

The Route Planning Services Approach for People with Disability

Kirill A. Kulakov, Anton I. Shabaev, Irina M. Shabalina
 Petrozavodsk State University (PetrSU)
 Petrozavodsk, Russia
 kulakov@cs.karelia.ru, {ashabaev@, i_shabalina}@petsu.ru

Abstract—This paper describes approach and methods used for the development of navigation services for people with disabilities. Our previous papers contained the description of informational environment development for disabled people. The environment includes several applications utilizing unified database and providing the information support to disabled people. The service “Accessibility Passports” was developed to collect information about accessibility of the objects. “Accessibility map” service visualizes the information on the geographical map. This paper is focused on the development of the key service of environment – “Social navigator”. Recently we have presented the service only conceptually without detailed description of navigation approach and methods with adaptation to personal mobility restriction. The paper describes developed methods of navigation for disabled people and elaborated algorithms. Also the practical opportunities of the service to support disabled people are presented. The results of approach study and development work are also presented in the paper.

I. INTRODUCTION

This work is a part of long-term project [1] related to the development of information environment for people with disabilities. The aim of the project is to improve life quality of people with disabilities by addressing issues related to social exclusion, accessibility and mobility of disabled people by means of advanced information and communications technology (ICT).

As it was noted the infrastructure of cities in Russia is not friendly for disabled people [2], [3]. Inaccessibility or limited accessibility of social facilities increases personal social isolation and exclusion from the life of the community.

The informational infrastructure that contains several mobile and web services is being developed in order to provide the informational support and to increase the accessibility of facilities. The information for filling the data resources of services is provided by authorities, volunteers and other users. The main aim of the services is to provide the required information about social objects and routes to the users with various types of restrictions and to propose the audio assistance in addition to visual

informing. The infrastructure includes not only mobile services – web solutions are also used for informing disabled people. The earliest service developed under the infrastructure was “Accessibility Passports” [3]. The service was developed and put into operation by the Ministry of Healthcare and Social Development of Karelia. Web-service was developed for preliminary collection of accessibility data about socially significant objects. But the whole information from the “Accessibility Passports” is not available via public service. The database contains special and “closed” data. “Accessibility Map” mobile service was elaborated for public submitting of collected accessibility information [4]. After mobile version of the service corresponding web-solution was developed. The specification of the service is presented in this paper. The service allows visualizing information about social facilities categorized by accessibility levels and types of disability.

Recently the conceptual description of the “Social Navigator” was presented as a mobile route planning service adapted for abilities of people with various restrictions [4]. The service is the key element of developing informational infrastructure because it integrates the collected accessibility information, uses the advantages of previously developed applications and provides functionality for on-line support of disabled people.

The state of the art evaluation of route-planning and navigation technologies showed that standard navigation solutions are not friendly enough for people with disabilities because they provide universal solution without taking into account disability restrictions. Standard decisions provide the pedestrian route which can be easily overcome by healthy person but cannot be passed by the person on wheelchair or visually restricted one.

Section II contains the review of libraries and technologies that are used for navigation and route planning solutions at the present time. Section III is dedicated to detailed description of “Social navigator” mobile service, its architecture and functions, developed methods of navigation for disabled people and elaborated algorithms.

Section IV describes the “Social navigator” service implementation. The section contains the description of user interface and demonstrates the practical usage of the application. In conclusion we summarize the overall results of the work on the “Social Navigator”.

II. RELATED WORK

At the present time the following libraries and technologies are used for navigation and route planning solutions.

The Google Directions API [4] is a service that builds routes between locations using an HTTP request. It allows searching for routes for several modes of transportation, including transit, driving, walking or cycling. The route could be specified as a text description of waypoints or as latitude/longitude coordinates. The Directions API can return multi-part route using series of waypoints in JSON and HTML formats. The service uses static (known in advance) coordinates for placement the route on a map. This service is not designed to respond in real time to user’s input. The service can be used only with Google Maps.

JavaScript API Yandex.Map [5] is a set of JavaScript-components targeted to creation of interactive maps. API provides the opportunity of automobile route building. The route is calculated automatically, the arbitrary number of stopping and transit waypoints could be determined before the calculation.

The Open Source Routing Machine (OSRM) is a C++ implementation of a high-performance routing engine for shortest paths in road networks [6]. It combines sophisticated routing algorithms with open and free road network data of the OpenStreetMap (OSM) project. OSRM allows computing the shortest path on a continental sized network.

Studying various sources we found several services provides route taking into account wheelchair accessibility [7]. But most of them do not work in Karelia region and provide ability for other types of disability.

First of all the presented libraries are not oriented to people without limited mobility’. The databases of presented services have enough information for moving by automobile or by public transport, but have a lack of description of pedestrian ways features. On the building process of pedestrian route for disabled people some insignificant for ordinary people features should be accounted. For example, the height and number of steps on the way, availability of ramps have a great influence on the accessibility of the route for disabled people. “Social navigator” should be able to make route planning considering individual restrictions of a person and availability of pedestrian paths. To create such service we should collect the database of obstacles and features of pedestrian ways and only after filling the base start to build

the routes. The routes built for disabled people can be different with the routes provided by the OSRM, Yandex.Map and Google libraries.

In this way the route building issue can be formulated as the task of searching obstacle-avoiding way. This can be solved by finding a smooth, obstacle-avoiding curve on the plane [6]. Other solution is to build polygonal path taking into account simple obstacles in a form of intersecting polygons [8]. Nonetheless the graph performance of paths is implemented in the most common route planning algorithms.

III. “SOCIAL NAVIGATOR” SERVICE DESCRIPTION

A. Description of the service’s functions

In “Social navigator” service the routes are described as a graph with weighted edges, where the weight is the rate of accessibility of the path parts [2]. The mathematical method for estimation of accessibility according to various types of ability restrictions is based on graph theory and provides an approach to formal ranging of the routes according to the accessibility level of people with limited mobility.

The “Social navigator” is a part of unify information environment. The service uses the same information as the “Accessibility map” and “Accessibility passports” and provides information to “Audio-assistant” [3]. The database of OpenStreetMap is utilized in the service. The user information on obstacles and road “trouble places” is also collected and stored in “Social Navigator” database.

The development of “Social Navigator” takes into account the requirements and features of route-planning software for people with disability and involves:

- development of mathematical approach and elaboration of algorithms of route building on the graph on condition of various types of obstacles;
- elaboration of the interaction technology between existing OpenStreetMap database which contains roads, pedestrian paths and obstacles, and the data gathered by users and local authorities.

In addition to the route planning, the service will provide functionality for collecting user feedback regarding conditions of the route. Such information can be summarized and provided to relevant authorities to help improve the infrastructure for people with disabilities in the most demanded areas. In this case the issue of elaborating the method to increase the user data reliability is arose.

“Social navigator” service allows planning the pedestrian route taking into account individual restrictions of the user [2]. The service also provides functionality for collecting user feedback regarding conditions of the selected route. The information can be forwarded to

relevant authorities to help to fix problems as soon as possible, or at least to inform other users. Using traces of real trips, the service will enable a better planning of next journeys, e.g. by using the actual travel time for the person with given restrictions. By analyzing trip planning requests the service will discover travel bottlenecks, i.e., unfriendly areas for people with disabilities that cause most of the inconveniences, e.g., need to take longer alternative routes. Such information could be summarized and provided to relevant authorities to help to improve the infrastructure for people with disabilities in the most demanded areas.

After analyzing the requirements and use cases the set of significant obstacles was determined. The service allows fixing information on crossroads, stairs, direction indicators, road defects and seasonal obstacles like snowdrifts. The route planning function uses information on possibility of obstacles for various types of personal mobility limits. If the person is blind, then most of the traffic lights, stairs and road defects are not a critical problem, but using uncontrolled crossroads and walk through seasonal obstacles increases complexity of the route and requires increased attention. Visual direction indicators without duplication to other senses make the route impassable too.

Thereby the "Social Navigator" service has the following main functions:

- Data collection. Information about obstacles and "trouble places" should be collected and stored in service database. New obstacle is stored into user's private space and can be copied to the public space and/or shared with friends. The completeness of data guarantees the quality of planned routes.
- Route planning. Using the current user position, road graph and complexity factors for each road section service provides optimal route based on the navigation algorithm.
- Route sharing. The service allows sharing route between friends using social networks like Facebook. The note about shared route includes image with map, link to the detailed info in service, complexity estimations and found obstacles.
- Audio-assistance provides the information about obstacles and objects around during the trip.

The service has to maintain the work in off-line mode. All required data can be automatically downloaded to the user's device, because Internet connection may be slow or disabled during the trip.

B. The high-level architecture

The high-level architecture of the service is divided into several interacting parts (see Fig.1). Social navigator

service includes several application storages and web-services.

The obstacles which are found by users are stored in the GeTS – Geo2tag-based server. It combines Geo2tag LBS platform [9], [10] and application programming interface for accessing via web. Users can store founded obstacles by using GeTS Supplement application or through special web-interface. Founded obstacles are divided into categories like "crossroad" or "ramp" and GeTS provides mapping between disabilities and obstacle categories.

By default points are stored in private space because they may be unreliable [2]. All data in private space are available only for owner. Then the user might propose these points to the public space. The data from public space are available to everyone. Trusted user might automatically publish points to the public space.

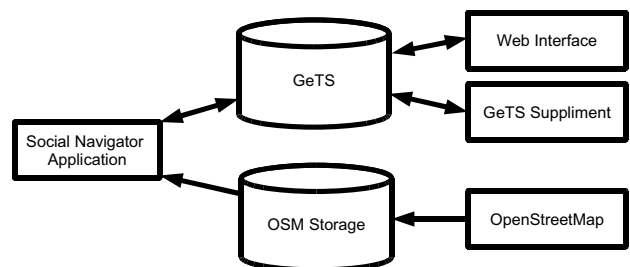


Fig.1. High-level architecture

The Route planning function is based on OpenStreetMap [11], [12]. Unfortunately, public OSM servers have small performance due to large amount of data and large number of requests. Also public OSM servers usage requires direct access to the Internet which reduces the usability. Therefore Social Navigator service includes its own storage of OSM maps. In this storage all maps are divided by regions and contain only routes. It reduces download's size.

The Social Navigator application collects routes from Local OSM storage and data from GeTS. The navigation algorithm works on the client-side. It reduces traffic and improves mobility. Social Navigator application also can save found obstacles to the GeTS and share route between friends through web.

C. Application architecture

The architecture of Social Navigator application is divided into several interacting parts (Fig.2).

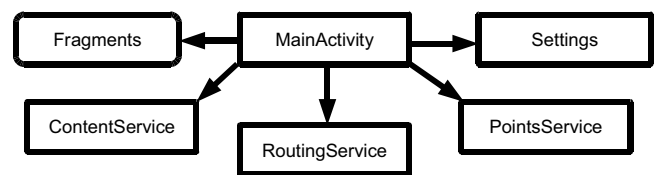


Fig.2. Social Navigator application architecture

MainActivity is a core module of application. It organizes communication between user interface, local storages and application services. Settings module stores user's settings. The User interface is represented as a set of fragments: responsible for displaying the map (MapFragment), browsing the nearest social facilities (PointsFragment), displaying the information about the objects (DetailFragment), route planning (RouteFragment) and audio support while moving (AudioFragment).

The ContentService module has a generic implementation and is used in various projects. It provides functionality for using maps:

- download list of available maps;
- download selected map fragments;
- provide objects to show map in the user interface.

The PointsService module provides access to the obstacles and categories from GeTS server. The architecture of PointsService module is presented on Fig.3.

The PointsProvider class implements interface to access third-party location-based services. It provides an opportunity to expand list of all used data sources. Currently SocialNavigator application supports GeTS service through implementation in GetsProvider class.

The GetsProvider class sends HTTP requests to the server and parses the response using GetsResponse class. There are two objects that could be obtained from GeTS server: a list of points and a list of categories. The object with points is parsed in KML class and object with categories is parsed in CategoriesContent class.

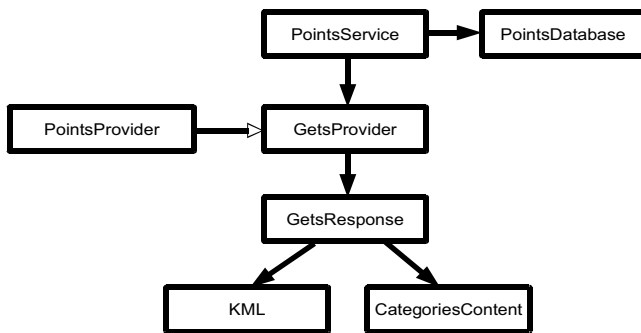


Fig.3. PointsService module architecture

All obtained points and categories are stored into the local SQLite storage through PointDatabase. It uses transactions for collecting points from several requests.

The RoutingService module provides access to the GraphHopper library [13]. It uses route graph from ContentService module and points from PointsService module to create list of routes. The architecture of RoutingService module is presented on Fig.4.

The Routing class is an interface of GraphHopper library. It is used for unified access from the application. Also RoutingService module contains CustomGraphHopper class which adds fixes to the GraphHopper library.

The BlockedEdge and PlockingWeighting classes are used to configure points and routes for GraphHopper library. It detects obstacle's difficulty level and calculates the weight of the route.

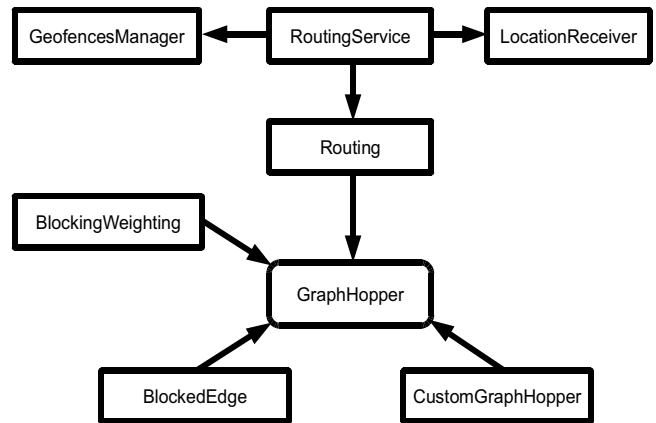


Fig.4. RoutingService module architecture

The GeofencesManager class detects user's movement and organizes route recalculation if the user deviates from the route. Also GeofencesManager class informs AudioFragment about the nearest obstacles or direction changes.

LocationReceiver class detects user coordinates using various methods and chooses a more precise location.

D. Algorithms

The common use case is presented in Fig.5. The user opens the application on device. After the start application initiates cyclic requests of coordinates from device.

Upon the receipt of user coordinates from multiple sources the application must determine the most appropriate source of origin. Application requires ability to accept or reject the new location.

We use the following algorithm. Let's assume that new location is stored in the variable newLocation and the previous location is stored in the variable oldLocation.

- 1) The device presents a new structure with location coordinates and application stores it in the newLocation variable.
- 2) If the time of the receipt newLocation is before than time of receipt oldLocation then newLocation is ignored.

- 3) If the difference between time of receipt newLocation and oldLocation is more than defined constant, then application considers that user switched off and switched on the application and newLocation is accepted.
- 4) If accuracy of newLocation is worse than accuracy of oldLocation then newLocation is ignore.
- 5) In other cases newLocation is accepted.

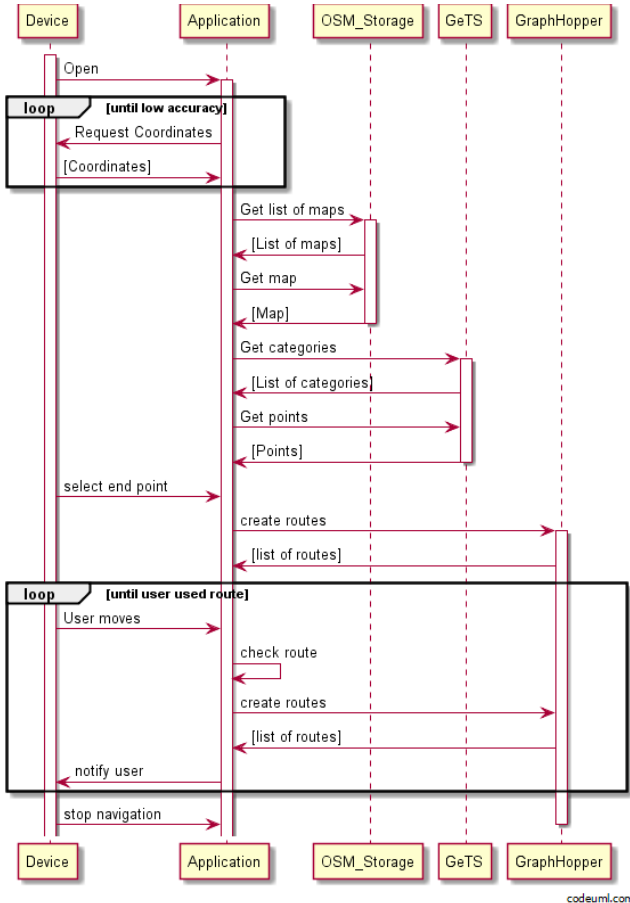


Fig.5. Common navigation use case

When accuracy reaches optimal value the application requests the List of maps from OSM storage. If the user selects the required map or there are no downloaded maps for the current location then the application sends request to download the required map. Also application requests a list of categories and a list of points for the current location. This information is stored in the local database.

When the user selects the end point the application defines relations between points and routes using Quadtree algorithm [14]. After that points are summed and the difficulty level is assigned to each route. The application runs GraphHopper library and receives list of routes: easy (total difficult equals 0), half-difficult and short-and-

difficult. By default the application selects easy or half-difficult, if the easy way is not available.

The navigation process starts from the route creation and ends by user's request. During the navigation the device sends its current location to the application, if it changes. The application checks new location and if the user leaves the route, then route construction starts again.

During the navigation the application notifies the user about the nearest obstacles and direction changes.

E. Mathematical approach to route estimation

For route building issue the Dijkstra method implemented in GraphHopper library was used. This method is a classical approach to finding of shortest way on the graph. But to use the method taking into account the obstacles and individual restrictions of the users we should estimate the accessibility of the parts of way. As we mentioned earlier the routes built for disabled people can be different with the routes provided by the standard decisions where all parts of the way meant as wholly accessible. The "Social Navigator" service uses accessibility estimation algorithm based on subjective user assessments extended to a mathematical model for evaluation of routes accessibility for each category of people with disabilities.

Let R – a set of not coincident routes connecting two geographical points A and B. The set R is defined on a graph $G(V, E)$, where V – a set of nodes (geographical points), E – the set of edges connecting the nodes. The route $r \in R$ contains m edges with corresponding distances l_k between graph nodes, $k = 1, 2, \dots, m$.

Let's denote a weight of edge k for category i as:

$$r_k^i = \sum_{j=1}^{N_k} \omega_{ij} p_{kj}^i,$$

where p_{kj}^i – an assessment of obstacle j of the edge k for category i of disabled person, ω_{ij} – the weight of obstacle j for category i , N_k – the number of obstacles on the edge k , $j = 1, \dots, N_k$, $k = 1, \dots, m$, $i = 1, \dots, n$. In current version we use the ten-level scale to evaluate ω_{ij} - the accessibility of the obstacles.

To present the obtained routes to users we calculate the accessibility level of route r for category $i = 1, \dots, n$, by the formula: $l_i(r) = \sum_{k=1}^m r_k^i$. We take into account the obtained level to inform user on the route which can be more appropriate to him. The user can choose the more complex but short way or more simple but long. The algorithm was elaborated in "Social navigator" service.

IV. SERVICE IMPLEMENTATION

The current version of Social Navigator application prototype [15] implements navigation scenario as follows.

Fig.6 shows user interface after the end point selection. The main part of the window shows map around the selected place, selected route obstacles and the end point (star icon). In the bottom application shows found routes with statistics. The user can select the type of route (pen icon, see Fig.6), accept route for navigation (check icon) or cancel (X icon). There is no safe route in the current case and the application shows Medium route by default.

The user can see detailed info about the selected route. The Fig.7 shows a list of obstacles for the selected route with difficulty level and the user can decide which route should be used. The application does not show the nearest obstacles because there are too many of them. For example, 1 sq.km. area in Petrozavodsk has more than thousand obstacles.

When the user selects the route, the application starts the navigation process. The Fig.8 shows the navigation interface.

The top panel shows the following nearest obstacle and route direction with distance. Also the user can see the map of current region and stop navigation process.

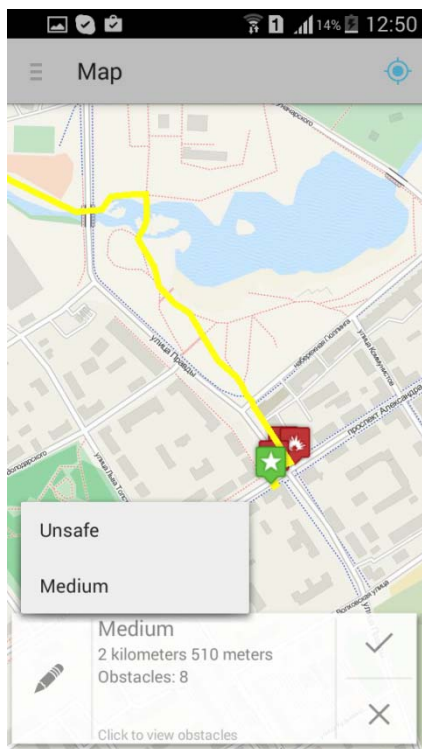


Fig.6. MapFragment interface with end point and route type selection

V. CONCLUSION

The presented work is a part of developed infrastructure that contains several mobile and web services. The “Social Navigator” is a key service of the infrastructure. The service aggregates the approaches, technologies, modules

and data investigated and elaborated during the project [1] implementation. The following mathematical and technological approaches were studied and applied on the “Social navigator” service.

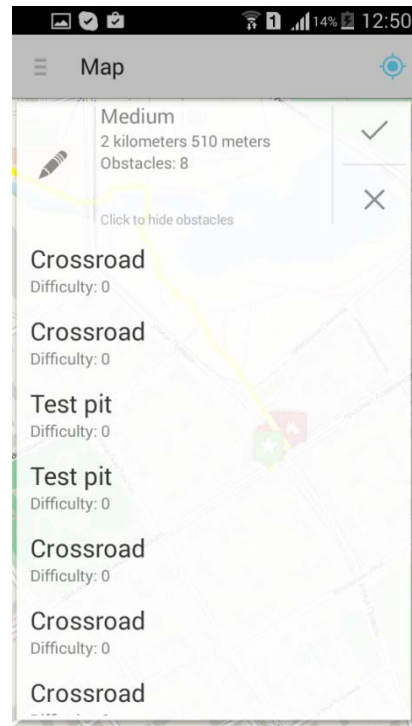


Fig.7. Obstacles list for selected route

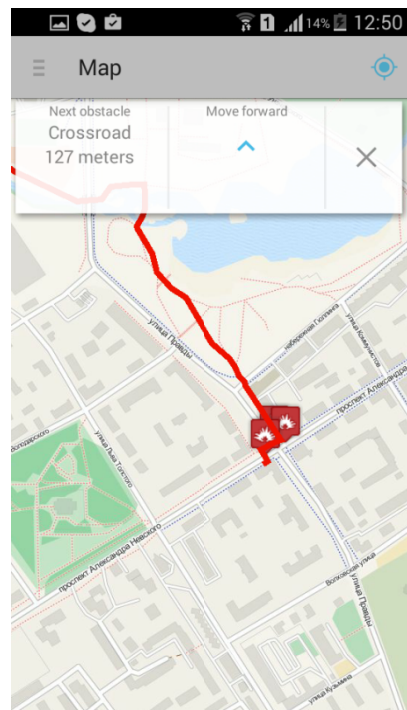


Fig.8. Navigation interface

The approach to service architecture development allows to utilize the advantages of the information infrastructure, operating the data of different sources and using of elaborated modules in other infrastructure services.

The up-to-date libraries and technologies of navigation and route planning solutions were studied. The Open Source Routing Machine (OSRM) as a high-performance routing engine for the shortest paths in road networks was applied in the development. The route building issue was formulated as the task of searching obstacle-avoiding way and connected methods and algorithms were studied. In “Social navigator” service the routes are described as a graph with weighted edges. The weight is the accessibility rate of path parts. The mathematical method for estimation of routes accessibility is based on user assessments and implemented in the service. For route building issue the Dijkstra method implemented in GraphHopper library was used. Also the technology of interaction between existed OpenStreetMap database and the data gathered by users and local authorities was developed.

As a result of the work the “Social Navigator” service was developed based on the route planning services approaches. Now the “Social navigator” service is provided to volunteers and disabled people. Created under the project [1] community of developers, community-based organizations and authorities will support the actuality of the data of the created information infrastructure.

ACKNOWLEDGMENT

This research is a part of grant KA432 “Journey planner service for disabled people (Social Navigator)” of Karelia ENPI programme, which is co-funded by the European Union, the Russian Federation and the Republic of Finland.

The article was published with financial support of the Strategic Development Program of Petrozavodsk State University.

REFERENCES

- [1] Journey planner service for disabled people (Social navigator) (KA432). Web: <https://www.kareliaenpi.eu/en/themes/332>
- [2] K. Kulakov, Y. Apanasik, A. Shabaev, I. Shabalina “Accessibility Map” and “Social Navigator” Services for People with Disabilities, *Proc. of the 15th Conference of Finnish-Russian University Cooperation in Telecommunications Program*. Spb.: SUAI, 2014, pp. 69-76.
- [3] Y. Apanasik, I. Shabalina, A. Shabaev “Accessibility Passports” Service for Information Social Infrastructure”, *Proc. of the 14th Conference of Finnish-Russian University Cooperation in Telecommunications Program*. Spb.: SUAI, 2013, pp. 3-8.
- [4] The Google Directions API. Web: <https://developers.google.com/maps/documentation/directions>
- [5] The Yandex.Maps API. Web: <https://tech.yandex.ru/maps>
- [6] Z. Li, D.S. Meek, D.J. Walton A smooth, obstacle-avoiding curve. *Computers & Graphics*, 30, 2006, p.p.581–587.
- [7] Routing/online routers – OpenStreetMap Wiki. Web: http://wiki.openstreetmap.org/wiki/Routing/online_routers
- [8] V. Tereschenko, D. Yanchik, D. Pustovoitov. The optimal way searching task on obstacles multiplicity. Web: <http://www.graphicon.ru/proceedings/2010/conference/RU/Se6/34.pdf>
- [9] Geo2tag: Open Source LBS Platform. Web: <http://geo2tag.org>
- [10] M. Zaslavskiy, K. Krinkin “Geo2Tag Performance Evaluation”, *Proceedings of 12th Conference of Open Innovations Association FRUCT*, SUAI, 2012, pp. 185-193.
- [11] OpenStreetMap. Web: <http://www.openstreetmap.org/>
- [12] Osmdroid – OpenStreetMap-Tools for Android. Web: <http://code.google.com/p/osmdroid/>
- [13] GraphHopper Road Routing in Java with OpenStreetMaps. Web: <http://graphhopper.com/>
- [14] Quadtree algorithms. Web: <https://github.com/varunpant/Quadtree>.
- [15] Social-navigator. Web: <https://github.com/oss-fruct-org/social-navigator>