

Outdoor Navigation Systems to Promote Urban Mobility to Aid Visually Impaired People

André Lima ¹, Daniela Mendes ¹, Sara Paiva ^{1,2*}

¹ Instituto Politécnico de Viana do Castelo, PORTUGAL

² ARCADigiT - Applied Research Center for Digital Transformation, PORTUGAL

*Corresponding Author: sara.paiva@estg.ipvc.pt

Citation: Lima, A., Mendes, D. and Paiva, S. (2018). Outdoor Navigation Systems to Promote Urban Mobility to Aid Visually Impaired People. *Journal of Information Systems Engineering & Management*, 3(2), 14. <https://doi.org/10.20897/jisem.201814>

Published: April 07, 2018

ABSTRACT

Promoting social inclusion is a field of active research and an emergent topic relates to blind and visually impaired people. Among the difficulties they face, we address two which are related to the use of public transports and performing pedestrian routes between points of reference. Regarding the use of public transports, we address the difficulty blind people feel of knowing where they are along the way preventing them to ring the bell for the driver to stop, in an independent way. Regarding pedestrian walks, we developed a mobile application that allows users to walk to reference points such as the City Hall, Finances, Health Centre, etc. In this paper, we present a case study that targets the two mentioned problems, developed in the Historical Centre of Viana do Castelo, a city on the north of Portugal, made in cooperation with a Visually Impaired Association.

Keywords: social inclusion, blind and visually impaired people, mobile solution, urban transport, pedestrian walks

INTRODUCTION

Mobile solutions are commonly used today for several purposes and domains. The evolution we have been assisting in these last few years regarding technology, processing power, supported sensors and functionalities made possible the development of a large range of different applications targeting different domains such as tourism, health and care, businesses, transportations, etc. In fact, mobile solutions can help us in our daily lives simplifying our routines. The relevance and utility of mobile application largely increase when we refer to disabled people, such as visually impaired people (VIP), to whom the benefits of mobile applications are greater than to the rest of the population. The limitations these people have prevents them, plenty of times, from walking alone in the streets which is one of the biggest problems VIP face: mobility. They can only go to places they previously learned with a sighted guide which means they can't handle simple tasks available to everyone else without any disability such as going to the City Hall or the Supermarket. Getting lost is a huge problem and asking a person who is nearby can sometimes be the only solution but of course there are no guarantees there will always be someone nearby. Another limitation has to do with public urban transportation, a service most cities provide to their citizens to allow them to reach a given part of the city without using their own car. This service is also very relevant for VIP but using these transportations is not always easy and problems come up such as knowing where they are and when to leave the bus. Again, the solution most of the times is to ask someone who is on the bus. Some of the most adequate technologies to systems to help solve these issues are GPS or RFID, but there are definitely some trade-offs in both. GPS cannot guarantee an accurate precision and can fail in routes between high buildings. On the other hand, a mobile solution can be relatively cheap when comparing to a RFID based-system that requires

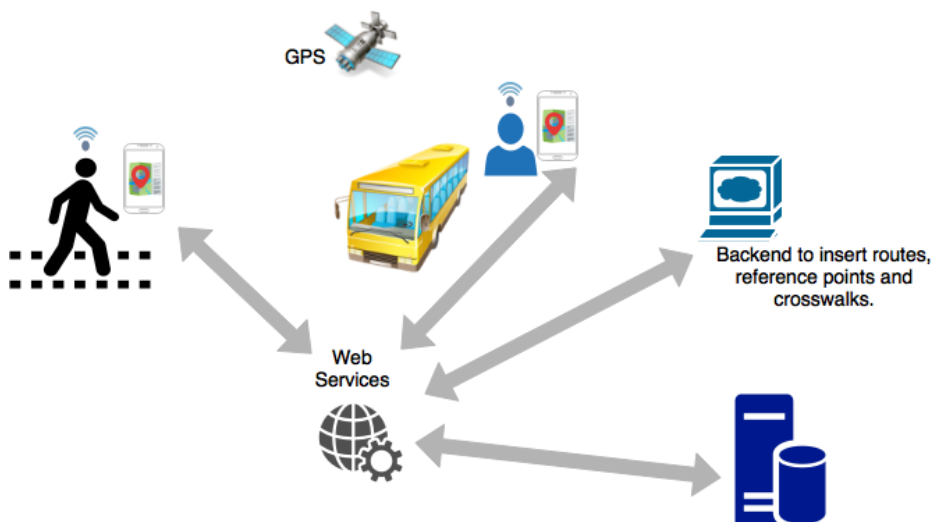


Figure 1. Proposed system architecture

the streets to be prepared with tags in the sidewalks. The benefit of this solution is that centimeters precision can be achieved.

In this paper, we present the first version of a platform to help VIP, developed in the scope of a case study in the Historical Centre of Viana do Castelo, a city in Portugal (Lima et al., 2017). The project had two main purposes: a mobile solution to help VIP to perform walks between strategic points of the city and another mobile solution to help VIP to use public urban transports.

LITERATURE REVIEW

Because of the growing research and development for disabled people, several platforms and systems have emerged in the last few years. One of such systems is detailed in (Stepnowski et al., 2011) where the authors present a novel prototype application of a system supporting street navigation and independent, outdoor movement of the blind. Nandish et. al present in (Nandish et al., 2014) a research of a navigation system for blind people in order to provide more precise location information. To identify the position and orientation and location of the blind person the authors rely on Global Positioning System (GPS) technology, TTS (Text-to-Speech) program and Google Maps APIs to provide navigation with voices. Another solution of a navigation scheme has been proposed in (Sohrawordi et al., 2015) where the authors materialized a solution for the blind and low-vision people in order to provide precise location information using Android smartphone. Another solution is presented by Dornhofer et al. (Dornhofer et al., 2014) is motivated by the fact that affordable technologies are not accurate enough to navigate blind persons on a safe trip. The authors defend positioning should be improved by telling the user the surrounding environment. They present a comparison between three different tools to route people (PgRouting, OpenTripPlanner and OpenSourceRoutingMachine). Finally, they present a prototype for Android to route blind people to a given destination with the following functionalities: allow the user to explore the whole trip on the screen, provide turn instruction by turn instruction, periodically speak the distance to the next crossing point. Yet another proposal is discussed in (Digole and Kulkarni, 2015) where the authors introduce a system that provides indoor navigation by using Radio Frequency Identifier (RFID), outdoor navigation by using Global Position System (GPS) as well as obstacle detection by using ultrasonic sensor. User will give the starting and ending location then this system will give voice instruction to reach at destination by detecting obstacle also. This system can specially use in big campus like industries, big institutes where it will act as guiding map.

PROPOSED SOLUTION

The architecture for the proposed solution for helping blind and visually impaired to use public transportation and to be more independent when walking alone is presented in [Figure 1](#).

The system is composed of several subsystems to be able to accomplish the two main goals we have defined: help blind pedestrians and blind people using a bus. The mobile user smartphone can either run Android or iOS operating system. The only requirement is that it has GPS. The user is always a blind or visually impaired person that can be a pedestrian when trying to walk between reference points in the historical center of Viana do Castelo; or by bus trying to get a given destination. The technological component of the architecture includes a set of REST

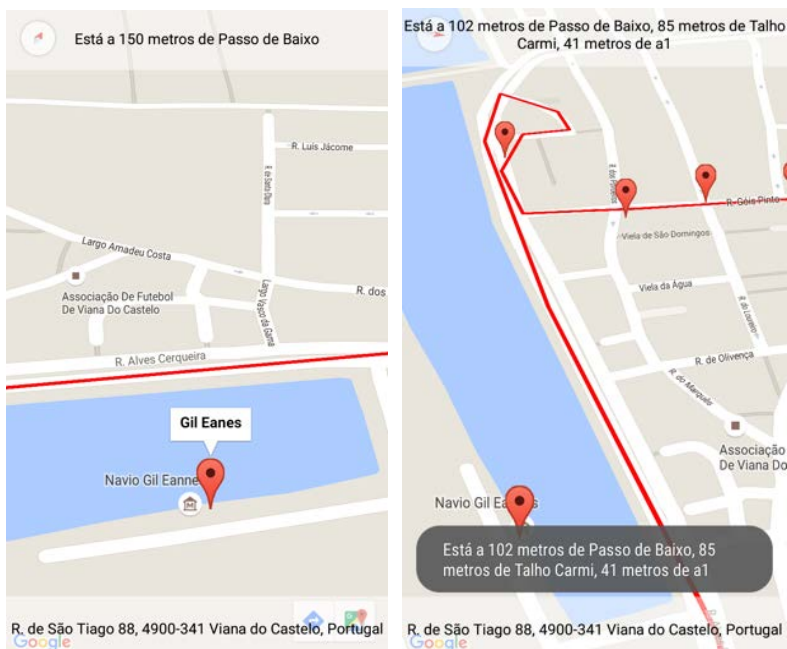


Figure 2. Mobile application for helping blind pedestrian when using public transports

JSON PHP web services that are accessed via the mobile apps, and that make the communication to a remote MYSQL database that mainly have information about routes (pedestrian or by bus), reference points and crosswalks. All this information is managed by a backend.

Backend

The backend was developed in PHP and has three main functionalities: manage (add, update, delete) a route (pedestrian or bus), manage reference points and manage crosswalks. The information managed by the backend is used by both mobile apps: the one that helps users during walking tours and in bus trips.

Mobile Applications

As afore mentioned, we have developed two mobile applications. The first is a mobile application that helps blind users during their walks between reference points in the city of Viana do Castelo. For these developments, we had the collaboration of a Blind People Association (BPA) in Viana do Castelo that explained to us the needs they have in their daily routines and made with us plenty of field tests that allowed to test and improve the applications. What happens with most BPA associates, is that they are explained a route by the BPA technicians, such as the route that allows them to get to the association from their house; or from the association to some major reference points (City Hall, Shopping Centre, Bus Station, Finances, etc.). The application would allow them to be more confident when performing these routes alone because they would have an extra help and guidance along the way. The other need the BPA transmitted us, had to do with the using public transportation alone and the simple fact that they have difficulty knowing when to ring the bell for the driver to stop because they easily lose track of where they are. The application purpose would be to inform them where they are along all route, so they can be more autonomous and do not depend on other people.

Layout Concerns

Before specifically addressing the mobile applications, it is important to first mention the concerns in the layout creation. Creating an application for a VIP is very different from creating an application for a person without visual difficulties. The VIPs that accompanied this work mentioned there are a lot of visual diseases. And while some people can see a map other rely completely on voice commands only. So, we decided not to exclude the map view from the apps. The strongest points of the app, however, are the voice commands guaranteed either with TalkBack (Google, 2016) or VoiceOver (Apple, 2016). Strong colors were also something to avoid as some VIP have trouble visualizing them.

Mobile App–use Public Transports

This application relies on several information managed by the backend. Namely, when it is opened, it checks if new or updates to existent routes/reference points exist and, if so, downloads them. The user is prompted with a simple interface that shows a map and informs the user where he is (name of the street) and of the next reference points he will find, as shown in [Figure 2](#). The application uses GPS signal that is received every 5 seconds and

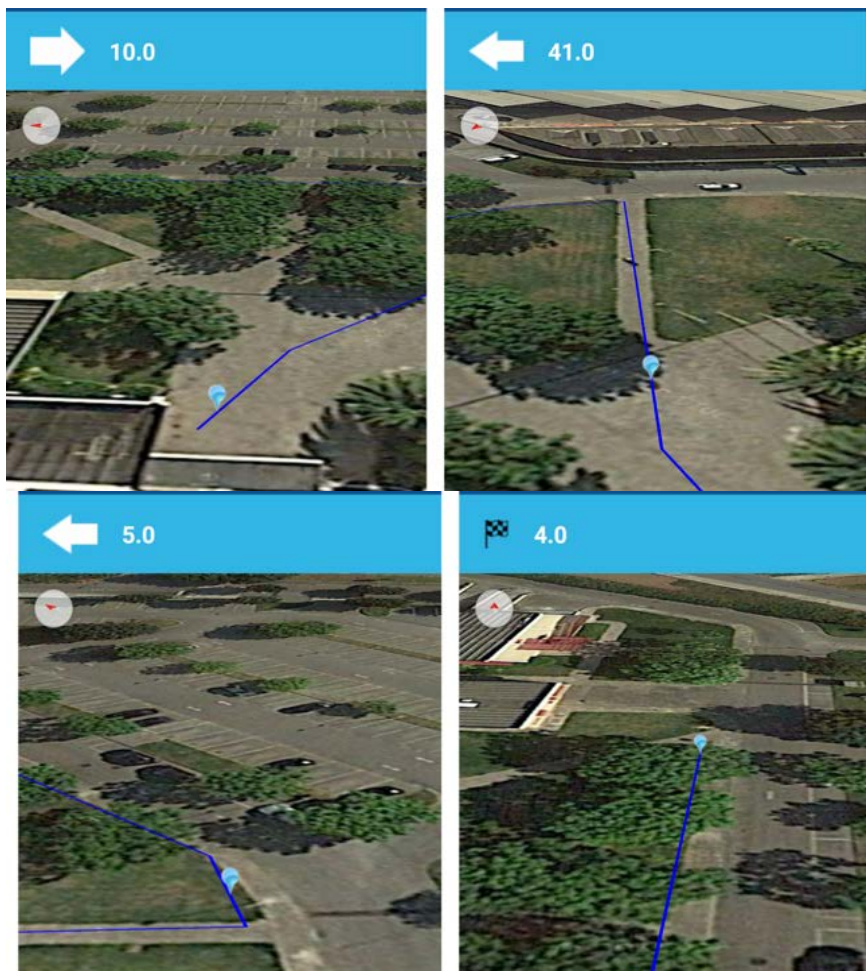


Figure 3. Final layouts to route blind people

checks the distance to the next reference point. When it is near 150 meters we show the message: “You are close to <reference point name>”. The value of 150 meters was what we thought was appropriate, with the field tests, so there is time to say the message before the user reaches the reference point, considering the bus is in motion. The BPA associates have not suggested any further changes for now as the application already helps them in the use of the bus.

Mobile App–pedestrian Walks

This application is much more defying than the previous one as it depends on GPS and there are places with bad reception. This is the main issue we knew we would face. Our goal is to have an app where VIP can choose the destination they want and be guided there, per a pre-configured route. Next, we start by presenting the final layout and functionalities of the developed application. Next, we detail the algorithm and all calculations needed to achieve the intended behaviour.

Application layouts

In **Figure 3** we present four scenarios that present the user current location in a route and the number of meters to the next turn or to reach the destination: a) in ten meters there will be a turn to the right; b) in 41 meters there will be a turn to the left; c) in 5 meters there will be a turn to the left; d) in 4 meters you will reach your destination.

Algorithm

To accomplish the behaviour explained in the previous section, we performed most calculations whenever a new location is received.

The implemented algorithm is presented in **Figure 4**. We defined three constants:

- **MAX_METERS**: represents the maximum distance between a received location and the route for the application to consider the user is on the route. We need to do this because there is always an error with the GPS location received and will never (or with a very low probability) receive a coordinate exactly on the route. We have currently set this value to 3 meters.

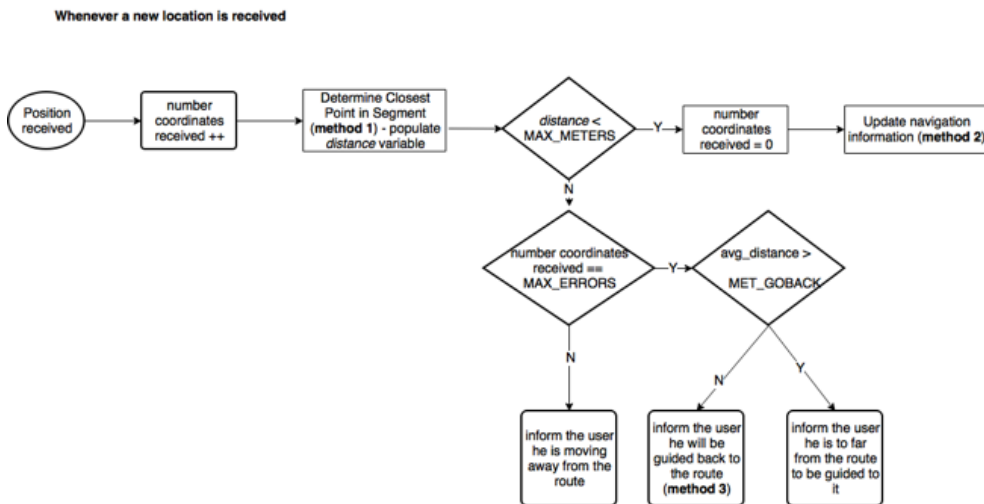


Figure 4. Algorithm implemented whenever a new location is received

- MET_GOBACK: represents the distance between the user's position and the route where we inform the user that he is too far away from the route for us to guide him back safely. We have currently set this value to 6 meters.
- MAX_ERRORS: represents the number of locations received for us to consider the user if off track and needs to be guided back. (if possible). We have currently set this value to 4.

When a new position is received, we start by incrementing by one the variable *number_times_received_coordinates* (we will use it ahead). Next, we calculate, for the received location, the distance to the route and for that we use *method1 – determineClosestPointInSegment*. This method will be explained later with more detail, but for now it is enough to mention it populates the variable *distance* with the shortest distance there is from the received point to the segments of the route. If that distance is inferior to MAX_METERS, the user is on the route and the navigation will start: we update navigation that includes rotating the map in the correct direction and calculate distance to the next turn. This is done in *method 2 – updateNavigationInfo*. If not (then the distance is greater than MAX_METERS), and if the number of received coordinates is equal to MAX_ERRORS, then we consider having a considerable stable position as we have already received MAX_ERRORS GPS positions and none was on the route. At this point, we test the average distance (considering the several positions received) with the constant MET_GOBACK. If the distance is greater, than we consider the user is very far from the route and we choose to inform the user he is too far, and he should get help, so we avoid dangerous situations. If the distance is inferior to MET_REC_ROUTE, then we are in conditions to guide him back to the route (*method 3 – getBackOnRoute* that will be explained later). Finally, if the distance is greater than MAX_METERS and if the number of received coordinates is not equal to MAX_ERRORS, we simply notify the user he is moving away from the route.

Method1 – determineClosestPointInSegment

This is the first method executed whenever a location is received. This method is divided in two parts: 1) find the segment closest to the current location; and 2) find the closest point in the obtained segment that is closest to the current location. Figure 5 shows a concrete situation of what we wish to achieve. Considering a route with several segments and considering point C is the received location, we first need to find the closest segment which in this case would be segment \overline{AB} . Next, we need to determine point P. This is necessary because we need to correct the received GPS position, so the navigation is fluent and the user location is shown always on the route (of course considering the received GPS location is less than MAX_METERS).

Find the segment closest to the current location

To find the closest segment to a given point C we process each segment and perform the following reasoning. We start by filling an array with intermediary point within the segment, such as shown in Figure 6. We start by calculating the line equation and then find multiple points within the segment. We measure the distance to each one of those points to point C. The closest segment is the one that has a point with the minimum distance to C.

Find the closest point in the obtained segment that is closest to the current location

After obtaining the segment closest to C, to find point P we use:

- the equation of a straight line to determine the equation of line r, based on two points: A and B
- the equation of a straight line to determine the equation of line s, based on the gradient and one point (C)
- the intersection of both lines to determine point P
- the distance between two points to determine the distance between C and P.



Figure 5. Example of finding the point in a segment that is closest to a GPS location



Figure 6. Example of filling intermediary points to determine the closest segment to a given point



Figure 7. Angle calculation between two line segments

Method 2 – updateNavigationInfo

At the moment this method executes, the user received a location which is considered to be on the route (distance < MAX_METERS). What we do is this method is: 1) Determine the angle of the turn; 2) Calculate meters till the end of the segment; 3) Get direction of the turn (left or right); 4) Update UI with the type of turn and meters left

Determine the angle of the turn

To determine the angle between two-line segments, we use the law of cosines and the notion of supplementary angles. So, considering the image represented in **Figure 7**, and assuming the user is somewhere in the line segment \overline{AB} , we want to determine the angle θ , which is the supplementary angle of β . To determine β we use the law of

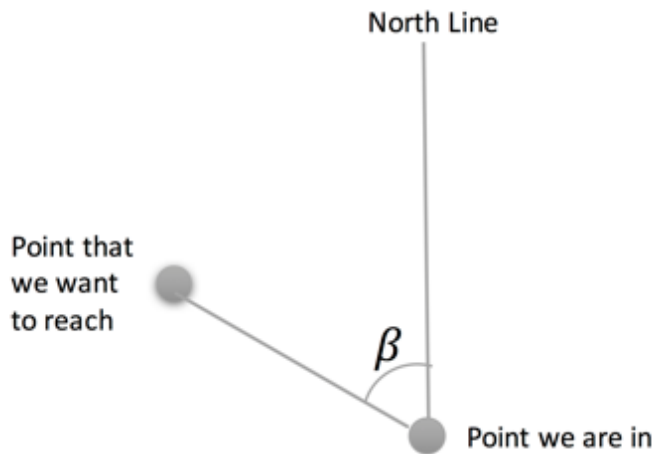


Figure 8. Calculate bearing with the north line

cosines that relates the lengths of the sides of a triangle to the cosine of one of its angles. The information we have is:

- length of \overline{AB} (c), which is given by the distance between points A and B
 - length of \overline{BC} (a), which is given by the distance between points B and C
- Considering θ is the supplementary angle of β , we have that:

$$\theta = 180 - \arccos \frac{(a^2 + c^2 - b^2)}{2ac}$$

Method 3 – getBackOnRoute

This method fires when the user is detected to be off the route but in a distance where is still safe to guide him back. Most of the times the user will not go very far; he will probably walk in circles when he detects he is moving away from the right route. What we want is to drive him to the last point of the route he was in but for that we first need to put him in the right direction, and for that we need to calculate the angle as shown in **Figure 8**.

Detection of angle

To calculate the angle mentioned in **Figure 8**, we must calculate the bearing. The bearing gives us the angle between the north line of earth and the line connecting the target (route) and the reference (us). So, to know our direction, we must calculate the bearing, adjust it with the heading, and we will have the angle the user must turn to be in the right direction to go to his destination. The information we need to calculate the angle is: θ : the latitude, λ : the longitude, δ : the difference between two values, β : the bearing, A: our point of reference, B: our point of target

The formula to calculate the angle is:

$$\beta = \text{atan2}(X, Y)$$

where X and Y are calculated as

$$X = \cos(\theta b) * \sin(\delta \lambda)$$

$$Y = (\cos(\theta a) * \sin(\theta b)) - (\sin(\theta a) * \cos(\theta b) * \cos(\delta \lambda))$$

and $\delta \lambda = \lambda b - \lambda a$.

From there, we will have an angle in radians. To get a value in degrees, we should multiply it by $180/\Pi$. After putting it in degrees, we will have two cases: β is negative or positive and we should get the “true β ” to match the heading values the device gives us (start at 0, north, and move toward the right to end up in 360, north again).

- β is negative: $\beta = -1 * \beta$
- β is positive: $\beta = 360 - \beta$

EVALUATION

After the solution was defined, we carried out field tests with some visually impaired users, associates of the BPA. We had some troubles in some places of the route where the GPS signal was not very good. To solve this, in the future we will define routes that go through streets that are not very narrow, so we can have a good reception, which should be possible as we have a considerable number of street alternatives.

The map in the interface can be relevant for some users but when the weather is very sunny, it is almost impossible to visualize the map and the sound support gains more importance. Besides this, we will add vibration when a new information to the user exists (a new turn for example). The degree of the turn should also be more precise. Finally, we realized routes should be drawn in the backend with straight lines that represent the major segments of the route and avoid little segments.

CONCLUSIONS

In this paper, we presented two mobile applications to help visually impaired people to walk around in a city, in pre-defined routes, between strategic reference points such as the City Hall, the Hospital or Finances; and another one to help them use public transports in a more autonomous way.

We presented the system's architecture and the navigation system designed for the app to help in pedestrian walks. The algorithm includes support for situations where the user distances himself too much from the route and needs to be guided back. The application was tested in the field with several visually impaired members of an association from Viana do Castelo city. The feedback was very positive although some adjustments will be necessary before production such as avoiding narrow streets where the GPS signal is weak, improve feedback in turns introducing vibration and more precise information of the angle of the turn.

REFERENCES

- Apple. (2016). VoiceOver app for iOS.
- Digole, R. N. and Kulkarni, P. S. M. (2015). Smart Navigation System for Visually Impaired Person. *International Journal of Advanced Research in Computer and Communication Engineering*, 4(7), 53–57.
- Dornhofer, M., Bischof, W. and Krajnc, E. (2014). Comparison of Open Source routing services with OpenStreetMap Data for blind pedestrians. *International Conference for Free and Open Source Software for Geospatial*.
- Google. (2016). TalkBack app for Android. Available at: https://play.google.com/store/apps/details?id=com.google.android.marvin.talkback&hl=pt_PT
- Lima, A., Mendes, D. and Paiva, S. (2017). Mobile solutions for visually impaired people - Case Study in Viana do Castelo Historical Center. In *Information Systems and Technologies (CISTI)*, 2017 12th Iberian Conference on (pp. 1-5). IEEE. <https://doi.org/10.23919/CISTI.2017.7975993>
- Nandish, M. S., Balaji, M. C., C. P. and P. S. (2014). An Outdoor Navigation with Voice Recognition Security Application for Visually Impaired People. *International Journal of Engineering Trends and Technology*, 10(10), 500–504. <https://doi.org/10.14445/22315381/IJETT-V10P298>
- Sohrawordi, M. et al. (2015). Android-Based Walking Assistant for Blind And Low-Vision People Suggesting The Shortest Path Using Floyd-Warshall Algorithm. *Journal of Innovation & Development Strategy*, 9(2).
- Stepnowski, A., Kamiński, Ł. and Demkowicz, J. (2011). Voice Maps—the system for navigation of blind in urban area. *Proceedings of the 10th WSEAS International Conference on Applied Computer and Applied Computational Science*. pp. 201–206.