-pohledu-z-vesmiru-karlovy-vary-praha-ostrava--1469125

KOLÁŘ, J. (1998): Geografické informační systémy 10. Vydavatelství ČVUT, Praha.

KOLÁŘ, J. (2004): Teoretická informatika. 2. vyd. Praha: Česká informatická společnost.

KONEČNÝ, M. (2012): Vyvíjíme pro Android [online]. [downloaded 8. 1. 2017]. https://www.zdrojak.cz/clanky/vyvijime-pro-android-zaciname/?utm_source=top&utm_campaign=category

KREJČÍ, P. (2015): Vývoj aplikace pro OS Android, bakalářská práce, VŠE Praha.

LIEDMAN, P.: Leaflet Routing Machine [online]. [downloaded 2. 1. 2017]. https://github.com/perliedman/leaflet-routing-machine

MIKUDÍK, R.: Android s iOS drtí mobilní svět. Ostatní jsou v klinické smrti [online]. [downloaded 2.1.2017]. http://mobil.idnes.cz/android-s-ios-drti-mobilni-svet-dmu-/mob_tech.aspx?c=A160225_194408_mob_tech_ram

NJUNJIC, I. (2012): Development Techniques for Android Platform Mobile Device Application, Doctoral Dissertation, 181 p., Eastern Michigan University, Ypsilanti, Michigan.

PETAZZONI, M.: GPX plugin for Leaflet [online]. [downloaded 8.1.2017]. https://github.com/mpetazzoni/leaflet-gpx

PHPMYADMIN TEAM [online]. [downloaded 2.1.2017]. http://www.phpmyadmin.net

PONDĚLÍČEK, M. (2012): Zeleň v urbánním prostoru jako indikátor kvality života města. Dizertační práce. Vysoké učení technické v Brně, Brno.

RAPANT, P (2002): Úvod do geografických informačních systémů. VŠB-TU, Ostrava.

ŠARATA, J. (2015): Pokročilé prostorové vyhledávání v mobilních GIS aplikacích. Diplomová práce, UPOL Olomouc.

TECHOPEDIA: Three-Tier Architecture. [online]. [downloaded 2.1.2017]. https://www.techopedia.com/definition/24649/three-tier-architecture

Udržitelný rozvoj: Co je to příroda? [online]. [downloaded 2. 1. 2017]. http://www.udrzitelny-rozvoj.cz/clanky/co-je-to-priroda

WIKIPEDIA: MariaDB [online]. [downloaded 2. 1. 2017]. https://cs.wikipedia.org/wiki/MariaDB

ZAJÍC, P.: MySQL [online]. [downloaded 2.1.2017]. http://www.linuxsoft.cz/mysql/