University of Patras
Master Thesis

Msc Integrated Circuits in Software & Hardware Systems

Geolocation Nearest Point alert mobile application
“Am Home”

Msc student : Papoutsis Georgios 193
Supervisor : Spyros Denazis, Associate Professor
Table of Contents

1.1 Description ....................................................................................................................................... 5
1.2 Significance of research .................................................................................................................. 6
1.3 Existing Applications for dementia ................................................................................................. 8
2.1 Introduction – Geolocation ............................................................................................................ 17
2.2 Geolocation Standard – W3C Geolocation API ......................................................................... 18
2.3 Important factors that affects Geolocation accuracy ...................................................................... 20
2.4 Algorithms for calculating the distance from given point ........................................................... 21
3.1 Architecture Short Description .................................................................................................... 35
3.2 Technical Overview ....................................................................................................................... 35
  3.2.1 Mobile application ................................................................................................................. 35
  3.2.2 Mobile application Interface Overview .................................................................................. 36
  3.3 Back office – Web Application .................................................................................................... 54
  3.3.2 PHP Structure ....................................................................................................................... 58
3.3 CSS Files and Structure ................................................................................................................. 62
3.4 JavaScript ...................................................................................................................................... 63
3.5 Installation of Front End - Back End Web application .................................................................. 64
3.6 Database Structure ....................................................................................................................... 64
3.7 HTML Structure ............................................................................................................................ 65
3.8 Administrator pages ....................................................................................................................... 66
3.9 Using Google Maps Geocoding API ............................................................................................ 67
4 Pilots .................................................................................................................................................. 70
5 Future Development & Optimizations ............................................................................................... 71
Terminology ........................................................................................................................................ 74
Source Code ....................................................................................................................................... 76
Bibliography of sources ..................................................................................................................... 76
This page left intentionally blank.
To my family and friends
for their love and support,
to my professors for their every day
commitment and dedication
1 The AmHome mobile application

1.1 Description

For the needs of the master in Integrated Circuits in Hardware and Software Systems a mobile application developed. The main functionality of this mobile application is to calculate the distance between user’s position and a predefined static point - point of interest like the home of the user. Moreover the user receives notification messages from mobile application in order to be notified whether he is far away from his home. So for example a user declare his home geolocation in the mobile application and the mobile application notifies the user and his relatives whether he is near to his home or to other defined point.

The purpose of this application is to provide help to people that suffer from brain diseases and their volunteer caregivers. Brain diseases cause decrease in the ability to think and remember that is great enough to affect a person's daily functioning. Dementia\(^1\), also known as senility, is a broad category of brain diseases. The most common type of dementia is Alzheimer's disease, which makes up 50% to 70% of cases. Other common types include vascular dementia (25%), Lewy body dementia (15%), and frontotemporal dementia. Less common causes include normal pressure hydrocephalus, Parkinson's disease, syphilis, and Creutzfeldt – Jakob disease among others. More than one type of dementia may exist in the same person. A small proportion of cases run in families. In the DSM-5, dementia was reclassified as aneurocognitive disorder, with various degrees of severity. Diagnosis is usually based on history of the illness and cognitive testing with medical imaging and blood work used to rule out other possible causes. The mini mental state examination is one commonly used cognitive test. Efforts to prevent dementia include trying to decrease risk factors such as high blood pressure, smoking, diabetes and obesity. Screening the general population for the disease is not recommended.
1.2 Significance of research

Today, as stated in world dementia report\textsuperscript{[2]} over 46 million people live with dementia worldwide, more than four times the population of Greece. This number is estimated to increase to 131.5 million by 2050. Dementia also has a huge economic impact. Today, the total estimated worldwide cost of dementia is 748 billion euro, and it will become a trillion euro disease by 2018. This means that if dementia care were a country, it would be the world’s 18th largest economy, more than the market values of companies such as Apple (742 euro billion), Google (368 euro billion) and Exxon (357 euro billion). In many parts of the world, there is a growing awareness of dementia, but across the globe it remains the case that a diagnosis of dementia can bring with it stigma and social isolation. Today, we estimate that 94% of people living with dementia in low and middle income countries are cared for at home. These are regions where health and care systems often provide limited or no support to people living with dementia or to their families. The 2015 World Alzheimer Report updates data on the prevalence, incidence, cost and trends of dementia worldwide. It also estimates how these numbers will increase in the future, leaving us with no doubt that dementia, including Alzheimer’s disease and other causes, is one of the biggest global public health and social care challenges facing people today and in the future. There are two organisations ADI, the only worldwide federation of Alzheimer associations and global voice on dementia, and Bupa, a purpose-driven global health and care company that is the leading international provider of specialist dementia care, caring for around 60,000 people living with dementia each year. Dementia becomes an international health priority. National dementia plans are the first step towards ensuring all countries are equipped to enable people to live well with dementia, and help to reduce the risk of dementia for future generations. There is now a growing list of countries which have such provision in place or which are developing national dementia plans, but it’s not enough. Given the epidemic scale of dementia, with no known cure on the horizon, and with a global ageing population, all governments and every part of society have to play an active role in helping to create a world where people can enjoy a better quality of life today, and also help reduce the risk of dementia for future generations. Providing a better quality of life for people with dementia can be a reality.
Considering the technological aspect of dementia systematic scientific research[3] into the application of technology for people with dementia has been limited, both in Greece and internationally. Based on the review by Lauriks et al. which focuses on the technology needs of people with dementia and their family caregivers, which originated from an earlier needs assessment carried out among patients with dementia and their volunteer caregivers, including:

- The need for **general and personal information**
- The need for **help with the symptoms of dementia**
- The need for **social contact and companionship**
- The need for **monitoring health and safety**

The main results on findings were that while reduced amount of websites do indeed provide useful information for caregivers, they offer very little and useless information to the patients who are actually suffering from dementia and the websites that were investigated provided very little personal information.

This literature review revealed that the technologies many times used to support “The need for help to cope with the symptoms of dementia” focus especially on supporting day-to-day activities, making better the quality of life and supporting the intake of medication. “The need for social contact and companionship” often concerns the contact between the patient, their family, and the caregivers through telecommunication or information networks. Multimedia systems for patients that enable one to view photos, listen to music, also play a significant role. Finally there is ‘The need for monitoring health and personal safety’, which came up in both the
national and the international studies. In all of the studies it was all about technology that could be used to safeguard patients and raise an alarm to alert a professional caregiver in a nursing home caregiver whenever a dangerous situation arose. In some cases it was also possible to get in touch with the patient during the period in which the alarm went off (like an example through a speaking or listening functionality on the personal alarm).

Many so-called broken products are now being marketed which maybe are not been researched at all. To be more precise this is inevitable with the constant introduction of new technologies. However, it make worries if implementing a partially-developed technology for people with dementia is absolutely without risk..For the effectiveness of the technology it is important that technology and caregivers together with the patients themselves are involved with its design from the outset.. There are plenty of technologies on the market that could be applied, but it is questionable whether or not the effectiveness of these have been systematically and reliably investigated. It is difficult to test the experiences and findings with people with dementia. But this applications are in different way than the am home application .

1.3 Existing Applications for dementia

Several market places for dementia mobile applications and independent software providers searched for their offered approach. For users live with an early stage of any form of dementia, the MindMate-App provides a unique everyday-tools package. This package contains list-making applications, a “My Story” timeline tool, and helpful, customisable reminders to shower, eat, brush teeth and take pills. MindMate’s aim: Helping individuals to stay independent as long as possible. And even better: These tools can be very useful tools for family carers to make the caring process easier.

Following are some applications from google play :

MyAreas

MyAreas primary purpose is to be a safety net for elderlies suffering from dementia and mild brain damaged in case they get lost outside their homes. The app makes it possible for these persons to become more self-reliant and to a higher extent able to move outside their homes. The memory challenges can suddenly appear for the dementia suffers and mild brain damaged while they are heading from A to B, and as a result they forget where they are heading. In case they derogate from the known and secure areas, the mobile will ring an alarm and buzz. They are encouraged to stand still and wait to be found. The app will help them to be found again by close relatives or caregivers.
Close relatives or caregivers will be made aware of the persons location in the app, if he/she is lost. If more than one person is responsible for the lost person, they can in the app coordinate which one of them should take action on the alert. The secure areas in which the elderly is expected to be located in is set-up from the AwareCare portal. An area can be valid 24 hours a day, e.g. the route for walking the dog. An area can also be set to cover a specific appointment e.g. visit to the dentist.

MyAreas have been tuned with some very advanced battery management to reduce battery consumption while the app is running, in order to get the GPS location services to last as long as possible and still ensure a high level of security for the Elderly.


Rating : 5 stars from 8 people

Elderly Care

Elderly Care is an app for caregivers of elderly persons. It is filled with inspiration and information, in order to act as a support and help in a complicated situation. Advice for daily life is combined with information and new findings. You can also register and take part of the latest news and share your own experience or post a question. (app link https://play.google.com/store/apps/details?id=com.sci.elderlycare)

Alzheimer's Disease Pocketcard

Manage Alzheimer’s disease with confidence. The Alzheimer’s Disease Pocketcard app helps physicians and other health care professionals care for patients with Alzheimer’s disease (AD) at the point of care. The app features clinically relevant information on AD and interactive tools to help clinicians efficiently assess patients and interview their caregivers.

Highlights
• Top 10 signs of Alzheimer’s disease
• The latest information on detection, diagnosis and management of Alzheimer’s disease, including pharmacotherapy guidelines and non-pharmacologic strategies
• Interactive tools to assess cognition and function, including the Mini-Cog™, Clock Drawing Test, Saint Louis University Mental Status Exam (SLUMS), Functional Activities Questionnaire (FAQ).
• An Annual Wellness Visit algorithm developed by national experts to help clinicians assess cognition more efficiently
• Current diagnostic criteria, including the DSM-5® and the updated diagnostic criteria and guidelines for Alzheimer’s disease from the National Institute on Aging and the Alzheimer’s Association
• Education/support packets (PDF brochures) from the Alzheimer’s Association that can be e-mailed directly to patients and caregivers
• Bookmarking and notes capability
• Index and full text search
The Alzheimer’s Disease Pocketcard app is a collaboration between the Alzheimer’s Association and Borm Bruckmeier.


geoPulse

With geoPulse application a person can share the location through text – for safety or for fun – to your customized list of contacts. geoPulse is a unique location sharing application that provides the user with a feeling of safety and comfort while offering a way to connect and share location information with family and friends.

• Useful in Emergency & Casual situations
• Tight integration with Google Maps to ensure up to date & accurate maps
• Highly accurate GPS when location services are turned on
• Never compromise on safety - when Panic mode is activated, trusted contacts receive an SMS with your location and can provide immediate help if needed
• Automated SMS alert updates –set your time intervals, turn panic mode on and the app does the rest
• Your location is only shared when YOU want it to be
• Share the world around you in privacy with close friends and family

Panic Usage: Automatically alert your list of pre-selected emergency contacts of your location at a set time interval. How it works:

• Select emergency contacts that will receive texts when you activate Panic Mode
• Create custom alert message
• Set time interval between Panic Mode texts (ex: every 15 mins)
• After activating Panic Mode, the app sends texts to contacts at your selected time interval with alert message and your most recent location
Another different approach to the problem of finding a person which suffers from dementia:

The Lost Person Behavior App is based upon the Search and Rescue (SAR) international gold standard reference tool on where to look for lost and missing persons. Every Public Safety Official needs to know where the missing person might be, what to ask, where to look, and what to do when minutes matter. The app was built around the field needs of Public Safety Officials. No matter how remote the location, the app continues to function since no network connection is required, once installed. This life-saving information is customized for 41 different subject categories. The categories are organized into a hierarchy of groups including external factors (abductions, aircraft), water incidents (boats and persons in the water), wheeled incidents (ATV, mountain bikes, vehicles, etc.), mental states (autism, dementia, despondent, intellectual disability, mental illness, substance intoxication), children (in five different age groups), eighteen outdoor activities (hiker, hunter, climber, gatherer, skier, etc.). The app guides the user in selecting the correct subject category by several means. The subject category wizard allows the user to select potential scenarios and then will suggest the best match for the subject category. For example, if a child was riding his bicycle and was then known to be abducted, the user could select Child, Wheel/Motorized, and Abduction; the wizard would suggest using the abduction category. Alternately, the user can go straight to a subject category by using favorites, an alphabetical list, or the hierarchy list. Once the subject category is selected, the Lost Person Behavior App provides the user with:

- behavioral profiles
- tactical briefings
- specific investigative questions
- spatial and survival statistics
- suggestions for initial tasks (reflex tasking).

The all-new tactical briefings distill the statistics and profiles to the key points that every field searcher must know. The more extensive profile, full statistics, and suggested initial tasks assist even the seasoned search planner. The highly detailed questions for the investigation resulted from decades of SAR experience and thousands of missions. The app even allows you to email a checklist of these critical tasks and interview questions so that you may print them out. The email could also be
used for providing remote support or documenting tasks that have been completed. Determining where to look is one of the major functions of the search statistics, which are now provided in a graphical and tabular format. The statistics from thousands of SAR missions allow the search planner to see where similar previous search subjects were found. The SAR statistics include:

- Distance from the Initial Planning Point
- Elevation model (likelihood of going up, down, or staying at the same elevation)
- Mobility model (how long will the subject keep moving)
- Dispersion angle (given a direction of travel, how well does that predict subject’s location)
- Track Offset model (how far away from a feature such as a road or trail is subject found)
- Find location (at what type of geographic feature was the subject found)
- Scenario (what caused the subject to become missing)
- Survivability overall (what is the overall chance of finding the subject alive or injured)
- Survivability rate (how does survivability rate change in 24-hour increments).

The Lost Person Behavior App can be easily customized to only show the statistical information you actually need. The user can customize for ecoregion, terrain, and urban incidents. The user can view the data in both metric and English units. Several additional helpful tools are built into the app. Contextual help is provided on every page with additional information boxes, which explain every element of the app on the page you need. The information page provides information about the bike wheel model used to describe initial tactical deployment (reflex tasking), a glossary, help, and contact information.

In apple’s iStore found applicatons that help persons with dementia or persons within IoS Following are some applications as reviewed in myageingparent we site [38]

iPad

Apple has revolutionized our lives with many of its products mainly being the iPhone and iPad. From chefs to pilots, even to the marines, people are using their iPads and iPhones to make their lives just a little bit, or a lot, better. When it comes to dementia,
it seems as if the iPad is made for people who suffer from it, especially considering the easy user interface and huge 9.7-inch screen. Here are the tried and tested top five apps that can help those suffering from dementia.

Talking Tom 2

You can talk to Tom and he repeats what you say in his own hilarious voice. You can also stroke him, poke him, challenge him and earn gold coins in a mini-game. You can dress him in new outfits and play silly mini games. Great for a bit of fun, interaction and contact with a talking virtual pet. Talking Ginger is another similar alternative.

My Reef 3D

This app is great for residents who can interact with the fish, or just sit back and enjoy life on a beautiful reef. Featuring 14 types of vibrant tropical fish, MyReef 3D lite includes species-specific movement and interaction. You can individually select the inhabitants of your aquarium, and even annoy them by tapping on the “glass”. You can also knock on the front “glass” to annoy your fish. Care home residents have loved the realistic appearance of this app and because it’s very gentle and easy to use, it’s ideal for individuals with advanced dementia.

Let’s Create! Pottery

With this app, you can touch the screen to make pottery but throwing clay on a virtual wheel. You can smooth the sides of the pots by dragging your finger across the screen and create a collection of pots of different shapes and sizes. This app creates a relaxing, therapeutic experience for individuals with dementia, particularly those who have creative interests.

YouTube

YouTube is an enormous collection of short videos which can be easily searched for and played. Everything from film clips to old news reels to pieces of music can be found on YouTube and this allows participants to request what they would like to see. In all of our sessions using YouTube in reminiscence sessions, only once have we not been able to fulfil a participant’s request. Because many people living in care homes have no concept of how easy it is to once again see their favourite films or hear their favourite music, every time we use this app in an activity session it yields great results.

Augment

Augment is an app which builds a 3D object, animal or person and, using the iPad camera, puts it in the room so that if you look at the screen (or the projector screen if the iPad is plugged in to it), you see everything just as it was before but there is a new object in the room. It also locks it to a particular position so even if you move the
Patra, February of 2016

iPad, it still seems to be there. Using what they know about the participants, Alive! presenters can choose objects which would be most appropriate for the session. For example, in one session Bob*, a former pig farmer, was amazed to see an image of a pig which looked exactly as though it was there in the room with him!

FlowerGarden

One of the best things about iPad apps is that they can play a part in bringing back activities which participants have missed from years gone by. FlowerGarden allows you to pick virtual seeds, plant them, water them and watch them grow. This is especially significant for participants who used to enjoy growing plants and are missing their former gardens. iPads are able to provide activities which replicate things which mean a great deal to older people.

MindMate

The MindMate App is not your ordinary Alzheimer’s or Dementia App. MindMate is like a friend and guardian angel – Always ready to help and always happy to entertain. It has a games section focussing on Attention, Memory, Speed and Problem Solving, a Life section, which helps medical staff and carers know more about the dementia sufferer, reminders to brush your teeth etc and exercise and nutritional advice. And it's free.

Games Section

The MindMate-App provides 8 fun and interactive Games to stimulate user’s cognitive abilities based on world-leading research and in consultation with University of Glasgow’s Geriatric medicine department. Another great thing: The users are able to track the progress in the four different sections: Attention, Memory, Speed and Problem Solving. As a result, the individual living with Dementia will find an engaging and interactive way to stimulate brain activities. Furthermore, MindMate provides access to all the latest research about dementia, interesting blog posts and a lot more. All these features can be found in the Learn More Section of our Dementia app.
My Life Section

Let people access all the personal information they need in our “Getting to know me” section. This is especially helpful, when a person with dementia has to leave their home and go to an hospital or care home. Aim: Help the carer or GP understand who the person is.

Labyrinth 2 HD ($7.99) Labyrinth 2 HD - Illusion Labs

An important aspect to those suffering dementia is to try to keep the brain active. In doing this, you can potentially slow down the disease’s symptoms. Featured in the Top 10 Apps That Will Work best on iPad 3, Labyrinth 2 HD is a great game to exercise the brain. From extremely easy levels to complicated levels with holes, cannons and more, Labyrinth offers a wide range of mazes making it an enjoyable experience for those with dementia.

Logos Quiz (Free) Logos Quiz Game - AticoD

Another way to slow down dementia and increase brain activity is through making the brain remember and recall things. This is why the Logos Quiz game is perfect. With over 500 brands to remember and recall, the Logos Quiz game will certainly prove challenge as well as bringing joy to the individual when they recall a logo correctly.

Draw Free (Free) Draw Free for iPad - David Porter Apps LLC

With this app taking advantage of the 9.7 inch screen on the iPad, Draw Free makes it easy for the individual to let their mind free and draw whatever comes into their head. It is a great app to encourage internal creativity, which is why it has come in at number three. Sometimes, the best apps for those suffering dementia are some of the simplest of apps.

iFish Pond HD - TriggerWave LLC
iFish Pond creates a beautiful pond to look at in the palm of your hands. With the option to choose what creatures lurk in the waters to the bottom of the pond itself, you can customize the pond to your desire. This is great for people with dementia for the fact that it is peaceful and relaxing. As well as this, the added bonus of interacting with the app by touching the screen (making ripples in the water) brings the natural beauty of the pond to life. This app is a must have.

YouTube (Pre-installed)

This may comes as a surprise to people that the best app for people suffering dementia is YouTube. But, if you think about what YouTube has to offer, it will make total sense. People with dementia rely heavily on interaction be it person to person or through touching things (like on iFish Pond). YouTube has million of videos to watch giving the individual endless possibilities to watch whatever they want. The friendly UI of the YouTube app makes using it a breeze for anyone too.

I hope you can see that if you have somebody that suffers dementia, the iPad can help exercise their brain and bring added joy into their lives. It’s not the complicated ‘stock exchange’ apps that are best for people with dementia; they’re more likely to benefit from simpler games such as Doodle Jump. The best apps for those suffering dementia are the simple apps, which replicate moments in time or encourage creativity.
2 Geolocation

2.1 Introduction – Geolocation

The main functionality and algorithms of the am home mobile application are based on the calculation of geolocation. So it is important for our research case to provide an overview about how works geolocation and which algorithms implement achieve the better accuracy. Following is described how geolocation general works as founded in Wikipedia.

Geolocation is the identification of the real-world geographic location of an object, such as a radar source, mobile phone or Internet-connected computer terminal. Geolocation may refer to the practice of assessing the location, or to the actual assessed location. Geolocation is closely related to the use of positioning systems but may be distinguished from it by a greater emphasis on determining a meaningful location (e.g. a street address) rather than just a set of geographic coordinates.

For either geolocating or positioning, the locating engine often uses radio frequency (RF) location methods, for example Time Difference Of Arrival (TDOA) for precision. TDOA systems often utilise mapping displays or other geographic information system. When a GPS signal is unavailable, geolocation applications can use information from cell towers to triangulate the approximate position, a method that is not as accurate as GPS but has greatly improved in recent years. This is in contrast to earlier radiolocation technologies, for example Direction Finding where a line of bearing to a transmitter is achieved as part of the process.

Internet and computer geolocation can be performed by associating a geographic location with the Internet Protocol (IP) address, MAC address, RFID, hardware embedded article/production number, embedded software number (such as UUID, Exif/IPTC/XMP or modern steganography), invoice, Wi-Fi positioning system, device fingerprint, canvas fingerprinting or device GPS coordinates, or other, perhaps self-disclosed information. Geolocation usually works by automatically looking up an IP address on a WHOIS service and retrieving the registrant's physical address.

IP address location data can include information such as country, region, city, postal/zip code, latitude, longitude and timezone. Deeper data sets can determine other parameters such as domain name, connection speed, ISP, language, proxies, company name, US DMA/MSA, NAICS codes, and home/business.
At times geolocation can be more deductive, as with crowdsourcing efforts to determine the position of videos of training camps, combats, and beheadings in Syria by comparing features detected in the video with publicly available map databases such as Google Earth, as practiced by sites such as Bellingcat.\cite{28}\cite{29}

The word \textit{geolocation} is also the latitude and longitude coordinates of a particular location. Term and definition standardized by ISO/IEC 19762-5:2008.

In the field of animal biology and ecology, the word \textit{geolocation} is also used to refer to the process of inferring the location of a tracked animal based, for instance, on the time history of sunlight brightness or the water temperature and depth measured by an instrument attached to the animal. Such instruments are commonly called archival tags (including microchip implants, Pop-up satellite archival tags, and data storage tags) or dataloggers.

### 2.2 Geolocation Standard – W3C Geolocation API

The W3C Geolocation API is an effort by the World Wide Web Consortium (W3C) to standardize an interface to retrieve the geographical location information for a client-side device. It defines a set of objects, ECMAScript standard compliant, that executing in the client application give the client's device location through the consulting of Location Information Servers, which are transparent for the application programming interface (API). The most common sources of location information are IP address, Wi-Fi and Bluetooth MAC address, radio-frequency identification (RFID), Wi-Fi connection location, or device Global Positioning System (GPS) and GSM/CDMA cell IDs. The location is returned with a given accuracy depending on the best location information source available.

Following is described how geolocation is used by web browsers:

Web pages can use the Geolocation API directly if the web browser implements it. Historically, some browsers could gain support via the Google Gears plugin, but this was discontinued in 2010 and the server-side API it depended on stopped responding in 2012.\cite{34}

The Geolocation API is ideally suited to web applications for mobile devices such as personal digital assistants (PDA) and smartphones. On desktop computers, the W3C Geolocation API works in Firefox since version 3.5, Google Chrome,\cite{34} Opera 10.6,\cite{34} Internet Explorer 9.0,\cite{34} and Safari 5. On mobile devices, it works on Android (firmware 2.0+),\textit{iOS}, Windows Phone and Maemo. The W3C Geolocation API is also supported by Opera Mobile 10.1 — available for Android and Symbian devices (S60 generations 3 & 5) since November 24, 2010.\cite{34}
Google Gears provided geolocation support for older and non-compliant browsers, including Internet Explorer 7.0+ as a Gears plugin, and Google Chrome which implemented Gears natively. It also supported geolocation on mobile devices as a plugin for the Android browser (pre version 2.0) and Opera Mobile for Windows Mobile. However, the Google Gears Geolocation API is incompatible with the W3C Geolocation API and is no longer supported.

The API specification itself is agnostic to the geolocation implementation. So while it's true that Mozilla Firefox and Google Chrome both rely on network-enabled geolocation via Google's Location Service database, that's not a requirement of the standard. For example, Mobile Safari on an iPhone will in some cases use GPS rather than WiFi-triangulation or cell-site geolocation, depending on what location service is most accurate and accessible at a given moment. W3C Geolocation API is built on extant technologies, and is heavily influenced by Google Gears Geolocation API. Example: Firefox's Geolocation implementation[8] uses Google's network location provider.[9]

Google Gears Geolocation works by sending a set of parameters that could give a hint as to where the user's physical location is to a network location provider server, which is by default the one provided by Google (code.l.google.com).[35] Some of the parameters are lists of sensed mobile cell towers and Wi-Fi networks, all with sensed signal strengths. These parameters are encapsulated into a JavaScript Object Notation (JSON) message and sent to the network location provider via HTTP POST. Based on these parameters, the network location provider can calculate the location. Common uses for this location information include enforcing access controls, localizing and customizing content, analyzing traffic, contextual advertising and preventing identity theft.[36]

In our application used html5 but not all browsers support HTML5. It’s important to note that not all browsers support the HTML5 Geolocation API, for example Internet Explorer 8. The HTML5 Geolocation API is built into the browser and is accessible using JavaScript methods that access the navigator object. In order to work it requires HTML5 support in the browser. There are some sites that provide information whether a particular browser supports Geolocation like:


Additionally, if a user has disabled JavaScript for some reason, then the Geolocation app won’t work in that browser. JavaScript code is needed to access the API.
2.3 Important factors that affects Geolocation accuracy

In addition HTML5 Geolocation requires an internet connection. If the internet connection lost then it is not able to access the Location Service. With no internet connection most browsers will not return a location. Sometimes can access a cached location that is stored in the browser by the API. But, that cached location is the last valid location that was calculated by the API. Regarding wifi connectivity if Wifi is turned off on a phone, desktop machine, laptop or tablet, the Geolocation API service will try to find the location by other methods which include the public IP address, Cell tower ID triangulation or GPS. Public IP addresses databases usually return a location for the internet providers Point of Presence or PoP. Furthermore, some internet providers offer rotating IP addresses. So it get to use one IP address for a particular time period such as 48 hours and then get a different one. So a Public IP address is usually only good enough to locate a particular City, or a general area of the City, or a Country depending on where are in the world.

As for Cell Tower IDs it depends on what type of information for particular phone and Telco Carrier provides to the API. Some smartphones only return information on the current tower that the phone is pinging, which obviously makes triangulation very difficult and decreases accuracy to within a radius around that tower.

It is noticed that the native Android browser is significantly less accurate without Wifi. Without it I typically see accuracy numbers in the 1000+ meters range. As soon as someone turn Wifi back on and is in a neighborhood or downtown area, the accuracy drops to less than 75 meters almost instantly.

Another important think that have to be taken under consideration is the area location like if it is a rural or urban. Granted the vast majority of users will be in urban locations. However if there are requirements for users traveling outside of urban areas then this section applies. Geolocation in rural areas is significantly less reliable. If Wifi is turned on but the user is not near any Wifi access points, then the Geolocation service will also attempt to fallback to the other methods mentioned above. Triangulation can be much more difficult in rural areas where towers are spread further apart, and for browsers that don’t use GPS the accuracy will suffer significantly.

Another factor that affects Geolocation accuracy is whether the client application is moving or not (stationary). Being stationary in an urban area offers far better accuracy with the Geolocation API than when are moving. On a native Android phones it’s rare to get an accurate reading while driving around town. Occasionally a sporadic result would be returned when stop at a light. In addition if a VPN is turned on, then the location will resolve to the VPN’s public IP address. For example, a user in Athens is logged into the company VPN which host is hosted at their headquarters office in a suburb of Iraklio. The HTML5 Geolocation API will resolve the location to the
headquarters public IP address in Iraklio and not the user’s actual location. Quite a few corporate users have VPNs for security reasons. But in our case are a few users that use VPN.

2.4 Algorithms for calculating the distance from given point

Google Maps API provides the basic functions in order to calculate the distance from a given point. The following is one method that solves the problem of distance calculation: It could easily calculate a bounding circle of x km around a given GPS point, and it is also easy to calculate points that fall on the circumference of this circle, for any angle.

Calculating the road distance from the GPS point to each snapped road point could be done with the directions service of the Google Maps API. Note that this will only work in countries that support directions in Google Maps, but more importantly, the road distance will almost always be greater than 1km, because our bounding circle has a 1km radius "as the crow flies". However if can work with approximate information, this may already be one possible solution. It can also considered starting with the above solution (1km bounding circle, calculate x points on the circumference, and snap them to the closest road), then calculate the road distance of each path (from the GPS point to each snapped point), and then it can be repeated this recursively for each path, each time using a smaller bounding circle, until reach a road distance close to 1km. It can decrease the bounding circle in each recursion, in proportion to the error margin, to make the algorithm more efficient.
Calculating distance, bearing between Latitude/Longitude points:

Following it presented a variety of calculations for latitude/longitude points, with the formulæ and code fragments for implementing them from *Movable Type Scripts web site*[^1].

All these formulæ are for calculations on the basis of a spherical earth (ignoring ellipsoidal effects) – which is accurate enough for most purposes. In fact, the earth is very slightly ellipsoidal; using a spherical model gives errors typically up to 0.3%[^1] – see notes for further details.

Great-circle distance between two points

Enter the co-ordinates into the text boxes to try out the calculations. A variety of formats are accepted, principally:

- deg-min-sec suffixed with N/S/E/W (e.g. 40°44′55″N, 73 59 11W), or
- signed decimal degrees without compass direction, where negative indicates west/south (e.g. 40.7486, -73.9864):

Point 1: 50 03 59N, 005 42 53W

Point 2: 58 38 38N, 003 04 12W

Distance: 968.9 km (to 4 SF*)

Initial bearing: 009° 07′ 11″

Final bearing: 011° 16′ 31″

Midpoint: 54° 21′ 44″ N, 004° 31′ 50″ W

**Distance**

This uses the ‘haversine’ formula to calculate the great-circle distance between two points – that is, the shortest distance over the earth’s surface – giving an ‘as-the-crow-flies’ distance between the points

\[
a = \sin^2\left(\frac{\Delta\varphi}{2}\right) + \cos \varphi_1 \cdot \cos \varphi_2 \cdot \sin^2\left(\frac{\Delta\lambda}{2}\right)
\]

*Haversine formula:*

\[
c = 2 \cdot \arctan\left(\sqrt{a}, \sqrt{1-a}\right)
\]

\[
d = R \cdot c
\]
\[ \phi \text{ is latitude, } \lambda \text{ is longitude, } R \text{ is earth’s radius (mean radius } = 6,371\text{km); note that angles need to be in radians to pass to trig functions!} \]

JavaScript:
```
var R = 6371000; // metres
var φ1 = lat1.toRadians();
var φ2 = lat2.toRadians();
var Δφ = (lat2-lat1).toRadians();
var Δλ = (lon2-lon1).toRadians();

var a = Math.sin(Δφ/2) * Math.sin(Δφ/2) + Math.cos(φ1) * Math.cos(φ2) * Math.sin(Δλ/2) * Math.sin(Δλ/2);
var c = 2 * Math.atan2(Math.sqrt(a), Math.sqrt(1-a));
```

\[ d = R \times c; \]

Note in these scripts, I generally use lat/lon for latitude/longitude in degrees, and φ/λ for latitude/longitude in radians – having found that mixing degrees & radians is often the easiest route to head-scratching bugs...

Historical aside: The height of technology for navigator’s calculations used to be log tables. As there is no (real) log of a negative number, the ‘versine’ enabled them to keep trig functions in positive numbers. Also, the \( \sin^2(0/2) \) form of the haversine avoided addition (which entailed an anti-log lookup, the addition, and a log lookup). Printed tables for the haversine/inverse-haversine (and its logarithm, to aid multiplications) saved navigators from squaring sines, computing square roots, etc – arduous and error-prone activities.

The haversine formula ‘remains particularly well-conditioned for numerical computation even at small distances’ – unlike calculations based on the spherical law of cosines. The ‘versed sine’ is \( 1-\cos0 \), and the ‘half-versed-sine’ is \( (1-\cos0)/2 = \sin^2(0/2) \) as used above. Once widely used by navigators, it was described by Roger Sinnott in Sky & Telescope magazine in 1984 (“Virtues of the Haversine”): Sinnott explained that the angular separation between Mizar and Alcor in Ursa Major – 0°11′49.69″ – could be accurately calculated on a TRS-80 using the haversine.

For the curious, \( c \) is the angular distance in radians, and \( a \) is the square of half the chord length between the points. A (remarkably marginal) performance improvement...
may be obtained by factoring out the terms which get squared. If \( \text{atan2} \) is not available, \( c \) could be calculated from \( 2 \cdot \sin(\min(1, \sqrt{a})) \) (including protection against rounding errors).

**Spherical Law of Cosines**

In fact, JavaScript (and most modern computers & languages) use ‘IEEE 754’ 64-bit floating-point numbers, which provide 15 significant figures of precision. By my estimate, with this precision, the simple spherical law of cosines formula (\( \cos c = \cos a \cos b + \sin a \sin b \cos C \)) gives well-conditioned results down to distances as small as a few metres on the earth’s surface. (Note that the geodetic form of the law of cosines is rearranged from the canonical one so that the latitude can be used directly, rather than the colatitude).

This makes the simpler law of cosines a reasonable 1-line alternative to the haversine formula for many geodesy purposes (if not for astronomy). The choice may be driven by programming language, processor, coding context, available trig functions (in different languages), etc – and, for very small distances an equirectangular approximation may be more suitable.

**Law of cosines:**
\[
d = \cos(\sin \varphi_1 \cdot \sin \varphi_2 + \cos \varphi_1 \cdot \cos \varphi_2 \cdot \cos \Delta \lambda) \cdot R
\]

\[
\text{var } d = \text{Math.acos( Math.sin(φ1)*Math.sin(φ2) + Math.cos(φ1)*Math.cos(φ2)*Math.cos(Δλ) ) } \cdot R;
\]

**JavaScript:**
\[
=\text{ACOS( SIN(lat1)*SIN(lat2) + COS(lat1)*COS(lat2)*COS(lon2-lon1) ) } \cdot 6371000
\]

**Excel:**
\[
=\text{ACOS( SIN(lat1*PI()/180)*SIN(lat2*PI()/180) + COS(lat1*PI()/180)*COS(lat2*PI()/180)*COS(lon2-lon1*PI()/180) ) } \cdot 6371000
\]

**Equirectangular approximation**

If performance is an issue and accuracy less important, for small distances Pythagoras’ theorem can be used on an equirectangular projection:

\[
x = \Delta \lambda \cdot \cos \varphi_m
\]
\[
y = \Delta \phi
\]

**Formula**
\[
d = R \cdot \sqrt{x^2 + y^2}
\]
\[ \text{JavaScript: } \var x = (\lambda_2 - \lambda_1) \times \cos((\varphi_1 + \varphi_2)/2); \]

\[ \var y = (\varphi_2 - \varphi_1); \]

This uses just one trig and one sqrt function – as against half-a-dozen trig functions for cos law, and 7 trigs + 2 sqrts for haversine. Accuracy is somewhat complex: along meridians there are no errors, otherwise they depend on distance, bearing, and latitude, but are small enough for many purposes – (and often trivial compared with the spherical approximation itself).

Alternatively, the polar coordinate flat-earth formula can be used: using the co-latitudes \( \theta_1 = \pi/2 - \varphi_1 \) and \( \theta_2 = \pi/2 - \varphi_2 \), then

\[ d = R \times \sqrt{\theta_1^2 + \theta_2^2 - 2 \times \theta_1 \times \theta_2 \times \cos \Delta\lambda}. \]

I’ve not compared accuracy.

Baghdad to Osaka – not a constant bearing!

**Bearing**

In general, the current heading will vary as follow a great circle path (orthodrome); the final heading will differ from the initial heading by varying degrees according to distance and latitude (if were to go from say 35°N,45°E (≈ Baghdad) to 35°N,135°E (≈ Osaka), would start on a heading of 60° and end up on a heading of 120°!).

This formula is for the initial bearing (sometimes referred to as forward azimuth) which if followed in a straight line along a great-circle arc will take from the start point to the end point:

\[ \theta = \text{atan2}( \sin \Delta\lambda \times \cos \varphi_2 , \cos \varphi_1 \times \sin \varphi_2 - \sin \varphi_1 \times \cos \varphi_2 \times \cos \Delta\lambda ) \]

\[ \text{JavaScript: } \var y = \sin(\lambda_2 - \lambda_1) \times \cos(\varphi_2); \]

*(all angles*)
in radians)  \( \text{var } x = \text{Math.cos}(\phi_1) \ast \text{Math.sin}(\phi_2) - \text{Math.sin}(\phi_1) \ast \text{Math.cos}(\phi_2) \ast \text{Math.cos}(\lambda_2 - \lambda_1); \)

\( \text{var brng} = \text{Math.atan2}(y, x).\text{toDegrees}(); \)

**Excel:**

\( =\text{ATAN2} (\text{COS(lat1)} \ast \text{SIN(lat2)} - \text{SIN(lat1)} \ast \text{COS(lat2)} \ast \text{COS(lon2-lon1)}, \text{SIN(lon2-lon1)} \ast \text{COS(lat2)}), \)

*note that Excel reverses the arguments to ATAN2 – see notes below*

Since atan2 returns values in the range -\( \pi \) ... +\( \pi \) (that is, -180° ... +180°), to normalise the result to a compass bearing (in the range 0° ... 360°, with –ve values transformed into the range 180° ... 360°), convert to degrees and then use (0+360) % 360, where % is (floating point) modulo.

For final bearing, simply take the initial bearing from the end point to the start point and reverse it (using 0 = (0+180) % 360).

**Midpoint**

This is the half-way point along a great circle path between the two points.  

\( B_x = \cos \phi_2 \cdot \cos \Delta \lambda \)
\( B_y = \cos \phi_2 \cdot \sin \Delta \lambda \)
\( \phi_m = \text{atan2}( \sin \phi_1 + \sin \phi_2, \sqrt{(\cos \phi_1 + B_x)^2 + B_y^2} ) \)

**Formula:**

\( \lambda_m = \lambda_1 + \text{atan2}(B_y, \cos(\phi_1) + B_x) \)

\( \text{var Bx} = \text{Math.cos}(\phi_2) \ast \text{Math.cos}(\lambda_2 - \lambda_1); \)
\( \text{var By} = \text{Math.cos}(\phi_2) \ast \text{Math.sin}(\lambda_2 - \lambda_1); \)
\( \text{var } \phi_3 = \text{Math.atan2}(\text{Math.sin}(\phi_1) + \text{Math.sin}(\phi_2), \text{Math.sqrt}( (\text{Math.cos}(\phi_1) + \text{Bx})^2 + \text{By}^2) ); \)

**JavaScript:**

\( \text{Math.sqrt((Math.cos(\phi_1)+Bx)*(Math.cos(\phi_1)+Bx) + By*By )}); \)

*all angles in radians*  

\( \text{var } \lambda_3 = \lambda_1 + \text{Math.atan2}(\text{By, Math.cos(\phi_1) + Bx}); \)

The longitude can be normalised to -180…+180 using (lon+540)%360-180

Just as the initial bearing may vary from the final bearing, the midpoint may not be located half-way between latitudes/longitudes; the midpoint between 35°N,45°E and 35°N,135°E is around 45°N,90°E.
Intermediate point

An intermediate point at any fraction along the great circle path between two points can also be calculated.

\[
\begin{align*}
a &= \sin((1-f) \cdot \delta) / \sin \delta \\
b &= \sin(f \cdot \delta) / \sin \delta \\
x &= a \cdot \cos \varphi_1 \cdot \cos \lambda_1 + b \cdot \cos \varphi_2 \cdot \cos \lambda_2 \\
y &= a \cdot \cos \varphi_1 \cdot \sin \lambda_1 + b \cdot \cos \varphi_2 \cdot \sin \lambda_2 \\
z &= a \cdot \sin \varphi_1 + b \cdot \sin \varphi_2 \\
\varphi_i &= \text{atan2}(z, \sqrt{x^2 + y^2})
\end{align*}
\]

*Formula:* \( \lambda_i = \text{atan2}(y, x) \)

where \( f \) is fraction along great circle route \((f=0 \text{ is point 1, } f=1 \text{ is point 2})\), \( \delta \) is the angular distance \( d/R \) between the two points.

Destination point given distance and bearing from start point

Given a start point, initial bearing, and distance, this will calculate the destination point and final bearing travelling along a (shortest distance) great circle arc.

Destination point along great-circle given distance and bearing from start point

Start point: 53°19′14″N, 001°43′47″W

Bearing: 096°01′18″

Distance: \( \text{km} \)

Destination point: 53° 11′ 18″ N, 000° 08′ 00″ E

Final bearing: 097° 30′ 52″

*Formula:* \( \varphi_2 = \text{asin}( \sin \varphi_1 \cdot \cos \delta + \cos \varphi_1 \cdot \sin \delta \cdot \cos \theta ) \)

\( \lambda_2 = \lambda_1 + \text{atan2}( \sin \theta \cdot \sin \delta \cdot \cos \varphi_1, \cos \delta - \sin \varphi_1 \cdot \sin \varphi_2 ) \)

\( \varphi \) is latitude, \( \lambda \) is longitude, \( \theta \) is the bearing (clockwise from north), \( \delta \) is the angular distance \( d/R \); \( d \) being the distance travelled, \( R \) the earth’s radius

*JavaScript:* \( \text{var } \varphi_2 = \text{Math.asin}( \text{Math.sin}(\varphi_1) * \text{Math.cos}(d/R) + \)
$\text{Math.cos}(\varphi_1) \times \text{Math.sin}(d/R) \times \text{Math.cos}(\text{brng})$; 

$\text{var}\; \lambda_2 = \lambda_1 + \text{Math.atan2}(\text{Math.sin(}\text{brng}) \times \text{Math.sin}(d/R) \times \text{Math.cos}(\varphi_1), \text{Math.cos}(d/R) - \text{Math.sin}(\varphi_1) \times \text{Math.sin}(\varphi_2))$;

The longitude can be normalised to $-180...+180$ using $(\text{lon}+540) \% 360 - 180$

$\text{lat}_2: = \text{ASIN}(\text{SIN}(\text{lat}_1) \times \text{COS}(d/R) + \text{COS}(\text{lat}_1) \times \text{SIN}(d/R) \times \text{COS}(\text{brng}))$ 

$\text{lon}_2: = \text{lon}_1 + \text{ATAN2}(\text{COS}(d/R) - \text{SIN}(\text{lat}_1) \times \text{SIN}(\text{lat}_2), \text{SIN}(\text{brng}) \times \text{SIN}(d/R) \times \text{COS}(\text{lat}_1))$

* Remember that Excel reverses the arguments to ATAN2 – see notes below

For final bearing, simply take the initial bearing from the end point to the start point and reverse it with $(\text{brng}+180) \% 360$.

### Intersection of two paths given start points and bearings

This is a rather more complex calculation than most others on this page, but I've been asked for it a number of times. This comes from Ed William’s aviation formulary. See below for the JavaScript.

Intersection of two great-circle paths

Point 1: 51.8853 N , 0.2545 E  Brng 1: 108.55°

Point 2: 49.0034 N , 2.5735 E  Brng 2: 32.44°

Intersection point: 50° 54′ 27″ N, 004° 30′ 31″ E

\[
\begin{align*}
\delta_{12} &= 2 \times \text{asin} \left( \sqrt{\text{sin}^2(\Delta \varphi/2) + \text{cos} \; \varphi_1 \times \text{cos} \; \varphi_2 \times \text{sin}^2(\Delta \lambda/2)} \right) \\
\theta_a &= \text{acos} \left( \text{sin} \; \varphi_2 - \text{sin} \; \varphi_1 \times \text{cos} \; \delta_{12} / \text{sin} \; \delta_{12} \times \text{cos} \; \varphi_1 \right) \\
\theta_b &= \text{acos} \left( \text{sin} \; \varphi_1 - \text{sin} \; \varphi_2 \times \text{cos} \; \delta_{12} / \text{sin} \; \delta_{12} \times \text{cos} \; \varphi_2 \right)
\end{align*}
\]

if \( \text{sin}(\lambda_2-\lambda_1) > 0 \)

\[
\begin{align*}
\theta_{12} &= \theta_a \\
\theta_{21} &= 2\pi - \theta_b
\end{align*}
\]

else

\[
\begin{align*}
\theta_{12} &= 2\pi - \theta_a \\
\theta_{21} &= \theta_b
\end{align*}
\]

**Formula:**

\[\alpha_1 = (\theta_{13} - \theta_{12} + \pi) \% 2\pi - \pi\]
\[ \alpha_2 = (\theta_21 - \theta_23 + \pi) \% 2\pi - \pi \]

\[ \alpha_3 = \arccos(-\cos \alpha_1 \cdot \cos \alpha_2 + \sin \alpha_1 \cdot \sin \alpha_2 \cdot \cos \delta_{12}) \]

\[ \delta_{13} = \arctan2(\sin \delta_{12} \cdot \sin \alpha_1 \cdot \sin \alpha_2, \cos \alpha_1 \cdot \cos \alpha_2 + \cos \alpha_1 \cdot \cos \alpha_3) \]

\[ \phi_3 = \arcsin(\sin \phi_1 \cdot \cos \delta_{13} + \cos \phi_1 \cdot \sin \delta_{13} \cdot \cos \theta_{13}) \]

\[ \Delta \lambda_{13} = \arctan2(\sin \theta_{13} \cdot \sin \delta_{13} \cdot \cos \phi_1, \cos \delta_{13} - \sin \phi_1 \cdot \sin \phi_3) \]

\[ \lambda_3 = (\lambda_1 + \Delta \lambda_{13} + \pi) \% 2\pi - \pi \]

where \( \% \) = (floating point) modulo

if \( \sin \alpha_1 = 0 \) and \( \sin \alpha_2 = 0 \): infinite solutions

if \( \sin \alpha_1 \cdot \sin \alpha_2 < 0 \): ambiguous solution

this formulation is not always well-conditioned for meridional or equatorial lines

Note – Here’s a new one: I’ve sometimes been asked about distance of a point from a great-circle path (sometimes called cross track error).

**Formula:**

\[ d_{xt} = \arcsin(\sin(\delta_{13}) \cdot \sin(\theta_{13} - \theta_{12})) \cdot R \]

\( \delta_{13} \) is (angular) distance from start point to third point

\( \theta_{13} \) is (initial) bearing from start point to third point

\( \theta_{12} \) is (initial) bearing from start point to end point

where \( R \) is the earth’s radius

**JavaScript:**

```javascript
var dXt = Math.asin(Math.sin(d13/R)*Math.sin(013-012)) * R;
```

Here, the great-circle path is identified by a start point and an end point – depending on what initial data you’re working from, can use the formulæ above to obtain the relevant distance and bearings. The sign of \( d_{xt} \) tells which side of the path the third point is on.

The along-track distance, from the start point to the closest point on the path to the third point, is

**Formula:**

\[ d_{at} = \arccos(\cos(\delta_{13}) / \cos(d_{xt})) \cdot R \]
\[ \delta_{13} \text{ is (angular) distance from start point to third point} \]
\[ \delta_{\alpha} \text{ is (angular) cross-track distance} \]

where \( R \) is the earth’s radius

JavaScript: \[ \text{var dAt} = \text{Math.acos(Math.cos(d13/R)/Math.cos(dXt/R))} \ast R; \]

Closest point to the poles

And: ‘Clairaut’s formula’ will give the maximum latitude of a great circle path, given a bearing \( \theta \) and latitude \( \phi \) on the great circle:

Formula: \[ \varphi_{\text{max}} = \text{acos}(|\sin \theta \cdot \cos \varphi|) \]

JavaScript: \[ \text{var } \phi_{\text{Max}} = \text{Math.acos(Math.abs(Math.sin(\theta)\astMath.cos(\phi))}); \]

Rhumb lines

A ‘rhumb line’ (or loxodrome) is a path of constant bearing, which crosses all meridians at the same angle.

Sailors used to (and sometimes still) navigate along rhumb lines since it is easier to follow a constant compass bearing than to be continually adjusting the bearing, as is needed to follow a great circle. Rhumb lines are straight lines on a Mercator Projection map (also helpful for navigation).

Rhumb lines are generally longer than great-circle (orthodrome) routes. For instance, London to New York is 4\% longer along a rhumb line than along a great circle – important for aviation fuel, but not particularly to sailing vessels. New York to Beijing – close to the most extreme example possible (though not sailable!) – is 30\% longer along a rhumb line.

Rhumb-line distance between two points

Point 1: 50 21 59N, 004 08 02W

Point 2: 42 21 04N, 071 02 27W

Distance: 5198 km

Bearing: 260° 07’ 38”
Midpoint: 46° 21′ 32″ N, 038° 49′ 00″ W

Destination point along rhumb line given distance and bearing from start point

Start point: 51° 07′ 32″ N, 001° 20′ 17″ E

Bearing: 116° 38′ 10″

Distance: km

Destination point: 50° 57′ 48″ N, 001° 51′ 09″ E

Key to calculations of rhumb lines is the inverse Gudermannian function\textsuperscript{1}, which gives the height on a Mercator projection map of a given latitude: \( \ln(\tan\varphi + \sec\varphi) \) or \( \ln(\tan(\pi/4 + \varphi/2)) \). This of course tends to infinity at the poles (in keeping with the Mercator projection). For obsessives, there is even an ellipsoidal version, the ‘isometric latitude’: \( \psi = \ln(\tan(\pi/4 + \varphi/2)) / \left[ (1-e\cdot\sin\varphi) / (1+e\cdot\sin\varphi) \right]^{\varphi/2} \), or its better-conditioned equivalent \( \psi = \operatorname{atanh}(\sin\varphi) - e\cdot\operatorname{atanh}(e\cdot\sin\varphi) \).

The formulæ to derive Mercator projection easting and northing coordinates from spherical latitude and longitude are then\textsuperscript{1}

\[ E = R \cdot \lambda \]
\[ N = R \cdot \ln(\tan(\pi/4 + \varphi/2)) \]

The following formulæ are from Ed Williams’ aviation formulary\textsuperscript{1}

**Distance**

Since a rhumb line is a straight line on a Mercator projection, the distance between two points along a rhumb line is the length of that line (by Pythagoras); but the distortion of the projection needs to be compensated for.

On a constant latitude course (travelling east-west), this compensation is simply \( \cos\varphi \); in the general case, it is \( \Delta\varphi/\Delta\psi \) where \( \Delta\psi = \ln(\tan(\pi/4 + \varphi_2/2) / \tan(\pi/4 + \varphi_1/2)) \) (the ‘projected’ latitude difference)

**Formula:** \( \Delta\psi = \ln(\tan(\pi/4 + \varphi_2/2) / \tan(\pi/4 + \varphi_1/2)) \) (‘projected’ latitude difference)

\[ q = \Delta\varphi/\Delta\psi \) (or \( \cos\varphi \) for E-W line)
\[ d = \sqrt{(\Delta \phi^2 + q^2 \cdot \Delta \lambda^2)} \cdot R \]  

(Pythagoras)

\( \phi \) is latitude, \( \lambda \) is longitude, \( \Delta \lambda \) is taking shortest route (<180°), \( R \) is the earth’s radius, \( \ln \) is natural log

\[ \text{var } \Delta \psi = \text{Math.log(Math.tan(Math.PI/4 + \phi_2/2)/Math.tan(Math.PI/4 + \phi_1/2))}; \]

\[ \text{var } q = \text{Math.abs(}\Delta \psi) > 10^{-12} \text{ ? } \Delta \phi / \Delta \psi : \text{Math.cos}(\phi_1); \]  // E-W course becomes ill-conditioned with 0/0

JavaScript:

// if dLon over 180° take shorter rhumb line across the anti-meridian:

\[ \text{if (Math.abs(}\Delta \lambda) > Math.PI) \Delta \lambda = \Delta \lambda > 0 \text{ ? } -(2*Math.PI-\Delta \lambda) : (2*Math.PI+\Delta \lambda); \]

\[ \text{var dist = Math.sqrt(}\Delta \phi \cdot \Delta \phi + q \cdot q \cdot \Delta \lambda \cdot \Delta \lambda) \cdot R; \]

**Bearing**

A rhumb line is a straight line on a Mercator projection, with an angle on the projection equal to the compass bearing.

\[ \Delta \psi = \text{ln}(\tan(\pi/4 + \phi_2/2) / \tan(\pi/4 + \phi_1/2)) \]  

(‘projected’ latitude difference)

\[ \theta = \text{atan2}(\Delta \lambda, \Delta \psi) \]

\( \phi \) is latitude, \( \lambda \) is longitude, \( \Delta \lambda \) is taking shortest route (<180°), \( R \) is the earth’s radius, \( \ln \) is natural log

\[ \text{var } \Delta \psi = \text{Math.log(Math.tan(Math.PI/4 + \phi_2/2)/Math.tan(Math.PI/4 + \phi_1/2))}; \]

// if dLon over 180° take shorter rhumb line across the anti-meridian:

\[ \text{if (Math.abs(}\Delta \lambda) > Math.PI) \Delta \lambda = \Delta \lambda > 0 \text{ ? } -(2*Math.PI-\Delta \lambda) : (2*Math.PI+\Delta \lambda); \]

JavaScript:

(all angles in radians) \[ \text{var brng = Math.atan2}(\Delta \lambda, \Delta \psi) \cdot \text{toDegrees}(); \]
**Destination**

Given a start point and a distance \( d \) along constant bearing \( \theta \), this will calculate the destination point. If maintain a constant bearing along a rhumb line, will gradually spiral in towards one of the poles.

**Formula:**

\[
\delta = \frac{d}{R} \quad \text{(angular distance)}
\]

\[
\Delta\psi = \ln\left( \frac{\tan(\pi/4 + \varphi_2/2)}{\tan(\pi/4 + \varphi_1/2)} \right) \quad \text{('projected' latitude difference)}
\]

\[
q = \frac{\Delta\varphi}{\Delta\psi} \quad \text{(or cos \( \varphi \) for E-W line)}
\]

\[
\Delta\lambda = \frac{\delta \cdot \sin \theta}{q}
\]

\[
\varphi_2 = \varphi_1 + \delta \cdot \cos \theta
\]

\[
\lambda_2 = \lambda_1 + \Delta\lambda
\]

**where**

- \( \varphi \) is latitude, \( \lambda \) is longitude, \( \Delta\lambda \) is taking shortest route (<180°), \( \ln \) is natural log, \( R \) is the earth’s radius

**JavaScript:**

```javascript
var Δφ = δ*Math.cos(θ);
var φ2 = φ1 + Δλ;

var Δψ = Math.log(Math.tan(φ2/2+Math.PI/4)/Math.tan(φ1/2+Math.PI/4));
var q = Math.abs(Δψ) > 10e-12 ? Δφ / Δψ : Math.cos(φ1);
// E-W course becomes ill-conditioned with 0/0
var Δλ = δ*Math.sin(θ)/q;
var λ2 = λ1 + Δλ;
```

- check for some daft bugger going past the pole, normalise latitude if so

The longitude can be normalised to −180…+180 using (lon+540)%360-180

Patra, February of 2016
Mid-point

This formula for calculating the ‘loxodromic midpoint’, the point half-way along a rhumb line between two points, is due to Robert Hill and Clive Tooth\(^1\) (thx Axel!).

**Formula:**

\[
\phi_m = \frac{(\phi_1 + \phi_2)}{2}
\]

\[
f_1 = \tan(\pi/4 + \phi_1/2)
\]

\[
f_2 = \tan(\pi/4 + \phi_2/2)
\]

\[
f_m = \tan(\pi/4 + \phi_m/2)
\]

\[
\lambda_m = \left[ \frac{(\lambda_2 - \lambda_1) \cdot \ln(f_m) + \lambda_1 \cdot \ln(f_2) - \lambda_2 \cdot \ln(f_1)}{\ln(f_2/f_1)} \right]
\]

*where* \(\phi\) *is latitude*, \(\lambda\) *is longitude*, \(\ln\) *is natural log*

```javascript
if (Math.abs(\(\lambda_2-\lambda_1\)) > Math.PI) \(\lambda_1 += 2*\text{Math.PI};\) // crossing anti-meridian

var \(\phi_3 = (\phi_1+\phi_2)/2;\)

var \(f_1 = \text{Math.tan}(\text{Math.PI}/4 + \phi_1/2);\)

var \(f_2 = \text{Math.tan}(\text{Math.PI}/4 + \phi_2/2);\)

var \(f_3 = \text{Math.tan}(\text{Math.PI}/4 + \phi_3/2);\)

var \(\lambda_3 = \frac{(\(\lambda_2-\lambda_1\))*\text{Math.log}(f_3) + \lambda_1*\text{Math.log}(f_2) - \lambda_2*\text{Math.log}(f_1)}{\text{Math.log}(f_2/f_1)};\)

**JavaScript:**

*(all angles in radians)* if (!isFinite(\(\lambda_3\)) \(\lambda_3 = (\lambda_1+\lambda_2)/2;\) // parallel of latitude

The longitude can be normalised to \(-180\ldots+180\) using \((\text{lon}+540)\%360-180\)
3 The Am Home mobile application

3.1 Architecture Short Description

The amhome suite includes the mobile application and the web application. From the web application the user can login in [http://178.62.194.55/amhome/admin/](http://178.62.194.55/amhome/admin/) with his credentials and configure all the points of interest that would help the person to find his way to home and notify relatives about the position. Points like user house, relatives house or other can have information of names, telephones, useful messages and addresses. Also these points are grouped into categories like relatives category, home, drugstore etc. In addition the user can create new categories. The points of interest can be created at [http://178.62.194.55/amhome/admin/index.php/main#](http://178.62.194.55/amhome/admin/index.php/main#) through Location List view as explained in other section of this document.

The mobile application has the role of Geolocation tracking system. Furthermore, in order to notify the user about his location the tracking button has to be enabled. By this way the user can receive notification messages to the mobile application interface in order to know where he is. Furthermore, the user can search for the predefined points of interest like his home, relatives or drugstore addresses and to be guided to this points safely.

3.2 Technical Overview

3.2.1 Mobile application

The mobile application developed with Html5 and javascript code. The main Framework that used for code development was the Cordova javascript Framework in order to be easily deployed over different platform devices (cross platform) . Also in order to build the application used the intel xdk tool which offers development editor, designer and cross platform (hybrid) deployment capabilities.

The Intel® XDK development system enables software developers to develop, test, preview and deploy HTML5 web and hybrid apps. To get started, have to download and install the Intel XDK application onto a Linux, Microsoft Windows or Apple OS X development system.
The Intel XDK application consists of a set of development tools that helps coding, debugging, testing and building mobile web apps and hybrid HTML5 apps for multiple target platforms.

### 3.2.2 Mobile application Interface Overview

The mobile application interface is sliced into three views:

- **a)** the main view in which it is displayed the Google maps functionality like distance route
- **b)** the left sidebar menu in which can find the buttons which trigger actions like to find nearby locations, location map which lists all locations by category and add location setting which stores current location
- **c)** the right sidebar in which it can be enabled the current user location

It worth to mention that the mobile application interface designed with all appropriate rules and guides prior to offer the best user interface - UI experience to end user. Graphic design and typography are utilized to support its usability, influencing how the user performs certain interactions and improving the aesthetic appeal of the design; design aesthetics may enhance or detract from the ability of users to use the functions of the interface. The design process must balance technical functionality and visual elements (e.g., mental model) to create a system that is not only operational but also usable and adaptable to changing user needs. Many test has been operated in order to decide the structure of application views would have the best effect to end user. It must be mentioned that the application tested on various smart-phones and tablets like Vodafone 4 max with Android 4.4.4 version, Samsung Galaxy s4 with Android 4.3.1, iPhone 4s with Ios 7 and tablet Bitmore with Android 4.1.1 in order to measure the application response time and the overall look and feel of the mobile application design.

In following pages there are both some screenshots of the am home mobile application and the corresponding html5 code with some explanations. It is remarkable that having this fragments of code as a template is extremely easy to extend the views of the application by writing some extra code.
In the above picture is depicted the left sidebar of the am home mobile application in which appeared five main menu selections 1)The location map, 2)the nearby places, 3)the locations list and the most important 4)save my location element. The location map view shows an interactive map (google map) with the current location of the user and points of interest. The nearby places view lists all places that are nearby to the current user location of am home application and are registered to the back office of the system. Like an example maybe there are places like relatives or caregivers houses. In this view the user can see not only the location of registered points but also relative information to this like phone numbers and comments that the person maybe have made.

The structure of the html5 content for the aforementioned main parts and screenshots are as follows:
It worth to mention also that in order to be added active style into menu add class "active" to (<li class="active">) or add icon insert tag (span="icon-class-name") inside tag <li>. In addition the following script in files js/app.js find lines 4-40 this is used to response slidebar left menu for action:
window.onpopstate = function(event) {
  var hash = window.location.hash;
  $('.scrollable ul li').removeClass("active");

  if (hash === "#page_home") {
    $('#li_page_home').addClass("active");
    $('.loading').hide();
  } else if (hash === "#page_location_map") {
    $('#li_page_location_map').addClass("active");
    setTimeout('init()',500);
  } else if (hash === "#page_nearby") {
    $('#li_page_nearby').addClass("active");
    myGeoloc();
    getNearby();
  } else if (hash === "#page_location_list") {
    $('#li_page_location_list').addClass("active");
    getLocationCategory();
  } else if (hash === "#page_show_location") {
    $('.scrollable ul li').removeClass("active");
  } else if (hash === "#page_detail") {
    $('#li_page_location_list').addClass("active");
  } else if (hash === "#page_save_location") {
    listMyLocation();
  } else if (hash === "#page_detail_save_location") {
    $('#li_page_save_location').addClass("active");
  } else if (hash === "#page_add_location") {
    $('#li_page_save_location').addClass("active");
    $('#page_add_location').find('form')[0].reset();
  } else {
    // default link active
    $('#li_page_home').addClass("active");
    $('.loading').hide();
  }
};

Following is depicted the slidebar right menu to display a list of all button setting menu app.
These layers are developed with the support of google layers api . This is a less known feature of the Google Maps API is that it provides support for different kind of layers. With am home with just 1 click you can enable support for the Weather, Traffic, Transit, Bicycle & Panoramio layers (please note not all layers are available in all parts of the world). The weather layer on - that way dementia suffered person may see the current weather in the area. By turning the Transit layer your users will get information about the public transport in the local area. Due to recent changes in the google Maps API this feature was broken and it seems that the gmaps team is not interested in fixing it (at least not in the foreseeable feature). Now it is used a custom infoWindow that is able to expand depending on the amount of info in it. If the text is heigher than 350px, the user will see a scroll bar in the infowindow. In the past we had no chance to do that and the content in the infowindow was actually overflowing the height of the box. If someone want to monitor the usage of the Google Maps API you can provide an API key and get usage data. In October this year Google is going to roll out a visual refresh of the standard map style. We've added an option for it and it is turned on per default. If you don't like the new styles you can turn the visual ober.
After that the only way to use the old google maps visual style is to create a custom map style.

Here is depicted the main page in which the user can search for help. Information for relatives, caregivers, home and drugstores could be found easily tapping the large icons of this view or searching inserting text in search text box.

The structure of the content is as follows find lines 72-125:
- The following script in files js/app.js find lines 58-72 this is used to save user setting in local storage:

```javascript
47 // Load setting into local storage
48 for (var i = 0; i < settingArray.length; i++) {
49     getSetting = localStorage.getItem(settingArray[i]);
50     if (getSetting == "true") {
51         $('\#' + settingArray[i]).prop("checked", true);
52     }
53 }
54
56 // Save setting into local storage
57 $('.'+toggle-control').bind('click', function() {
58     var id = $(this).attr('id');
59     if($(this).prop('checked')){
60         localStorage.setItem(id, "true");
61     }else{
62         localStorage.setItem(id, "false");
63     }
64     // Reload map after click setting
65     init();
66     getNearby();
67 });
```

**Page Home**

Page to display the dashboard application.

- The structure of the content is as follows:

```html
<!-- PAGE HOME -->
<body>

</ul>
</div>
</div>
</div>
</div>
</div>
</div>
</ul>
</div>
</div>
<a id="nearby" transition="side" href="#page_nearby">
    <div class="fa fa-search-arrows-open">
        <strong>Nearby</strong>
    </div>
</a>
</li>
</ul>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>

Patra, February of 2016
To add icon insert tag div(data-icon="icon name") inside tag <a>, for list icon can see it in fonts/icomoon/list-icon.html.

**Page Location List**

![Page Location List](image)

Page to display the category of the page list location.

- The structure of the content is as follows:

```html
  <!-- PAGE LOCATION LIST -->
  <div id="page_location_list" data-role="page" data-title="Page Location List">
    <div class="wrapper">
      <ul id="list-category" class="list">
      </ul>
    </div>
  </div>
  <!-- END PAGE LOCATION LIST -->
```

- The following script in files js/maps.js find lines 282-292 this is used to load data from database:
In this picture is depicted all appropriate information that must be displayed to the user in order to find help.

function getLocationCategory()

$('#list-category').empty();
var url = urlRequest+'service/get_category';
getRequest(url, function(data) {
  var data = JSON.parse(data.responseText);
  for (var i = 0; i < data.length; i++) {
    $('#list-category').append('<li onclick="getLocationList('+data[i]['category_id']+')">' + data[i]['category_name'] + '</li>');</strong+'</span><span class="chevron"></span></a></li>');
})
}
Page Nearby

Page to display the nearest location that was around the user.

- The structure of the page is as follows:

```html
<!-- PAGE NEARBY -->
<div id="page_nearby" data-role="page" data-title="Page Nearby">
  <div class="wrapper">
    <ul id="list-nearby" class="list">
    </ul>
  </div>
</div>
<!-- END PAGE NEARBY -->
```

- The following script in files js/maps.js find lines 334-347 this is used to load data from database:

```javascript
var url = urlRequests.service.get_nearbyList+"\&myLoca=\"+myLoca+\"\&options=\"+distances;
getRequest(url, function(data){
  var data = JSON.parse(data.responseText);
  $('#list-nearby').empty();
  $('#list-nearby').append('<li class="list-dividers">Near "address"</li>');
  for(var i = 0; i < data.length; i++) {
    var logo = urlRequests.service.upload_logo+data[i]['markers_logos'];
    $('#list-nearby').append('<li class="list-dividers">DetailShowLocation("data[i]["markers_id"]")</li>');
    $('#list-nearby').append('<strong data[i]["markers_name"]">"</strong>"+" data[i]["markers_address"]"</p>');
    $('#list-nearby').append('<span class="distance">"data[i]["distance"]"">" distances +"</span>');
    $('#list-nearby').append('div class="img-box"');
    }
});
```

Page Save Location

In this page the user have to set up and store his current location by only tapping in to save button. After that the location is stored and the mobile application can use this location for the calculation of the distance. In addition to the location the user can add some comments that come along with the location in the view.
Page to display the data from the storage location of the user.

The html5 structure of the page is as follows:
The structure of the page is as follows:

```html
<!-- PAGE ADD SAVE LOCATION -->

<form>
  <input type="text" id="form-title" placeholder="Title">
  <textarea rows="3" id="form-desc" placeholder="Description"></textarea>
  <input id="form-add" class="button-primary button-block" href="#page_save_location" value="Add Entry">
  <input id="show-location-address" class="count"> No Detect Location</input>
</form>

<!-- END PAGE ADD SAVE LOCATION -->
```

Pages to store data where the location and a description of the current user.

- The following script in files js/maps.js find lines 378-390 this is used to load data local storage:

```javascript
function listMyLocation() {
  $('#list-save-location').empty();
  $('#loading').hide();
  $('#show-location-address').html('<h4>Near: *address*);
  for(var i in tkSaveLocation){
    var clt = JSON.parse(tkSaveLocation[i]);
    var date = new Date(clt.Dates * 1000);

    $('#list-save-location').append("<li><span onclick="listMyLocation("+i")">
      <strong>"+title+"</strong><p="date.toLocaleString()"></p>
      <a onclick="deleteMyLocation("+i")" class="button-negative">Delete</a></li>");
  }
}
```

Patra, February of 2016
- The following script in files js/maps.js find lines 361-375 this is used to save local storage:

```javascript
function addMyLocation(){
    var client = JSON.stringify({
        Title : ($('#form-title').val(),
        Desc : ($('#form-desc').val(),
        Lon : myLong,
        Lat : myLat,
        Location : address,
        Date : now
    });

    tbSaveLocation.push(client);
    localStorage.setItem('tbSaveLocation', JSON.stringify(tbSaveLocation));
    alert("The data was saved.");
    return true;
}
```

**Page Detail Save Location**

In the details save location page the user can retrieve the stored data geolocation and to see in the google map with some additional information that was provided.

- The structure of the page is as follows:
- The following script in files js/maps.js find lines 361-375 this is used to save local storage:

```javascript
function addMyLocation()
{
    var client = JSON.stringify({
        Title : $('#form-title').val(),
        Desc : $('#form-desc').val(),
        Lon : myLon,
        Lat : myLat,
        Location : address,
        Dates : now
    });

    tbSaveLocation.push(client);
    localStorage.setItem('tbSaveLocation', JSON.stringify(tbSaveLocation));
    alert("The data was saved.");
    return true;
}
```

Page Location Map

The page location map display the location data into map. All points of interest are depicted in this view with some markers. Also are draw the directions to the points of interest.
- The structure of the page is as follows:

```html
<!-- PAGE LOCATION MAP -->
<div id="page_location_map" data-role="page" data-title="Page Location Map">
  <div id="map_canvas" style="width: 100%;" data-snap-ignore="true"></div>
</div>
<!-- END PAGE LOCATION MAP -->
```

- The following script in files js/maps.js find lines 91-182 this is used load data map:
Page Detail Location

In this page are displayed the detail location information. First of all the category that belongs to that list with also the name of entity and the corresponding contact details like address and phone.
- The structure of the page is as follows:

```html
<!-- PAGE DETAIL LOCATION -->
<div id="page_show_location" data-role="page">
  <nav class="bar-standard">
    <ul class="segmented-controller">
      <li><a id="btn-show-map" href="#">Map</a></li>
      <li><a id="btn-show-street" href="#">Street</a></li>
      <li><a id="btn-show-directions" href="#">Directions</a></li>
      <li><a id="btn-show-route" href="#">Route</a></li>
    </ul>
  </nav>
  <div class="wrapper-bar">
    <div class="slider data-snap-ignore="true">
      <ul id="show-images">
        </ul>
    </div>
  </div>
  <div class="input-group">
    <div class="input-row">
      <label>Category</label>
      <div id="title-show-category"></div>
    </div>
    <div class="input-row">
      <label>Name</label>
      <div id="title-show-name"></div>
    </div>
    <div class="input-row">
      <label>Phone</label>
      <div id="title-show-phone"></div>
    </div>
    <div class="input-row">
      <label>Url</label>
      <div id="title-show-url"></div>
    </div>
    <div class="input-row">
      <label>Address</label>
      <div id="title-show-address"></div>
    </div>
    <div class="input-row">
      <label>Description</label>
      <div id="title-show-desc"></div>
    </div>
  </div>
</div>
<!-- END PAGE DETAIL LOCATION -->
```

- The following script in files js/maps.js find lines 412-442 this is used load detail location information:
Page Location Route

Page to display the location route.

- The structure of the page is as follows:
- The following script in files js/maps.js find lines 105-116 into function init() this is used show map route:

```javascript
// Request result directions maps
if (status == 1) {
    directionsDisplay.setMap(map);
    directionsDisplay.setPanel(document.getElementById('directions-route'));
} else {
    directionsDisplay.setMap(null);
    $('#directions_panel').empty();
}
```

### 3.3 Back office – Web Application

The back office system provides the administration side of the mobile application. In this side a person like a caregiver or the self person that suffers from dementia can administer the contact persons and generally the points of interest that a person want to have in the mobile application.

Following there is depicted the Login Page to login into the admin page-view. The source file (application/views/login.php).
- The structure of the page is as follows:

```html
<section class="main">
  <form class="form-login" action="login/login_check">
    <h1>Maps Finder App</h1>
    <label for="login"><i class="icon-user"></i>Username</label>
    <input type="text" id="user_name" name="user_name" placeholder="Username">
  </p>
  <label for="password"><i class="icon-eye-blocked"></i>Password</label>
  <input type="password" id="user_password" name="user_password" placeholder="Password" class="showpassword">
  <input type="reset" class="log-btn" value="Cancel">
  <input id="login" type="submit" name="submit" value="Log in">
</form>
</section>

- The following script find lines 42-67 this is used to do the processing login sistem:
2.2 Page Home

Page to display the location of all the data that has been input into the database, source file (aplication/views/home.php).

- The structure of the page is as follows:
Page Add Category

Page to login into the admin page, source file (application/views/category.php).

- The structure of the page is as follows:
The following script finds lines 208-226; this is used to perform the data request to the database:

```javascript
var oTable = $('#tables').dataTable({
    "bProcessing": false,
    "bServerSide": true,
    "sAjaxSource": "category/get",
    'sPaginationType': 'full_numbers',
    "fnServerData": function( sUrl, aoData, fnCallback ) {
        $.ajax({
            "url": sUrl,
            "data": aoData,
            "success": fnCallback,
            "dataType": "jsonp",
            "cache": false
        });
    }).columnFilter({
        // Set filter type
        aoColumns: [{ type: "text" },
                     { type: "text" }]
    });
```

### 3.3.2 PHP Structure
On the admin page application using CodeIgniter framework, in which to apply the concept of MVC.

1. Datatables

The application is equipped with a libraries datatables where just need to follow the example of its use as follows:

```
$result = $this->datatables->setData("Insert table name", "Insert column table", "Insert join table 1", "Insert join table 2", "Insert where data");

echo $result;
```

Sample use in category, source file (aplication/controllers/category.php):

```
// Get request data from datatables.

public function get()
{
     // Get data category
     $result = $this->datatables->getData("category", array("category_name", "category_name", "category_icon", "category_image", "category_id");
     echo $result;
}
```

2. Insert Data

Sample use insert data in category, source file (aplication/controllers/category.php):

Controllers
```php
public function insert()
{
    // Set upload folder
    $config['path'] = './upload/marker/';
    // Set images type
    $config['format'] = array('jpg', 'png', 'gif', 'bmp');
    // Set images size
    $config['size'] = '1024';
    // Load library
    $this->load->library('ajaxupload');
    $this->ajaxupload->getUpload($config, "category_marker");
    $query = $this->ajaxupload->query();
    // Cek images submit
    if($query['file_name'] == ''){
        $img = $this->input->post('category_marker_old');
    }else{
        $img = $query['file_name'];
    }

    $insert_id = $this->input->post('category_id');
    $data = array(
        'category_name' => $this->input->post('category_name'),
        'category_desc' => $this->input->post('category_desc'),
        'category_icon' => $this->input->post('category_icon'),
        'category_marker' => $img,
    );

    // Get process data insert to model category
    $this->load->model('category_model');
    $result = $this->category_model->insert($data, $insert_id);
    // Cek data insert or data update
    if(!$insert_id)
        if($result)
            echo "Data insert was successful!";
        else
            echo "Data insert not success!";
    else
        if($result)
            echo "Data update was successful!";
        else
            echo "Data update was successful!";
}
```
3. Remove Data

Sample use remove data in category, source file (aplication/controllers/category.php):

a. Controllers

```php
/*
Action insert or update
*/
function insert($data,$data_id)
{
    // Check id data, if empty id insert else update
    if ($data_id == '')
    {
        $result = $this->db->insert('category',$data);
        return $result;
    }
    else{
        $this->db->where('category_id', $data_id);
        $result = $this->db->update('category',$data);
        return $result;
    }
}
```

a. Models

```php
/*
Get action handle remove data.
*/
public function remove()
{
    $data_id = $this->input->post('remove_category_id');

    $this->load->model('category_model');
    $result = $this->category_model->remove($data_id);

    if($result)
        echo "Data remove was successful!";
    else
        echo "Data remove was successful!";
}
```
3.3 CSS Files and Structure

Front End

The theme comes with several CSS (stylesheet) files grouped in the CSS folder:

1. app.css - the main CSS styling for the app.
2. ratchet.css - CSS file containing the layout styling for the app.
3. icomoon.css - CSS file for icon app.
4. snap.css - CSS for sidebar menu app.
5. transition.min.css - CSS for transition page app.

Back End

The theme comes with several CSS (stylesheet) files grouped in the CSS folder:

1. style.css - the main CSS file containing the layout styling for the app styling for the admin.
2. chrome-bootstrap.css - CSS styling for the admin.
3. icomoon.css - CSS file for icon admin.
4. login.css - file containing the layout styling for the login admin.
5. tables.css - CSS for style datatables.

If would like to edit or change any color, font, or style of any elements in the template, must edit the general CSS file (style.css). If would like to change the general background for admin content for example, would change the following:

```css
body {
    font: 13px/20px 'Helvetica Neue', Helvetica, Arial, sans-serif;
    color: #404040;
    background-image: url(../images/retina_wood.png);
}
```
3.4 JavaScript

Front End

This app imports Four Javascript files. All JS files are placed in a separate folder called "js". The list of files is as follows:

1. app.js
2. maps.js
3. ratchet.js
4. snap.min.js
5. transition.min.js
6. jquery.min.js

All JS functions related to the app content management are placed in maps.js. The maps.js file is responsible for the main functions like connect to database, and etc

- For example - If calls synchronous requests GET and POST:

```javascript
function getRequest(url, callback) {
    $('#.loading').show();
    var request;
    if (WINDOW.XMLHttpRequest) {
        request = new XMLHttpRequest(); // IE7+, Firefox, Chrome, Opera, Safari
    } else {
        request = new ActiveXObject("Microsoft.XMLHTTP"); // IE6, IE5
    }
    request.onreadystatechange = function() {
        if (request.readyState == 4 && request.status == 200) {
            callback(request);
            $('#.loading').hide();
        }
    }
    request.open("GET", url, true);
    request.send();
}
```

Back End

This app imports Four Javascript files. All JS files are placed in a separate folder called "js". The list of files is as follows: map.js

1. map.location.js
2. jquery.min.js
3. jquery.form.js
4. jquery.dataTables.columnFilter.js
5. jquery.dataTables.min.js
6. jquery.dataTables.reload.js
3.5 Installation of Front End - Back End Web application

1. Server Requirements

- **PHP** version 5.1.6 or newer.
- A Database is required for most web application programming. Current supported databases are MySQL (4.1+), MySQLi, MS SQL, Postgres, Oracle, SQLite, and ODBC.

2. Upload Files

- Extract the main zip file, and then upload the application folder to the server for front end
- Extract the main zip file, and then upload the admin folder to the server for back end.

3. Configuration Files

Change this with the url path of this admin.

Open database.php file from folder application/config and edit these values:

```php
// Database Connection Settings
$db['default']['hostname'] = 'localhost';
$db['default']['username'] = 'root';
$db['default']['password'] = 'password';
$db['default']['database'] = 'maps';
$db['default']['dbdriver'] = 'mysql';
```

Change this with the database settings (host, user, password, database name, database driver).

4. Database Setup

- Create a database in mysql, I suggest using phpMyAdmin.
- In the phpMyAdmin, type the database name in the Create new database field, and click Create.
- After database created, click on the Import menu, browse install directory, select database/maps.sql (if want to install database + sample data).
- Click Go to import the sql file.
- The database should be installed now.

3.6 Database Structure

The database has 4 tables: category, markers, images and user.
1) Category

- category_id
- category_name
- category_icon
- category_marker
- category_desc

2) Markers

- markers_id
- markers_category_id
- markers_name
- markers_logo
- markers_phone
- markers_address
- markers_lat
- markers_lng
- markers_url
- markers_desc

3) Images

- images_id
- images_markers_id
- images_name
- images_url
- images_desc
- images_update

4) User

- user_id
- user_name
- user_password
- user_update

3.7 HTML Structure

Front End

On this application composed only of a single HTML document contain multiple "pages" that are loaded together by stacking multiple divs with a data-role of "page". Each "page" block have unique id (id="foo") that will be used to link internally
between "pages" (href="#foo"). When a link is clicked, the framework will look for an internal "page" with the id and transition it into view.

Here is an list of internal pages that exist in the index.html file:

1. Main
2. Sidebar Left Menu
3. Sidebar Right Menu
4. Page Home
5. Page Location List
6. Page Nearby
7. Page Save Location
8. Page Add Save Location
9. Page Detail Save Location
10. Page Location Map
11. Page Detail Location
12. Page Location Route

### 3.8 Administrator pages

Admin pages are built using the CodeIgniter framework with ajax technology to process login, load the page and processing the form.

A list of pages in admin:

1. Login
2. Page Home
3. Page Add Category
4. Page Add Location List
5. Page Add Location Images
6. Page Setting

**Front End Main page**

The common markup and data structures main as follows:
In order to add a new page can insert it into the div (class="content").

### 3.9 Using Google Maps Geocoding API

To provide the maps view and interaction in the amhome mobile application it was used the google maps api through Google Maps Web Services. The Google Maps Web Services are a collection of HTTP interfaces to Google services providing geographic data for the maps applications. Following there is some guide links that serves only to introduce the web services and host information common to all of the different services. Individual documentation for each service is located below:

**Google Maps Elevation API**

The Google Maps Elevation API provides elevation data for all locations on the surface of the earth, including depth locations on the ocean floor (which return negative values). In those cases where Google does not possess exact elevation measurements at the precise location you request, the service will interpolate and return an averaged value using the four nearest locations.
With the Google Maps Elevation API, you can develop hiking and biking applications, mobile positioning applications, or low resolution surveying applications.

You access the Google Maps Elevation API through an HTTP interface. Users of the Google Maps JavaScript API may also access this API directly by using the ElevationService() object. (See Elevation Service for more information.)

Google Maps Roads API

The Google Maps Roads API allows you to map GPS coordinates to the geometry of the road, and to determine the speed limit along those road segments. The API is available via a simple HTTPS interface, and exposes two services:

- **Snap to roads** This service returns the best-fit road geometry for a given set of GPS coordinates. This service takes up to 100 GPS points collected along a route, and returns a similar set of data with the points snapped to the most likely roads the vehicle was traveling along. Optionally, you can request that the points be interpolated, resulting in a path that smoothly follows the geometry of the road.

- **Speed limits** This service returns the posted speed limit for a road segment. The Speed Limit service is only available to Google Maps APIs Premium Plan customers. If you are an existing customer, you can contact your account manager or file a ticket in the Premium Plan support portal to enable the Google Maps Roads API.

Google Maps Time Zone API

The Google Maps Time Zone API provides time offset data for locations on the surface of the earth. Requesting the time zone information for a specific Latitude/Longitude pair will return the name of that time zone, the time offset from UTC, and the Daylight Savings offset.

Google Places API Web Service

Get data from the same database used by Google Maps and Google+ Local. Places features more than 100 million businesses and points of interest that are updated frequently through owner-verified listings and user-moderated contributions.
Google Maps Directions API
Google Maps Distance Matrix API
Google Maps Geocoding API
Google Maps Geolocation API
4 Pilot

The mobile application will undergo extensive trials with end user and deployed in a number of relevant environments. So, to provide sufficient information for mobile application efficiency and usability appropriate person with dementia took part in the trial. The participation of final users in this activity should mimic a situation as closest as possible to an actual deployment of the system on most of cases. To this end, external persons representing various target groups will be involved in the trials.

For one month a pilot-person with dementia carried his mobile phone in a custom pocket with installed the am home mobile application. The am home mobile application was open continuously during this pilot. As an outcome of this was to track the position of the user and to have a couple of notifications alerts during the phase of the trial.

The am home generally worked as expected and when the application identified that the person moved away from home (for the needs of the trial the radius was equal with 100 meters) notifications like emails send to three relatives that have also participated in the trial .All participants have registered their mails to am home back office.

Problems during trial:

One of the main problems that identified during the phase of the trial was the problem that a person has to charge the mobile phone because of the energy consumption of the application (battery level).

Another problem was that the mobile application have to be enabled continuously or running in the background. In that way is difficult to keep awake the application. So in order to solve that problem a healthy person has to care about the application openness.

The person that used the application found the am home difficult to use because of the bad visibility that had in the screen. So a mobile smart phone with 6 inch (vodafone T max model) screen replaced that with 3.5 inch (Samsung galaxy) in order to improve the situation.

Regarding the geolocation accuracy several points used in order to measure the effectiveness of wifi and 3G Geolocation position in rural and city center points. The measurement deduced with the support of http://andygup.net/samples/html5geo/ website that provides the accuracy of position in meters.
5 Future Development & Optimizations

Software optimization

As technology make bigger steps into communication protocols and as these protocols became mature there is the need of use them. Such a transfer protocol is MQ Telemetry Transport Protocol MQTT which have the potential to replace HTTP\(^{[57]}\) in our application in the future. That will happens because in the mobile environment, response times, throughput, lower battery use and lower bandwidth are key design criteria. Compared with HTTP, MQTT features faster response and throughput, and lower battery and bandwidth usage, making it well suited to use cases where:

- connectivity is intermittent
- bandwidth is at a premium
- an enterprise application needs to interact with one or more phone apps
- phone or tablet apps need to send data reliably without requiring code retry logic

Another advantage of MQTT over HTTP is that it is integrated with enterprise messaging middleware, so it works with enterprise-level applications that push data to mobile apps. MQTT can also be integrated with IBM Worklight in such a way that developers can create mobile applications using HTML and Javascript and yet have the messaging function working at the native layer, in native Java code, deployed on Android.

Basically, MQTT is designed for low latency, assured messaging and efficient distribution. HTTP is not optimized for low power usage or minimizing the amount of bytes flowing.

As the hybrid application lack in performance an improvement could be to develop the application in native code. Building native applications means using the native language of the platform, Objective-C on iOS, and Java on Android. The main advantage of native applications is their performance. Native apps are compiled into machine code (Dalvik byte code under Android), which gives the best performance can get from the mobile phone. Best performance includes fast and fluid animations as well as full access to phone hardware, multi touch support and the latest APIs. Native development is far from easy. Despite the great number of resources that can be found, it may not be understandable to everyone. As code must be written specifically for each platform, the same code will have to largely be rewritten with little able to be
The logic may be the same, but the language, APIs and the development process is different. This process can be relatively long for complex applications.

**Hardware Optimization**

The premise of the Internet of Things (IoT) is that this new technology trend will connect billions of devices using the internet starting around 2020, with ecosystems that will address wearables, smart home, automotive, smart cities, the workspace and industrial applications. The IoT system consists of three domains: **Sensors, Connectivity and Applications**. Most attention for IoT has been focused on the applications for the home (consumer), transport (mobility), health (body), buildings (infrastructure), factory (industrial) and cities (utilities, security). What is missing from much of the discussion are the underlying hardware and sensor technologies that enables the IoT applications, intelligence and links to the 'cloud'.

The size, weight, power and cost (SWaP-C) demands for the IoT ecosystems will force the creation of a new paradigm for the hardware. These metrics must be improved by factors of 10 to 100 in order to make IoT realizable. If today’s hardware costs $10 a piece for a 100 million device market, then the same function may have to be well under $1 to address a 20 to 50 billion device market. Most electrical devices today have ready access to prime power to energize their circuits. In the IoT world, there are many use cases where connecting the device to the wall outlet or charging the battery is a showstopper. Therefore, improved power efficiency, smart power management, energy harvesting and wireless power transmission will all need to be investigated and made viable for IoT applications. In today’s hardware, milliwatt dissipation may be sufficient. In the IoT world of 2020, microwatts or even nanowatt power dissipation will be required. In many sensor applications, the IoT device must operate at a very low duty cycle; waking up for milliseconds to perform its function, transmit its data payload and then go back to sleep.

The good news is that the advanced silicon CMOS technologies being developed today in the world's leading foundries feature sizes ranging from 32nm down to 10nm. The design of the next generation of low-power RF transceivers, mixed-signal ADCs/DACs and micro-controllers will not be easy nor is first-pass design success assured. Even more challenging will be the design and fabrication of packaging, interconnects and PWB to meet the same IoT metrics. EDA CAD tools must also evolve to design, simulate and lay out the highly integrated microsystems and IoT System-on-a-Chip (SOCs) realizations. There has been much discussion about the end of Moore’s Law as we approach 10nm geometries, but this may be overstated.
So as hardware improves maybe there is place for a wearable system that tracks the Geolocation of a user and embed the whole application as it is in one single micro IoT wearable capable to offer the overall functionality that offers the mobile application.
Appendix

Terminology

HTTP : The Hypertext Transfer Protocol (HTTP) is an application protocol for distributed, collaborative, hypermedia information systems. HTTP is the foundation of data communication for the World Wide Web. Hypertext is structured text that uses logical links (hyperlinks) between nodes containing text. HTTP is the protocol to exchange or transfer hypertext. The standards development of HTTP was coordinated by the Internet Engineering Task Force (IETF) and the World Wide Web Consortium (W3C), culminating in the publication of a series of Requests for Comments (RFCs). The first definition of HTTP/1.1, the version of HTTP in common use, occurred in RFC 2068 in 1997, although this was obsoleted by RFC 2616 in 1999.

MQTT : MQTT (formerly MQ Telemetry Transport) is an ISO standard (ISO/IEC PRF 20922) publish-subscribe based "light weight" messaging protocol for use on top of the TCP/IP protocol. It is designed for connections with remote locations where a "small code footprint" is required or the network bandwidth is limited. The publish-subscribe messaging pattern requires a message broker. The broker is responsible for distributing messages to interested clients based on the topic of a message. Andy Stanford-Clark and Arlen Nipper of Cirrus Link Solutions authored the first version of the protocol in 1999.

VPN : A virtual private network (VPN) extends a private network across a public network, such as the Internet. It enables users to send and receive data across shared or public networks as if their computing devices were directly connected to the private network, and thus are benefiting from the functionality, security and management policies of the private network. A VPN is created by establishing a virtual point-to-point connection through the use of dedicated connections, virtual tunnelling protocols, or traffic encryption.

Native applications : Building native applications means using the native language of the platform, Objective-C on iOS, and Java on Android.

Hybrid applications : Building hybrid applications means using cross platform languages like html5 and javascript.
IoT: The Internet of Things (IoT) is the network of physical objects—devices, vehicles, buildings and other items embedded with electronics, software, sensors, and network connectivity—that enables these objects to collect and exchange data. The Internet of Things allows objects to be sensed and controlled remotely across existing network infrastructure, creating opportunities for more direct integration of the physical world into computer-based systems, and resulting in improved efficiency, accuracy and economic benefit; when IoT is augmented with sensors and actuators, the technology becomes an instance of the more general class of cyber-physical systems, which also encompasses technologies such as smart grids, smart homes, intelligent transportation and smart cities. Each thing is uniquely identifiable through its embedded computing system but is able to interoperate within the existing Internet infrastructure. Experts estimate that the IoT will consist of almost 50 billion objects by 2020.

Beacon Networks: A beacon is a small Bluetooth radio transmitter. It’s like a lighthouse: it repeatedly transmits a single signal that other devices can see. Instead of emitting visible light though, it broadcasts a radio signal that’s made up of a combination of letters and numbers transmitted on a regular interval of approximately 1/10th of a second. A Bluetooth-equipped device like a smartphone can “see” a beacon once it is in range, much like sailors looking for a lighthouse to know where they are.

Pythagorean theorem: In mathematics, the Pythagorean theorem, also known as Pythagoras's theorem, is a relation in Euclidean geometry among the three sides of a right triangle. It states that the square of the hypotenuse (the side opposite the right angle) is equal to the sum of the squares of the other two sides. The theorem can be written as an equation relating the lengths of the sides a, b and c, often called the "Pythagorean equation": \( a^2 + b^2 = c^2 \),

Equirectangular projection definition: Given a spherical model,

\begin{align}
  x & = \lambda \cos \varphi_1 \\
  y & = \varphi
\end{align}

where

\( \lambda \) is the longitude;
φ is the latitude;

φ1 are the standard parallels (north and south of the equator) where the scale of the projection is true;

x is the horizontal position along the map;

y is the vertical position along the map.

The point (0,0) is at the center of the resulting projection. The plate carrée (French, for square plate), is the special case where φ1 is zero. This projection maps x to be the value of the longitude and y to be the value of the latitude, and therefore is sometimes called the latitude/longitude or lat/lon(g) projection or is said (erroneously) to be “unprojected”. While a projection with equally spaced parallels is possible for an ellipsoidal model, it would no longer be equidistant because the distance between parallels on an ellipsoid is not constant.

Source Code

The overall system source code is provided at http://178.62.194.55/sourcecode.zip.

Bibliography of sources

1. Dementia
2. WorldAlzheimerReport
3. eHealth for people with dementia in home-based and residential care
4. Geolocation 101: How It Works, the Apps, and The Privacy
   b. Review of ICT-based services for identified unmet needs in people with dementia
5. Intel xdk online documentation
6. Intel forum
7. HTML5 geolocation
8. Stackoverflow.com
9. Ratchet
   http://maker.github.io/ratchet/
10. jQuery
    http://jquery.com/
11. Snap JS
    https://github.com/jakiestfu/Snap.js/
12. Datatables
    http://datatables.net
13. Jquery Form
    http://malsup.com/jquery/form/
14. Intel XDK
15. Cordova
16. Google Maps Api
17. Icomoon
    http://icomoon.io/
18. Patterns
    http://subtlepatterns.com
19. Map Icons
    http://www.webiconset.com/map-icons/
20. Notifications
21. iWander: An Android Application for Dementia Patients
22. Geolocation
    https://en.wikipedia.org/wiki/Geolocation
27. David Leveille (2014-08-29). "This hobbyist-turned-journalist is a one-man digital detective agency". PRI.
33. Geolocation specification API https://www.w3.org/TR/geolocation-API/
37. Calculate distance, bearing and more between Latitude and Longitude http://www.movable-type.co.uk/scripts/latlong.html
38. Top Seven iPad Apps for People With Dementia https://myageingparent.com/technology/communication/top-six-ipad-apps-for-people-with-dementia/
39. 5 Good Apps For Dementia Sufferers http://fanappic.com/5-good-apps-for-dementia-sufferers/

78