Abstract

For a majority of blind and visually impaired persons, public transport is the only viable obility option to seek education, work and social connectivity. They face major difficulties in independently accessing public buses since they cannot read the route number and are unsure about the physical location of the bus and its entry/exit door. Despite constantly seeking help from sighted fellow travelers, blind persons frequently miss their desired bus, are unable to reach the gate and get hurt in the process; causing fear and anxiety.

This project aims to study, in a developing country context, (i) the challenges faced by blind persons in identifying and boarding public buses, (ii) the nature and effectiveness of help sought from fellow travelers and (iii) the resulting user anxiety and frustration.

Additionally, a system has been presented here which will enable a blind person to overcome all these problems and be self-dependent. An Android app, which uses the GPS co-ordinates of the Blind person and the Bus will notify the blind person. Also, voice notification will be given if the blind person is about to reach his destination.

Acknowledgement

I would like to express my gratitude towards all the professors who helped during this project. Especially, I would thank Prof.Vinay Hegde, Prof.Swarnalatha, Prof.Suma B. I would also like to thank our beloved HOD Dr. Shobha G, for her continuous support in everything. I would also like to express my gratitude towards our Principal Dr. BS Satyanarayana for his encouragement.

List of Contents

List of Figures

1. Introduction

In major cities, buses play an important role for the transportation. For a majority of blind and visually impaired persons, public transport is the only viable mobility option to seek education, work and social connectivity. Those people live in a limited environment and have difficulty to sense what happen around them, which reduces their activities in several fields, such as education and transportation since they depend only on their own intuition. In addition, as the population ages, the number of VIPs has increased. At present, statistic showed that 285 million people are visually impaired worldwide: 39 million are blind and 246 have low vision. As of India, around 8 million people in India are blind. India is now home to the world's largest number of blind people with 20% of the whole world. Hence, we need to make their lives more comfortable by introducing a system that helps them enjoy transportation services independently and freely like ordinary people, without relying on others.

Many papers have been published and much work has been done on this issue yet this project aims to present an all new and simple approach towards the design. An IEEE paper, “Bus Identification System for Visually Impaired Person,”, Next Generation Mobile Applications, Services and Technologies (NGMAST) has been referred. Many electronic devices have already been developed and implemented in some areas to assist the blind people like Sonic Guide, Mowat Sensor, Laser Cane and Navbelt. But all of them are to assist them while pedestrian crossing and there hasn’t been developed any successful device to assist them boarding a bus. This paper intends to fill that cavity.

2. Literature Survey

[1] G Lavanya M.E., Assistant Professor, Preethy. W, Shameem.A , Sushmitha.R IVyear, Biomedical Engineering, Passenger BUS Alert System for Easy Navigation of Blind , 2013 International Conference on Circuits, Power and Computing Technologies [ICCPCT2013].

[2] T.Quoc, M. Kim, H. Lee, and K. Eom, “Wireless Sensor Network apply for the Blind U-bus System,” International Journal of u- and e- Service Science and Technology. Vol. 3, No. 3, Sep. 2010.

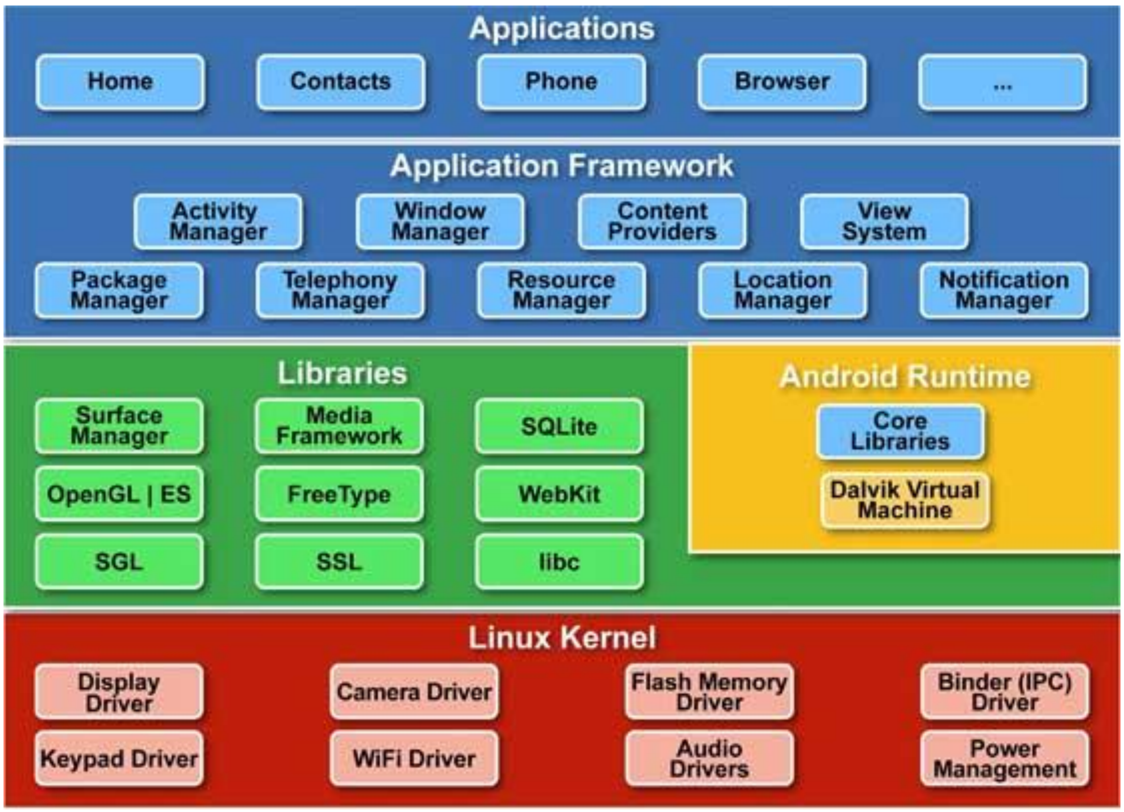
[3] M. Zikrul, I. Ismarani, and M. Saaid, “Bus Detection Device for the Blind Using RFID Application,” in the 5th International Colloquium on Signal Processing & Its Applications, CSPA, pp. 6-7 Sep. 2009.

[4] http://assistech.iitd.ernet.in/onboard.php

[5] http://link.springer.com/chapter/10.1007/978-3-642-02710-9_44

3. Android Architecture

Android operating system is a stack of software components which is roughly divided into five sections and four main layers as shown below in the architecture diagram.



3.1. Linux kernel

At the bottom of the layers is Linux - Linux 2.6 with approximately 115 patches. This provides basic system functionality like process management, memory management, device management like camera, keypad, display etc. Also, the kernel handles all the things that Linux is really good at such as networking and a vast array of device drivers, which take the pain out of interfacing to peripheral hardware.

3.2. Libraries

On top of Linux kernel there is a set of libraries including open-source Web browser engine WebKit, well known library libc, SQLite database which is a useful repository for storage and sharing of application data, libraries to play and record audio and video, SSL libraries responsible for Internet security etc.

3.3. Android Runtime

This is the third section of the architecture and available on the second layer from the bottom. This section provides a key component called Dalvik Virtual Machine which is a kind of Java Virtual Machine specially designed and optimized for Android.

The Dalvik VM makes use of Linux core features like memory management and multi-threading, which is intrinsic in the Java language. The Dalvik VM enables every Android application to run in its own process, with its own instance of the Dalvik virtual machine.

The Android runtime also provides a set of core libraries which enable Android application developers to write Android applications using standard Java programming language.

3.4. Application Framework

The Application Framework layer provides many higher-level services to applications in the form of Java classes. Application developers are allowed to make use of these services in their applications.

3.5. Applications

You will find all the Android application at the top layer. You will write your application to be installed on this layer only. Examples of such applications are Contacts Books, Browser, Games etc.

4. Dalvik Virtual Machine

Dalvik is a process virtual machine (VM) in Google's Android operating system that executes applications written for Android. This makes Dalvik an integral part of the Android software stack (in Android versions 4.4 "KitKat" and earlier) that is typically used on mobile devices such as mobile phones and tablet computers, as well as more recently on devices such as smart TVs and wearables.

Programs are commonly written in Java and compiled to bytecode for the Java virtual machine, which is then translated to Dalvik bytecode and stored in .dex (Dalvik EXecutable) and .odex (Optimized Dalvik EXecutable) files; related terms odex and de-odex are associated with respective bytecode conversions. The compact Dalvik Executable format is designed for systems that are constrained in terms of memory and processor speed.

Dalvik is open-source software. It was originally written by Dan Bornstein, who named it after the fishing village of Dalvík inEyjafjörður, Iceland.

An alternative runtime environment called Android Runtime (ART) was included in Android 4.4 "KitKat" as a technology preview.[3][4]ART replaces Dalvik entirely in Android 5.0 "Lollipop".

4.1. Architecture

Unlike Java VMs, which are stack machines, the Dalvik VM uses a register-based architecture that requires fewer, typically more complex virtual machine instructions. Dalvik programs are written in Java using the Android application programming interface (API), compiled to Java bytecode, and converted to Dalvik instructions as necessary.

A tool called dx is used to convert Java .class files into the .dex format. Multiple classes are included in a single .dex file. Duplicate strings and other constants used in multiple class files are included only once in the .dex output to conserve space. Java bytecode is also converted into an alternative instruction set used by the Dalvik VM. An uncompressed .dex file is typically a few percent smaller in size than a compressed Java archive (JAR) derived from the same .class files.

The Dalvik executables may be modified again when installed onto a mobile device. In order to gain furtheroptimizations, byte order may be swapped in certain data, simple data structures and function libraries may be linked inline, and empty class objects may be short-circuited, for example.

Being optimized for low memory requirements, Dalvik has some specific characteristics that differentiate it from other standard VMs.

- The VM was slimmed down to use less space.
- The constant pool has been modified to use only 32-bit indices to simplify the interpreter.
- Standard Java bytecode executes eight-bit stack instructions. Local variables must be copied to or from the operand stack by separate instructions. Dalvik instead uses its own 16-bit instruction set that works directly on local variables. The local variable is commonly picked by a four-bit "virtual register" field. This lowers Dalvik's instruction count and raises its interpreter speed.

According to Google, the design of Dalvik permits a device to run multiple instances of the VM efficiently.

Android 2.2 "Froyo" brought trace-based just-in-time (JIT) compilation into Dalvik, optimizing the execution of applications by continually profiling applications each time they run and dynamically compiling frequently executed short segments of their bytecode into native machine code. While Dalvik interprets the rest of application's bytecode, native execution of those short bytecode segments, called "traces", provides significant performance improvements.

4.2. Performance

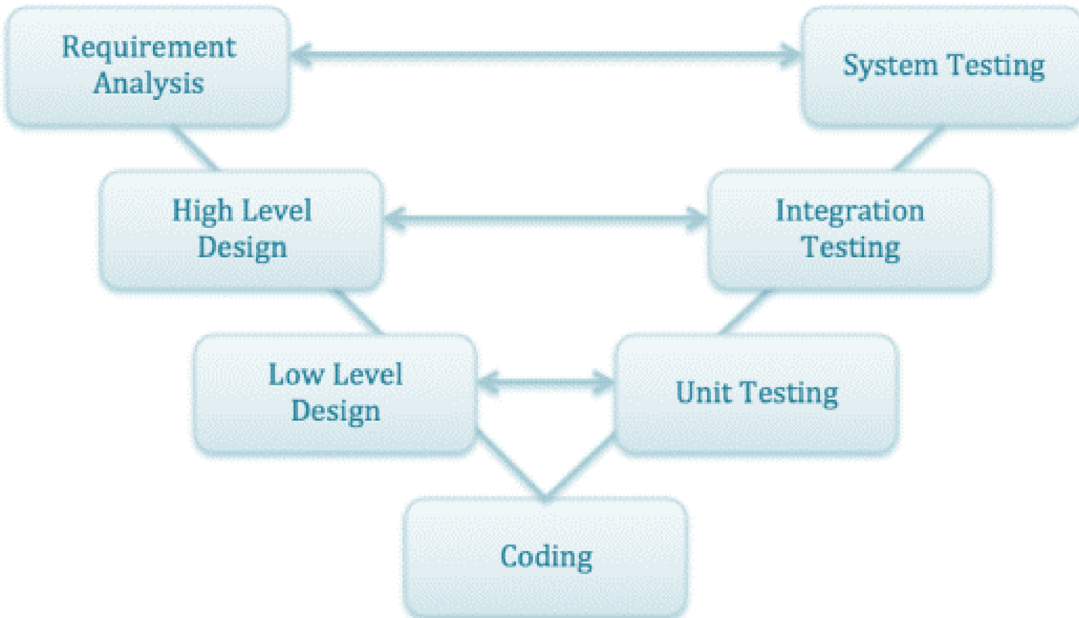
The relative merits of stack machines versus register-based approaches are a subject of ongoing debate. Generally, stack-based machines must use instructions to load data on the stack and manipulate that data, and, thus, require more instructions than register machines to implement the same high level code, but the instructions in a register machine must encode the source and destination registers and, therefore, tend to be larger. This difference is primarily of importance to VM interpreters for which opcode dispatch tends to be expensive along with other factors similarly relevant to just-in-time compilation.

However, tests performed on ARM devices in 2010 by Oracle (owner of the Java technology) with standard non-graphical Java benchmarks on Java SE embedded seemed to show it to be two to three times faster than Android 2.2, which is the initial release to include a just-in-time (JIT) compiler.

In 2012, academic benchmarks confirmed the factor of 3 between HotSpot and Dalvik on the same Android board, also noting that Dalvik code was not smaller than Hotspot.

Furthermore, benchmarks performed on Android device still show (as of March 2014) up to a factor 100 between native applications and a Dalvik application on the same Android device. Upon running benchmarks using the early interpreter of 2009, both JNI and native code showed an order of magnitude speed up.

5. Application of Engineering principles



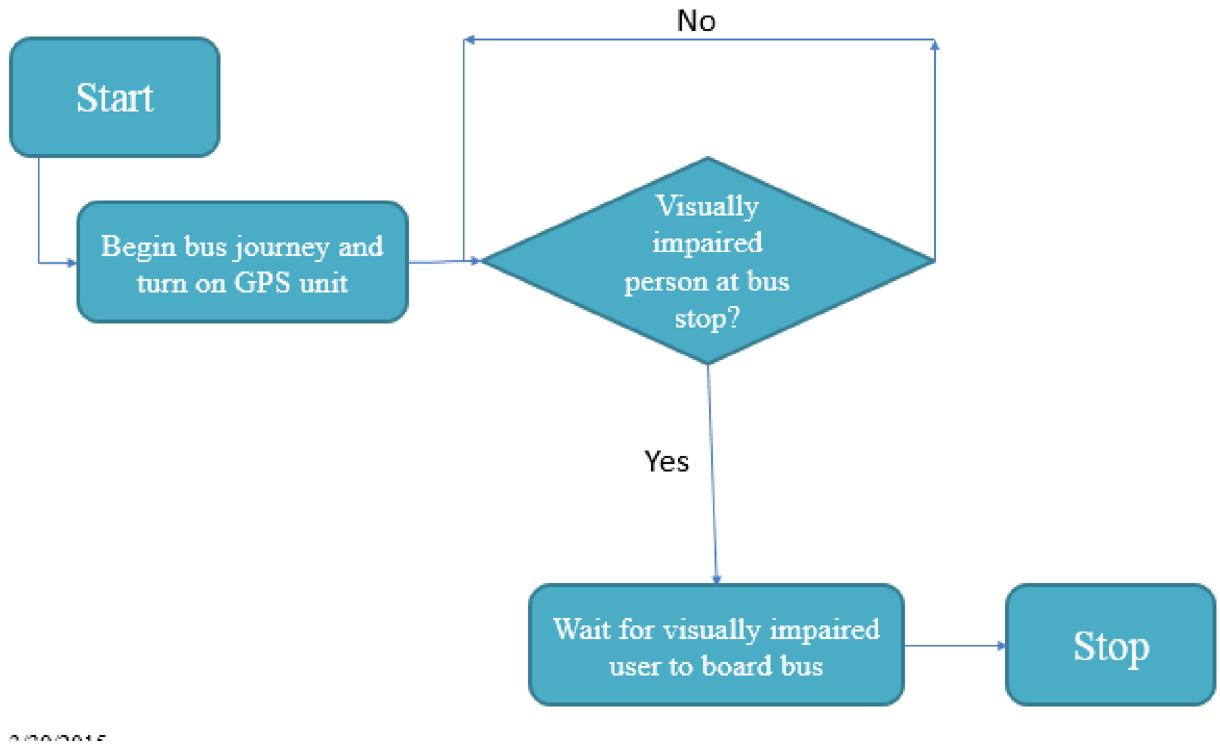
At the very beginning, a requirement analysis was done to list out all the high level requirements. Then, using the requirements, the system was designed at a high level to achieve the required functionality. Then, each component in the high level design was split into separate units, and a low level design was prepared. Now, considering all the modules, coding of each component was done. Component-wise, unit testing was done to ensure the working of individual components. Then, all the components were integrated to perform an integration testing and the System testing to ensure all the working functionalities.

6. System Design

6.1. High level design

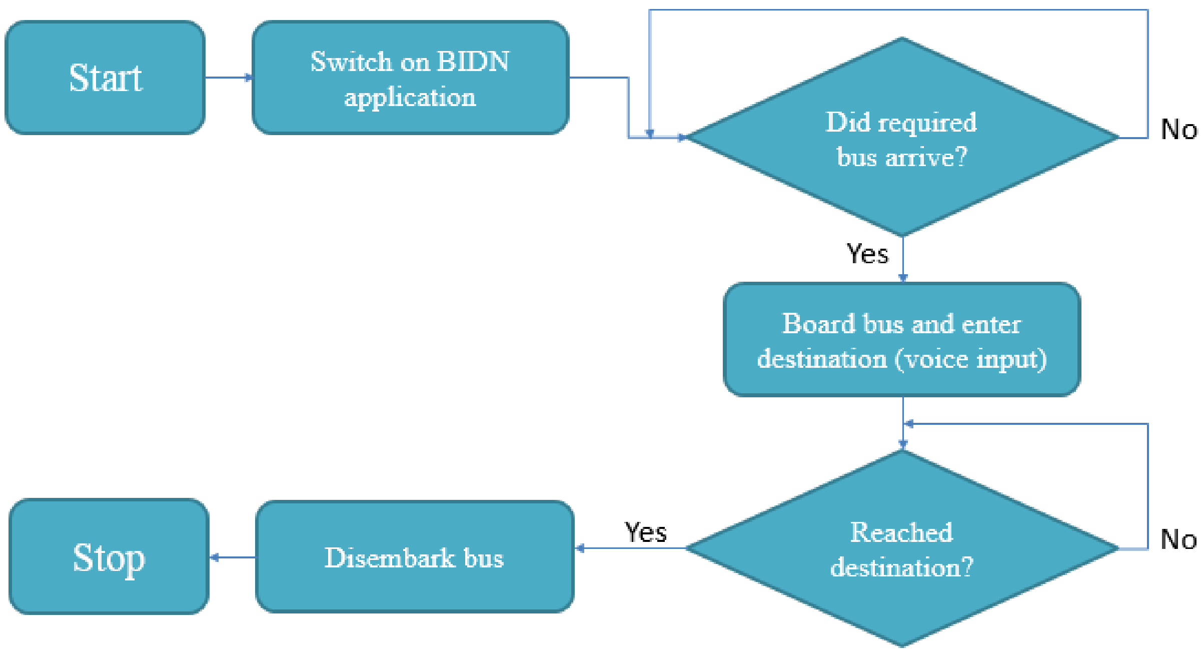
We have 2 modules here. The User module, and the Bus module. The Bus module is an android app which will be installed by the Bus driver or the conductor, which will update the instantaneous location of that particular bus. Now the User module, also an android app is used by the blind person. He needs to select the Destination. After that, he can switch off the app. Whenever a bus is heading to the person's destination and is coming through his location, it alerts the blind person through voice notification.

6.2. Functioning of the Bus module



When a bus begins its journey, it needs to turn on its bus module. Basically, the driver or the conductor should run the app for once. Then, the location co-ordinates of the bus will be constantly updated to the cloud database. Then, assume that a visually impaired person boards the bus.

6.3. Functioning of the User module



The blind person, whenever he wants to go to a place, opens this app, then selects his destination. This way, he updates his current location and his desired destination. This will be used by the cloud scripts to determine which bus is the desired bus to an user and notifies accordingly.

7. Implementation

7.1. Designing android application using Android Studio.

Android Studio is a software tool which is used to develop the Android apps, apart from Eclipse. The 2 android apps were developed keeping in mind all the android community. So, the minimum sdk version used for the app was 15, whereas the current version being 22.

7.2. Cloud Web services using Openshift

Since a cloud database was needed to store the User and the Bus co-ordinates, a cloud database such as openshift was used. Also, a cloud hosting and php runtime support was needed, which was also provided by openshift.

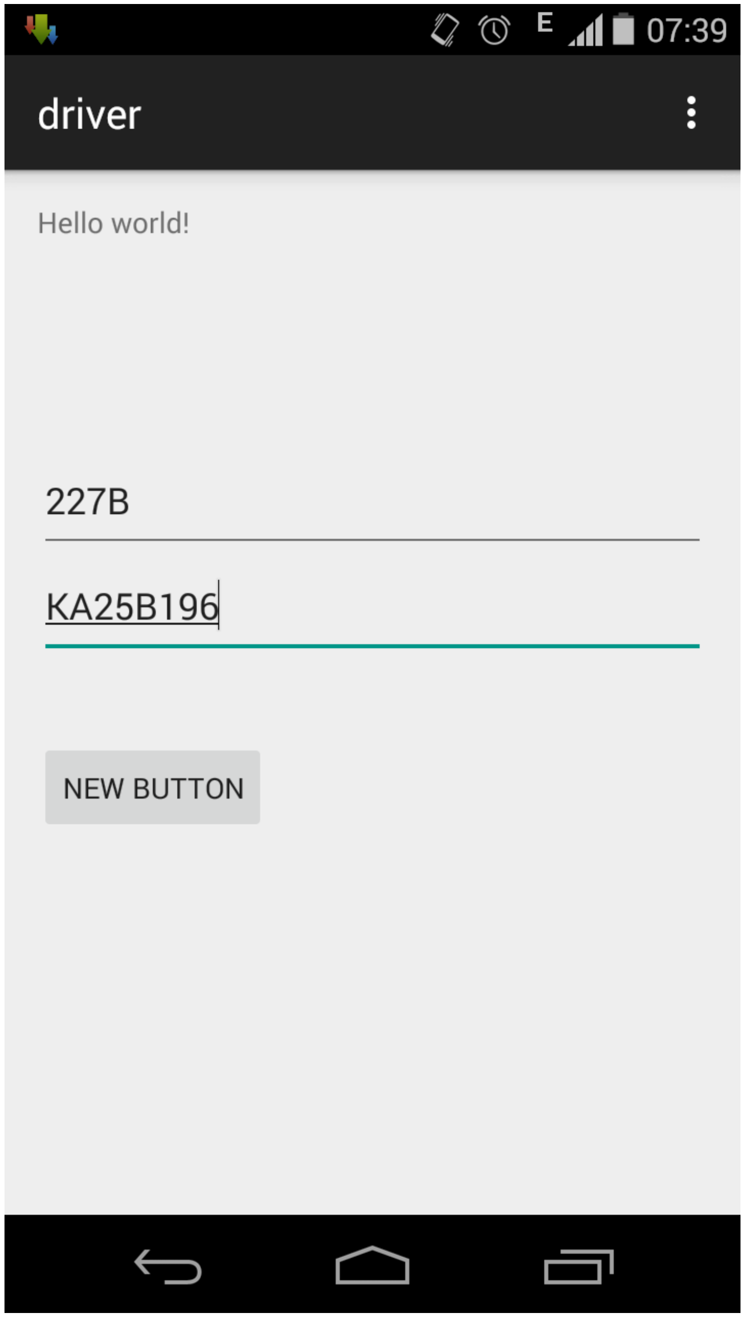
7.3. Google Cloud Messenger Service

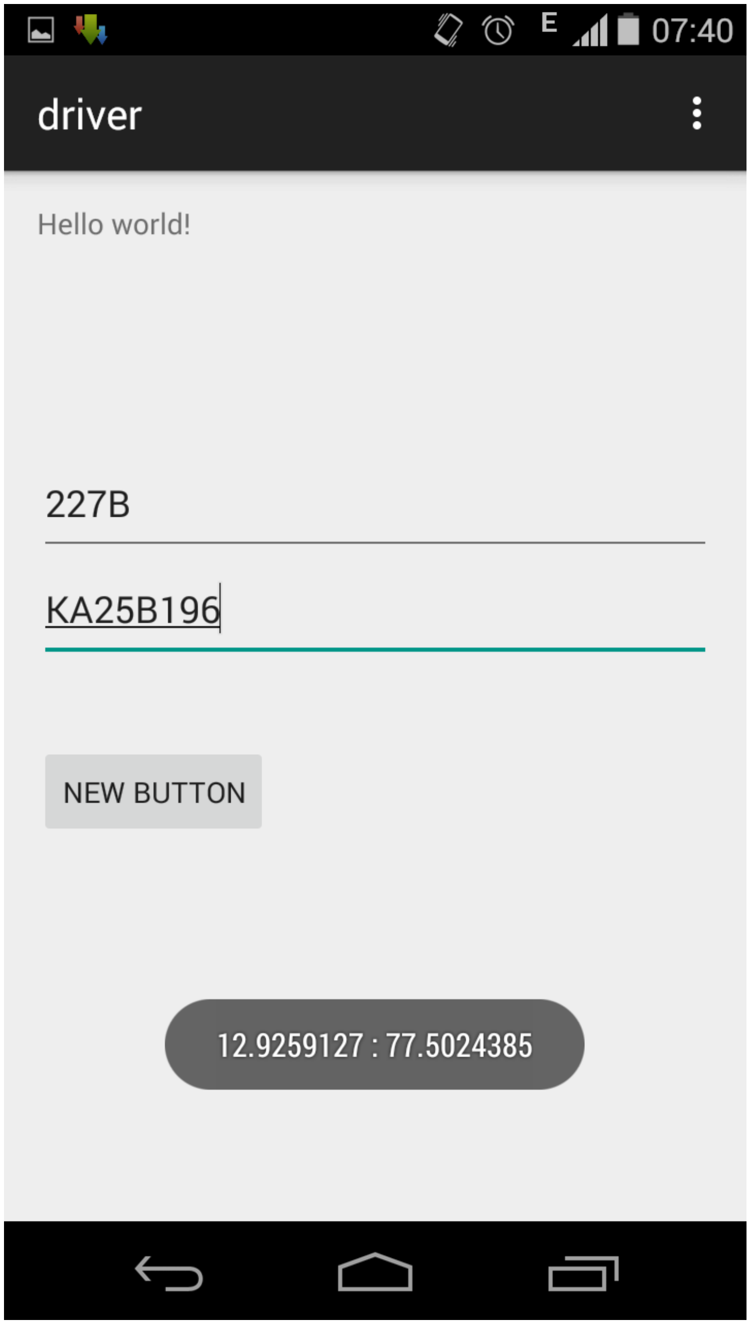
Google Cloud Messenger (GCM) is a very popular service for sending messages to an android phone, uniquely recognized by a GCM id. We use this service to send notifications to the Blind person.

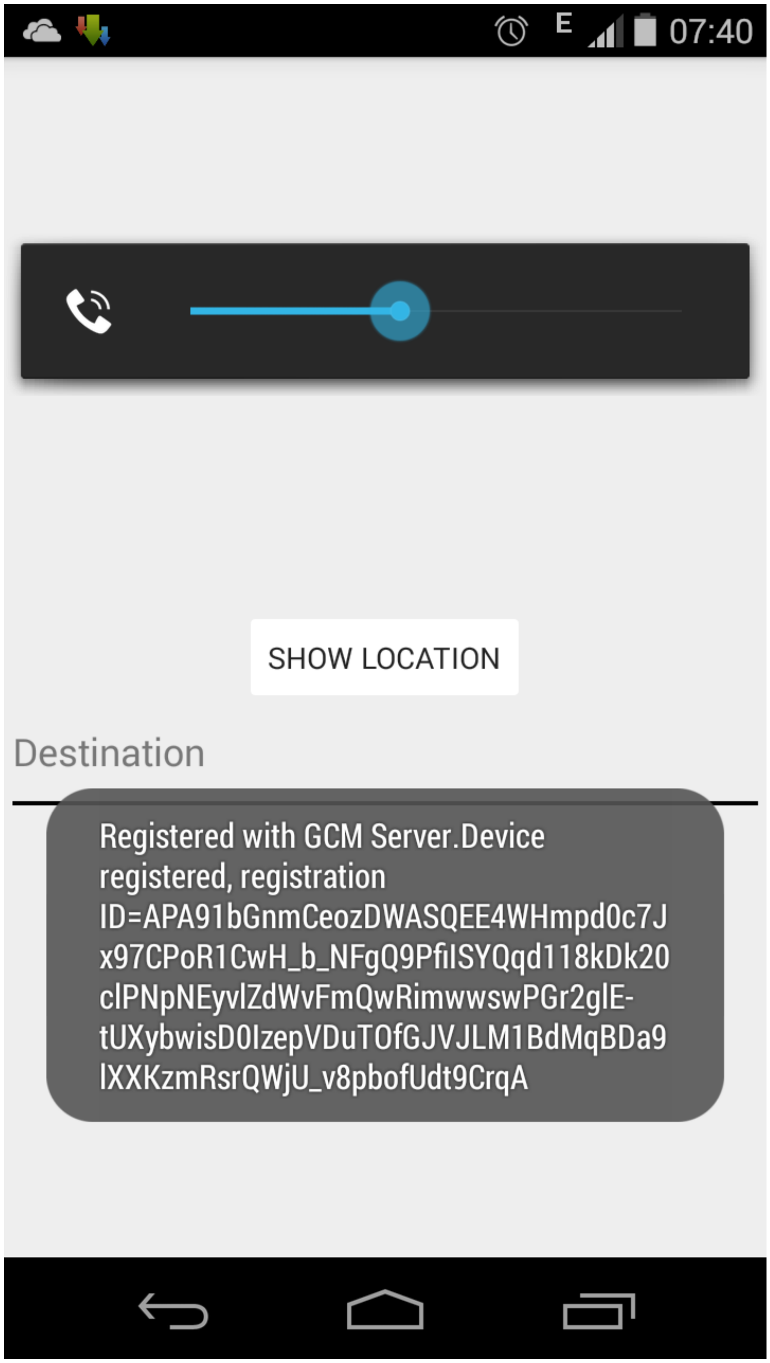
7.4. Google Speech to Text and Text to Speech Services

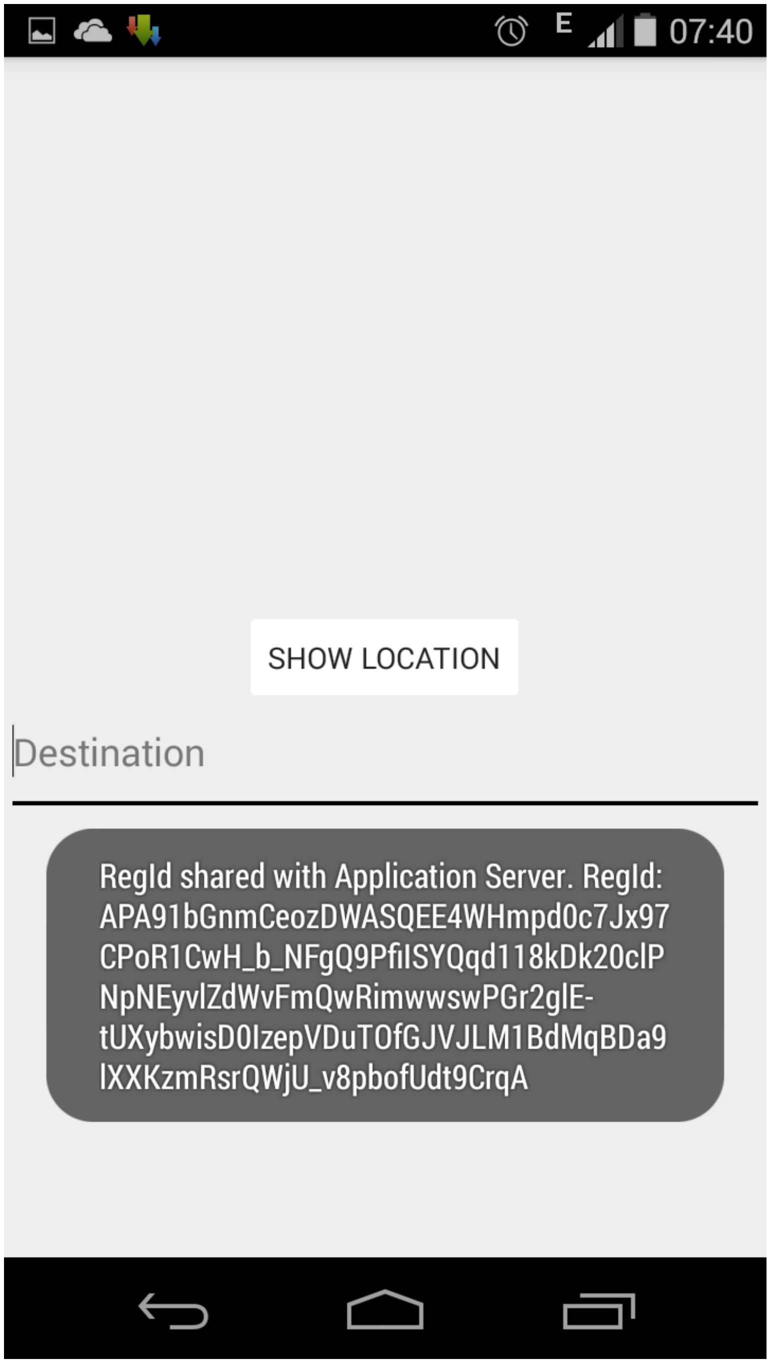
Since a blind person cannot read the output on his phone, that had to be spoken to him. So, we used the Text-to-Speech service to enable voice notification. Also I future, Speech-to-text destination selection will be implemented.

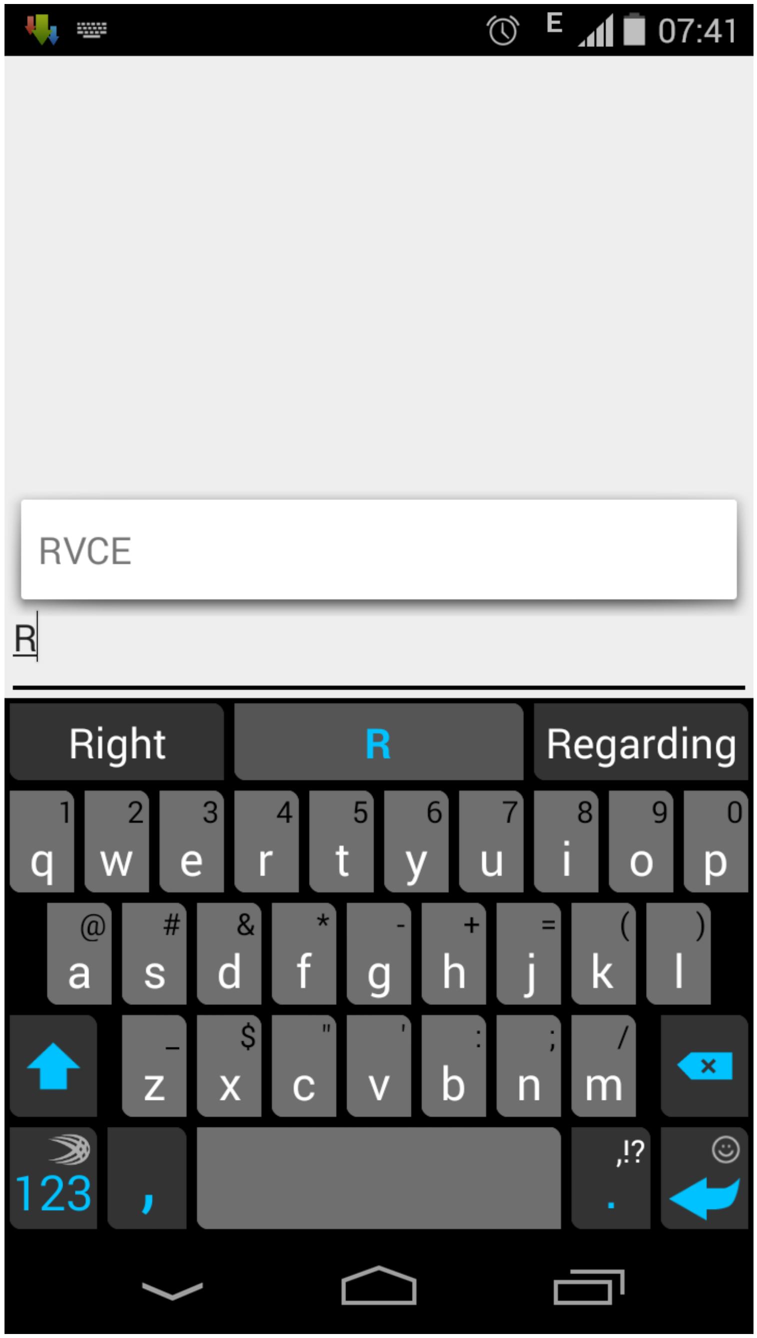
8. Screenshots

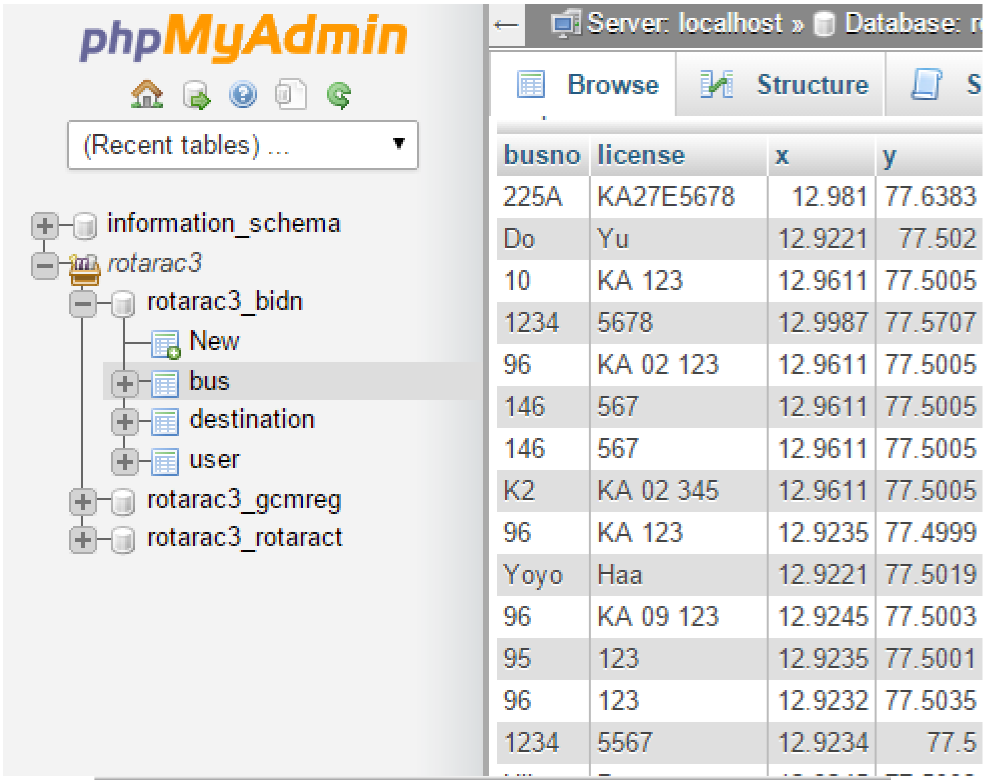


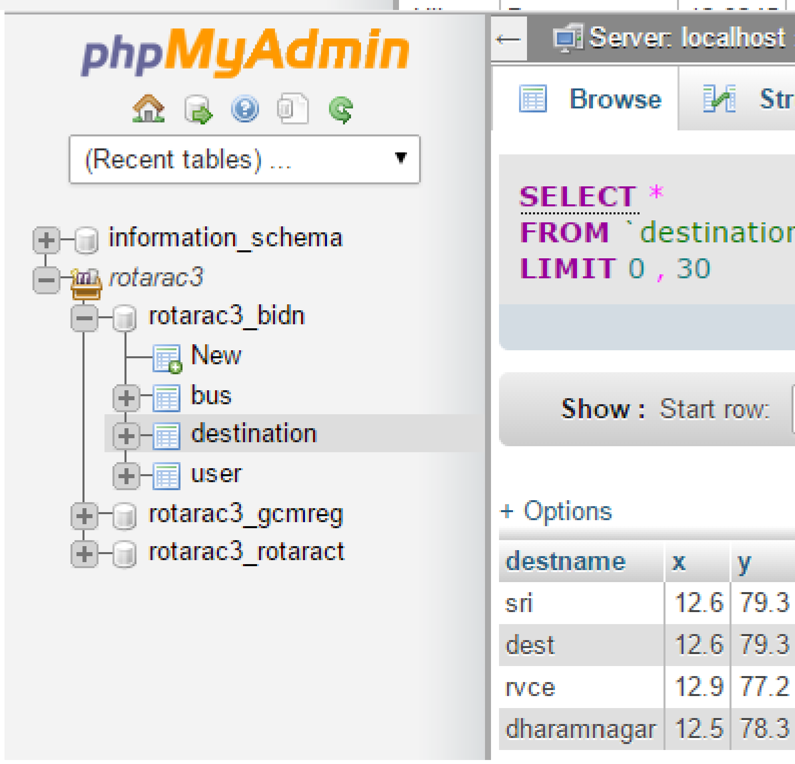

07:40
driver
Hello world!
227B
KA25B196
NEW BUTTON
12.9259127 : 77.5024385

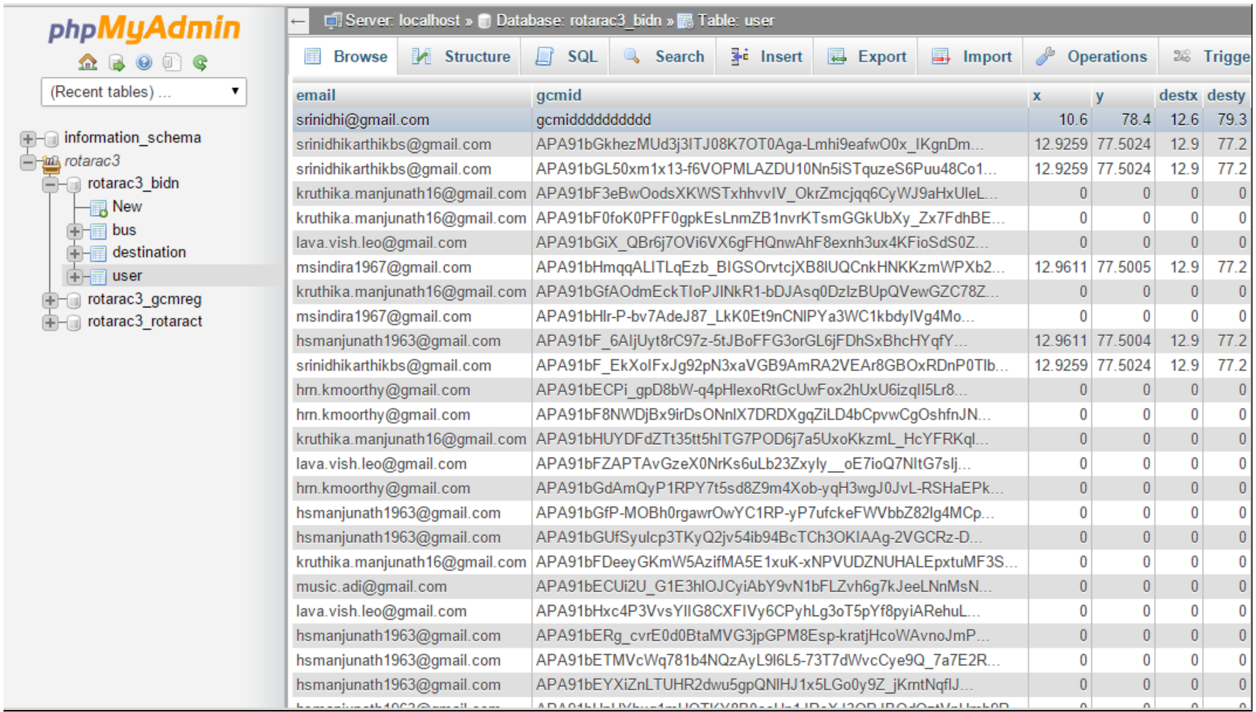

07:40
SHOW LOCATION
Destination
Registered with GCM Server.Device registered, registration ID=APA91bGnmCeozDWASQEE4WHmpd0c7Jx97CPoR1CwH_b_NFgQ9PfilSYQqd118kDk20clPNpNEyvlZdWvFmQwRimwwswPGr2glE-tUXybwisD0IzepVDuTOfGJVJLM1BdMqBDa9lXXKzmRsrQWjU_v8pbofUdt9CrqA


07:40
SHOW LOCATION
Destination
RegId shared with Application Server. RegId: APA91bGnmCeozDWASQEE4WHmpd0c7Jx97 CPoR1CwH_b_NFgQ9PfiISYQqd118kDk20clP NpNEyvlZdWvFmQwRimwwswPGr2glE-tUXybwisD0IzepVDuTOfGJVJLM1BdMqBDa9 lXXKzmRsrQWjU_v8pbofUdt9CrqA


07:41
RVCE
R
Right
R
Regarding
123


phpMyAdmin
(Recent tables) ...
information_schema
rotarac3
rotarac3_bidn
New
bus
destination
user
rotarac3_gcmreg
rotarac3_rotaract
Server: localhost » Database: r
Browse
Structure
S
busno license x y
225A KA27E5678 12.981 77.6383
Do Yu 12.9221 77.502
10 KA 123 12.9611 77.5005
1234 5678 12.9987 77.5707
96 KA 02 123 12.9611 77.5005
146 567 12.9611 77.5005
146 567 12.9611 77.5005
K2 KA 02 345 12.9611 77.5005
96 KA 123 12.9235 77.4999
Yoyo Haa 12.9221 77.5019
96 KA 09 123 12.9245 77.5003
95 123 12.9235 77.5001
96 123 12.9232 77.5035
1234 5567 12.9234 77.5


phpMyAdmin
(Recent tables) ...
information_schema
rotarac3
rotarac3_bidn
New
bus
destination
user
rotarac3_gcmreg
rotarac3_rotaract
Server: localhost
Browse
Stru
SELECT *
FROM `destination
LIMIT 0 , 30
Show : Start row:
+ Options
destname x y
sri 12.6 79.3
dest 12.6 79.3
rvce 12.9 77.2
dharamnagar 12.5 78.3



email	gcmid	x	y	destx	desty
srinidhi@gmail.com	gcmiddddddddd	10.6	78.4	12.6	79.3
srinidhikarthikbs@gmail.com	APA91bGkhezMUd3j3ITJ08K7OT0Aga-Lmhi9eafwO0x_IKgnDm...	12.9259	77.5024	12.9	77.2
srinidhikarthikbs@gmail.com	APA91bGL50xm1x13-f6VOPMLAZDU10Nn5iSTquzeS6Puu48Co1...	12.9259	77.5024	12.9	77.2
kruthika.manjunath16@gmail.com	APA91bF3eBwOodsXKWSTxhhvvIV_OkrZmcjqq6CyWJ9aHxUleL...	0	0	0	0
kruthika.manjunath16@gmail.com	APA91bF0foK0PFF0gpkEsLnmZB1nvrKTsmGGkUbXy_Zx7FdhBE...	0	0	0	0
lava.vish.leo@gmail.com	APA91bGiX_QBr6j7OVi6VX6gFHQnwAhF8exnh3ux4KFioSdS0Z...	0	0	0	0
msindira1967@gmail.com	APA91bHmqqALITLqEzb_BIGSOrvtcjXB8lUQCnkHNKKzmWPXb2...	12.9611	77.5005	12.9	77.2
kruthika.manjunath16@gmail.com	APA91bGfAOdmEckTIoPJINkR1-bDJAsq0DzlzBUpQVewGZC78Z...	0	0	0	0
msindira1967@gmail.com	APA91bHlr-P-bv7AdeJ87_LkK0Et9nCNlPYa3WC1kbdylVg4Mo...	0	0	0	0
hsmanjunath1963@gmail.com	APA91bF_6AIjUyt8rC97z-5tJBoFFG3orGL6jFDhSxBhcHYqfY...	12.9611	77.5004	12.9	77.2
srinidhikarthikbs@gmail.com	APA91bF_EkXolFxJg92pN3xaVGB9AmRA2VEAr8GBOxRDnP0Tlb...	12.9259	77.5024	12.9	77.2
hrn.kmoorthy@gmail.com	APA91bECPi_gpD8bW-q4pHlexoRtGcUwFox2hUxU6izqlI5Lr8...	0	0	0	0
hrn.kmoorthy@gmail.com	APA91bF8NWDjBx9irDsONnlX7DRDXgqZiLD4bCpvwCgOshfnJN...	0	0	0	0
kruthika.manjunath16@gmail.com	APA91bHUYDFdZTt35tt5hITG7POD6j7a5UxoKkzmL_HcYFRKql...	0	0	0	0
lava.vish.leo@gmail.com	APA91bFZAPTAvGzeX0NrKs6uLb23Zxyly__oE7ioQ7NltG7slj...	0	0	0	0
hrn.kmoorthy@gmail.com	APA91bGdAmQyP1RPY7t5sd8Z9m4Xob-yqH3wgJ0JvL-RSHaEPk...	0	0	0	0
hsmanjunath1963@gmail.com	APA91bGfP-MOBh0rgawrOwYC1RP-yP7ufckeFWVbbZ82lg4MCp...	0	0	0	0
hsmanjunath1963@gmail.com	APA91bGUfSyulcp3TKyQ2jv54ib94BcTCh3OKIAAg-2VGCRz-D...	0	0	0	0
kruthika.manjunath16@gmail.com	APA91bFDeeyGKmW5AzifMA5E1xuK-xNPVUDZNUHALEpxtuMF3S...	0	0	0	0
music.adi@gmail.com	APA91bECUi2U_G1E3hlOJCyiAbY9vN1bFLZvh6g7kJeeLNnMsN...	0	0	0	0
lava.vish.leo@gmail.com	APA91bHxc4P3VvsYIIG8CXFIVy6CPyhLg3oT5pYf8pyiARehuL...	0	0	0	0
hsmanjunath1963@gmail.com	APA91bERg_cvrE0d0BtaMVG3jpGPM8Esp-kratjHcoWAvnoJmP...	0	0	0	0
hsmanjunath1963@gmail.com	APA91bETMVcWq781b4NQzAyL9l6L5-73T7dWvcCye9Q_7a7E2R...	0	0	0	0
hsmanjunath1963@gmail.com	APA91bEYXiZnLTUHR2dwu5gpQNlHJ1x5LGo0y9Z_jKrntNqflJ...	0	0	0	0

Conclusion

The 2 applications were successfully developed and deployed. It is tested by our team, and is found to be working fine. The Blind person is able to get voice output notification.

Future Enhancements

Optimizing the Battery life by somehow updating the location of the user not through his phone. Reducing Heating problems by consuming less processor cycles. Bring it into production and actually help blind people. Bring this into the notice of Government officials and somehow bring this into practical use.

References

[1] G Lavanya M.E., Assistant Professor, Preethy. W, Shameem.A , Sushmitha.R IVyear, Biomedical Engineering, Passenger BUS Alert System for Easy Navigation of Blind , 2013 International Conference on Circuits, Power and Computing Technologies [ICCPCT2013].

[2] T.Quoc, M. Kim, H. Lee, and K. Eom, “Wireless Sensor Network apply for the Blind U-bus System,” International Journal of u- and e- Service Science and Technology. Vol. 3, No. 3, Sep. 2010.

[3] M. Zikrul, I. Ismarani, and M. Saaid, “Bus Detection Device for the Blind Using RFID Application,” in the 5th International Colloquium on Signal Processing & Its Applications, CSPA, pp. 6-7 Sep. 2009.

[4] http://assistech.iitd.ernet.in/onboard.php

[5] http://link.springer.com/chapter/10.1007/978-3-642-02710-9_44