mEYEtrak

Requirements Document

Breaking down barriers between carriers

• Ernie Costa • Michael Daniels • Lindsay Graham • Erik Olson • Dion St. Hilaire •

Table of Contents

1 System Context and Scope

1.1 Abstract

The product mEYEtrak is a non-biased inter-provider mobile application to quickly communicate and locate friends in a secure, sociable manner. Users will have the ability to add buddies and transmit information between each other through a central server, providing a controllable means of securing that data.

1.2 Project Goals

While a few providers already offer a similar service, none of them are free, let alone independent of the originating brand. Boost Mobile users can only do this between each other on stock phones. Apple subscribers have iPhone programs that can email friends their GPS location via Google Maps, but not alert preset buddies when they're within a set distance.

Our project aims to conjoin all these abilities together under the same roof regardless of carrier. Currently, The Pink Tacos aim to develop the proof of concept on the Windows Mobile platform with plans to expand to other environments (J2ME, iPhone, etc) in the near future.

1.3 Features

1.3.1 Buddy List

Revision: 0.1 (9-14-2008)

Description: Users have the ability to add and remove other subscribers of this service to their list of “buddies”. This serves as the primary means of communication between users. Buddy Lists will also have the ability to add specified users into groups set and named by the user (see feature description below).

1.3.2 Buddy List Grouping

Revision: 0.1 (9-14-2008)

Description: Users can add and remove buddies from their Buddy List into definable groups. This feature serves as a means of allowing for user-customized privacy settings for various groups of people.

1.3.3 Privacy Settings

Revision: 0.1 (9-14-2008)

Description: There are three privacy settings: *public*, *semi-private*, and *private*. A specific setting can be global or per-group basis. *Public* and *semi-private* can both trigger the *Friend in Range* event while *private* will not. The difference between *public* and *semi-private* is that the latter requires user acceptance of a location request while *public* automatically accepts the request.

1.3.4 Friend in Range

Revision: 0.1 (9-14-2008)

Description: When a user’s buddy enters within a user-defined radius of his/her location, an alert will be generated allowing the user to request location. Privacy settings can influence this decision for the user.

1.3.5 Request Friend Location

Revision: 0.1 (9-14-2008)

Description: A user has the ability to send a request to any user on his/her buddy list asking for the buddy's exact location.

1.3.6 Send Friend Location

Revision: 0.1 (9-14-2008)

Description: A user has the ability to send a message to any user on his/her buddy list containing their exact location.

1.3.7 User Status

Revision: 0.1 (9-14-2008)

Description: A user can update their status to be viewed on their friend's Buddy Lists, indicating mood, location, etc.

1.3.8 View Map

Revision: 0.1 (9-14-2008)

Description: When a buddy returns their location from a request, the user has the ability to open a map displaying their approximate location (civilian GPS is accurate to within 15 meters)[1]. All buddies that are *public* to the user can be placed on a single map.

1.3.9 Stolen Phone Trak

Revision: 0.1 (9-14-2008)

Description: If a user's phone ever was stolen, he/she can log into the website and remotely force the phone to log-off, preventing any sending or requesting of information to other users. Meanwhile, the server will query the phone periodically for an up-to-date location to track the phone.

2 Requirements

2.1 About Requirements

2.1.1 Priorities

🌍 (1) – Requirement has little impact on the overall usefulness of the product. Failure to meet this requirement will cause at most a mild inconvenience.

🌍🌍 (2) – Requirement is required for a secondary feature. Failure to meet this requirement will make the product slightly less convenient, but most of the functionality will still exist.

🌍🌍🌍 (3) – Requirement is necessary for a moderately important feature. Failure to meet this requirement will make the product much less convenient, but basic functionality will still exist.

🌍🌍🌍🌍 (4) – Requirement is necessary in order for key features to be functional. Failure to meet this requirement will cripple the product and most functionality will be lost.

[1] http://www8.garmin.com/aboutGPS/

🌍🌍🌍🌍🌍 (5) – Requirement is essential to the functionality of the product. If this requirement is not fulfilled, the product will be unusable.

2.2 Account Information Requirements

Note: This section is not in the standard requirement form (i.e. risk and priority are omitted). All of the specified account information is essential to the functionality of the product.

2.2.1 Login Information

2.2.1.1 *Email Address*

Revision: 1.0 (09.27.2008)
Description: The user's email address is the account key and will be used for account login.

2.2.1.2 *Password*

Revision: 1.0 (09.27.2008)
Description: The user's password must be a minimum of 8 characters, must contain at least one of each of the following: lower case letter, upper case letter, number.

2.2.2 Personal Information

2.2.2.1 *Full Name*

Revision: 1.0 (09.27.2008)
Description: The user's first name, last name, middle initial, and suffix.

2.2.2.2 *Mobile Phone Number*

Revision: 1.0 (09.27.2008)
Description: The user's phone number of the mobile device that will run the client application.

2.2.2.3 *Mobile Phone Provider*

Revision: 1.0 (09.28.2008)
Description: The provider of the user's mobile phone.

2.2.2.4 *Location*

Revision: 1.0 (09.27.2008)
Description: The user's most up to date GPS coordinates along with the time of last location synchronization.

2.2.3 Personal Settings

2.2.3.1 *Range Radius*

Revision: 1.1 (09.28.2008)
Description: A buddy is considered in range if and when the distance between the user and the buddy is less than the range radius. This is a measure of distance. It is specific to each buddy on the buddy list, but there will be a default value that can be specified by the user.

2.2.3.2 *Enabled Alerts*

Revision: 1.0 (09.27.2008)

Description: Alerts, such as Buddy in Range and Location Request, can be enabled and disabled as a convenience.

2.2.3.3 *Buddy List*

Revision: 1.0 (09.28.2008)
Description: A list of users that the user monitors. Each buddy on the buddy list will be assigned a range radius.

2.2.4 Privacy Settings

2.2.4.1 *Privacy Choices*

Revision: 1.0 (09.27.2008)
Description: The following privacy options are available:

1. Public – The user's exact location is available without confirmation.
2. Semi-private – The user's location can trigger a Buddy in Range notification, but a Location Request will have to be approved in order for the exact location to be known by a buddy.
3. Private – The user's location is hidden and will not trigger a Buddy in Range notification. Buddies can still submit a Location Request.

2.2.4.2 *Buddy List Group*

Revision: 1.0 (09.27.2008)
Description: Each group is assigned a privacy setting that applies to all buddies in that group.

2.3 Client Application Requirements

2.3.1 System Requirements

2.3.1.1 *Internet Connectivity*

Revision: 0.1 (09.27.2008)
Description: The client phone must have an internet connection.
Risk: Without an internet connection, the client application will be unable to connect with the server. This would make the application completely nonfunctional.
Priority: 🌍🌍🌍🌍🌍 (5)

2.3.1.2 *GPS Capability*

Revision: 0.1 (09.27.2008)
Description: The client phone is required to have a Global Positioning System (GPS).
Risk: If this requirement is not met, then the position of the client cannot be known. The client would not be able to synchronize its location with the server, but would still be able to request the location of buddies.
Priority: 🌍🌍🌍🌍 (4)

2.3.1.3 *Mobile Environment*

Revision: 0.1 (09.27.2008)
Description: The client phone must be running Windows Mobile 6.
Risk: If this requirement is not met, then the application will not be able to install on the mobile device.
Priority: 🌍🌍🌍🌍🌍 (5)

2.3.1.4 *SMS Enabled*

Revision: 0.1 (09.27.2008)
Description: The client phone must have SMS capabilities.
Risk: If this requirement is not met, then the validation process will not work.
Priority: 🌍🌍🌍🌍🌍 (5)

2.3.2 Functional Requirements

2.3.2.1 *Run conditions*

Revision: 1.0 (09.27.2008)
Description: The client application should start when the phone is turned on. It should automatically login, and then lock the application. The user will need to enter their password to close the application or to unlock the application.
Risk: If this requirement is not met, then Stolen Phone Trak will not be effective if the phone is turned off and then back on.
Priority: 🌍🌍🌍 (3)

2.3.2.2 *Contact List Access*

Revision: 1.0 (09.27.2008)
Description: The client must be able to access the contact list on the host.
Risk: If this requirement is not met, will not be able create and sync original buddy list.
Priority: 🌍🌍🌍🌍🌍 (5)

2.3.2.3 *Current GPS Location*

Revision: 1.0 (09.27.2008)
Description: The client must be able to access the host's GPS chip.
Risk: If this requirement is not met, the location system will not work.
Priority: 🌍🌍🌍🌍🌍 (5)

2.3.2.4 *Current Cell Tower Location*

Revision: 1.0 (09.27.2008)
Description: The client must be able to access the cell tower triangulation function of the host.
Risk: If this requirement is not met, the secondary location system will not work.
Priority: 🌍🌍🌍🌍🌍 (5)

2.3.2.5 *Google Maps*

Revision: 1.0 (09.27.2008)
Description: The client must be able to integrate with Google Maps.
Risk: If this requirement is not met, the graphical location of buddies will not be able to be displayed.
Priority: 🌍🌍🌍🌍🌍 (5)

2.3.2.6 *Update Location*

Revision: 1.0 (09.27.2008)
Description: The client application must be able to update its location on the server.
Risk: If this requirement is not met, locations syncing will not be able to occur.
Priority: 🌍🌍🌍🌍🌍 (5)

2.3.2.7 *Fetch Buddy List*

Revision: 1.0 (09.27.2008)

Description: The client must be able to fetch buddy list and corresponding locations from the server.

Risk: If this requirement is not met, the system will not be able to determine which buddies are in range

Priority: 🌍🌍🌍🌍🌍 (5)

2.3.2.8 *Add Buddy*

Revision: 1.0 (09.27.2008)

Description: The client must be able to add a buddy via phone number or email address.

Risk: If this requirement is not met, the user will only be able to locate users in the host's contact list.

Priority: 🌍🌍🌍🌍🌍 (5)

2.3.2.9 *Remove Buddy*

Revision: 1.0 (09.27.2008)

Description: The client must be able to integrate remove a buddy.

Risk: If this requirement is not met, the user will not be able to remove unwanted buddies. This would be a major inconvenience.

Priority: 🌍🌍🌍🌍 (4)

2.3.2.10 *Confirm Buddy*

Revision: 1.0 (09.27.2008)

Description: The client must be able to confirm a buddy request via the client application.

Risk: If this requirement is not met, buddies would never be able to be added, and the graphical location of buddies will not be able to be displayed.

Priority: 🌍🌍🌍🌍🌍 (5)

2.3.2.11 *Request Buddy Location*

Revision: 1.0 (09.27.2008)

Description: The client must be able to request the location of a buddy.

Risk: If this requirement is not met, the user will be unable to request the location of a buddy.

Priority: 🌍🌍🌍🌍🌍 (5)

2.3.2.12 *Buddy in Range Alert*

Revision: 1.0 (09.27.2008)

Description: The client must be able to alert the user when a buddy is in range.

Risk: If this requirement is not met, the user will not know when a buddy is in range.

Priority: 🌍🌍🌍🌍🌍 (5)

2.3.2.13 *Location Request Alert*

Revision: 1.0 (09.27.2008)

Description: The client must be able to alert the user when a buddy requests the user's location.

Risk: If this requirement is not met, the user will be unable to request the location of a buddy.

Priority: 🌍🌍🌍🌍🌍 (5)

2.3.2.14 *Stolen Phone Alert*

Revision: 1.0 (09.27.2008)

Description: A Stolen Phone Alert can be sent via the web interface. The client application needs to receive this alert and take the following actions (silently):

1. Lock the application – This will prevent the user from accessing any of the features of the client application.
2.
3. Continue synchronizing the GPS location.
4. Prevent the application from being closed.
5. Ensure the application will be started if the phone is turned off and back on.

Risk: If this requirement is not met, then the Stolen Phone Trak feature will not be functional.
Priority: 🌍🌍🌍🌍 (4)

2.3.2.15 *Range Bar*

Revision: 1.0 (09.27.2008)
Description: Buddies that are inside of the specified range radius will be displayed in a graphical, linear form ordered by distance. The scale of this graph will be determined by the range radius. Each user will be displayed with information (i.e. name, number, picture, and etcetera).
Risk: This is a bonus requirement. If it is not completed, then the user will have a less enjoyable experience, but will not lose any key functionality.
Priority: 🌍🌍 (2)

2.3.2.16 *Account Information Modification*

Revision: 1.0 (09.27.2008)
Description: The user will be able to modify all information and settings in the Account Information Requirements section with the following exceptions: email address, location.
Risk: If this requirement is not met, then the user will not be able to modify their settings/information via their phone. This would be a major inconvenience.
Priority: 🌍🌍🌍🌍 (4)

2.3.3 Non-functional Requirements

2.3.3.1 *Easily Learnable*

Revision: 1.0 (09.27.2008)
Description: The client application should be easy to learn. The average user should be able to begin using the application and quickly familiarize ~~himself~~ his/her self with basic features without needing to read a manual.
Risk: Products that are overly complicated and difficult to use frustrate customers and tend to drive them away from the product.
Priority: 🌍🌍 (2)

2.3.3.2 *Efficient*

Revision: 1.0 (09.27.2008)
Description: The client application should use as few resources as possible. The average battery life of the phone should not be heavily impacted by the use of the application.
Risk: If the client application uses too many resources, then the battery life of the phone will decrease drastically. This will cause an inconvenience to users.
Priority: 🌍🌍🌍 (3)

2.4 Web Interface Requirements

2.4.1 System Requirements

2.4.1.1 *Compatible Web Browser*

Revision: 1.0 (09.27.2008)

Description: The client's access to the website should be as streamlined and productive as possible. In turn, newer technologies must be in use, namely Firefox 3 or Internet Explorer 7. While other browsers exist that replicate the features of these, we cannot be guarantee other products will work correctly.

Risk: If the client cannot view the website in a compatible browser, their user experience will be severely crippled. In some cases, functionality may be completely lost.

Priority: 🌍🌍🌍🌍 (4)

2.4.1.2 *Internet Connection*

Revision: 1.0 (09.27.2008)

Description: Without an internet connection, the user will not be able to log into the website to edit and maintain their account. It is also recommended that a broadband internet connection is used. While there won't be bandwidth intensive content on the website, the client should have some form of high-speed internet connection to experience the full capabilities of the system.

Risk: Without the ability to log into the website, the client will not be able to use features exclusive to it, such as Stolen Phone Trak.

Priority: 🌍🌍🌍🌍🌍 (5)

2.4.2 Functional Requirements

2.4.2.1 *Google Maps*

Revision: 1.0 (09.27.2008)

Description: The website must be able to integrate with Google Maps through the available API.

Risk: If this requirement is not met, the graphical location of buddies will not be able to be displayed.

Priority: 🌍🌍🌍 (3)

2.4.2.2 *Fetch Buddy List*

Revision: 1.0 (09.27.2008)

Description: The web interface must be able to fetch buddy list and corresponding locations from the server.

Risk: If this requirement is not met, the system will not be able to determine which buddies are in range.

Priority: 🌍🌍🌍🌍🌍 (5)

2.4.2.3 *Add Buddy*

Revision: 1.0 (09.27.2008)

Description: The user must be able to add a buddy via phone number or email address using the web interface.

Risk: If this requirement is not met, then the user will only be able to add buddies through the client application.
Priority: 🌍🌍🌍 (3)

2.4.2.4 *Remove Buddy*

Revision: 1.0 (09.27.2008)
Description: The user must be able to remove a buddy via the web interface.
Risk: If this requirement is not met, the user will only be able to remove buddies via the client application.
Priority: 🌍🌍🌍 (3)

2.4.2.5 *Confirm Buddy*

Revision: 1.0 (09.27.2008)
Description: The client must be able to confirm a buddy request via the web interface.
Risk: If this requirement is not met, then the user will only be able to confirm buddies via the client application.
Priority: 🌍🌍🌍 (3)

2.4.2.6 *Request Buddy Location*

Revision: 1.0 (09.27.2008)
Description: The client must be able to request the location of a buddy.
Risk: If this requirement is not met, the user will only be able to send a Location Request via the client application.
Priority: 🌍🌍🌍 (3)

2.4.2.7 *Buddy in Range Alert*

Revision: 1.0 (09.27.2008)
Description: The client must be able to alert the user when a buddy is in range.
Risk: If this requirement is not met, the user will not know when a buddy is in range via the web interface.
Priority: 🌍🌍🌍 (3)

2.4.2.8 *Location Request Alert*

Revision: 1.0 (09.27.2008)
Description: The client must be able to alert the user when a buddy requests the user's location.
Risk: If this requirement is not met, the user will only be able to view and approve Location Requests via the client application.
Priority: 🌍🌍🌍 (3)

2.4.2.9 *Range Bar*

Revision: 1.0 (09.27.2008)
Description: Buddies that are inside of the specified range radius will be displayed in a graphical, linear form ordered by distance. The scale of this graph will be determined by the range radius. Each user will be displayed with information (i.e. name, number, picture, and etcetera).
Risk: This is a bonus requirement. If it is not completed, then the user will have a less enjoyable experience, but will not lose any key functionality.
Priority: 🌍🌍 (2)

2.4.2.10 *Account Information Modification*

Revision: 1.0 (09.27.2008)

Description: The user will be able to modify all information and settings in the Account Information Requirements section with the following exceptions: email address, location.

Risk: If this requirement is not met, then the user will not be able to modify their settings/information via the web interface.

Priority: 🌍🌍🌍🌍 (4)

2.4.2.11 *Stolen Phone Track Alert*

Revision: 1.0 (09.27.2008)

Description: The user will be able to send a Stolen Phone Trak Alert from the web interface.

Risk: If this requirement is not met, then the Stolen Phone Trak feature will not be functional.

Priority: 🌍🌍🌍🌍 (4)

2.4.3 Non-functional Requirements

2.4.3.1 *Easily Learnable*

Revision: 1.0 (09.27.2008)

Description: The website should be easy to learn. The average user should be able to begin using the application and quickly familiarize ~~himself~~ his/her selfwith basic features without needing to read a manual.

Risk: Products that are overly complicated and difficult to use frustrate customers and tend to drive them away from the product.

Priority: 🌍🌍 (2)

2.4.3.2 *Fast Loading*

Revision: 1.0 (09.27.2008)

Description: The website should be rendered quickly in the user's browser, allowing for quick navigation without waiting for the page to load.

Risk: Websites that take their time loading tend to frustrate customers and drive them away from the product.

Priority: 🌍🌍 (2)

2.4.3.3 *Live Updates*

Revision: 1.0 (09.27.2008)

Description: The website should alert the user whenever a buddy changes his or her status. An example of is when a user comes online and changes status, a popup will inform the user of the change.

Risk: Without Live Updates, the user would have to constantly refresh the website to see any changes in status, etc.

Priority: 🌍🌍 (2)

2.5 Server Application Requirements

2.5.1 System Requirements

2.5.1.1 *Microsoft Windows Server*

Revision: (09.28.2008)

Description: The server requires a recent version of Windows Server, preferably 2008.

Risk: If the server does not have Windows Server installed on it, some functionality may not work properly, namely concerning the requirements below.

Priority: 🌍🌍 🌍 🌍 (4)

2.5.1.2 *Microsoft .NET Framework 3.5*

Revision: (09.28.2008)

Description: The server will need the .NET 3.5 framework installed to run.

Risk: Since the server application will be coded in C# and using the most up-to-date libraries available, it will not run unless .NET 3.5 is installed

Priority: 🌍🌍 🌍🌍 🌍 (5)

2.5.1.3 *Microsoft SQL Server 2005*

Revision: (09.28.2008)

Description: The server is using SQL Server 2005 to handle its database needs.

Risk: If the server doesn't have SQL Server 2005 installed, then there will be no means of storing data in a manageable means.

Priority: 🌍🌍🌍🌍🌍 (5)

2.5.1.4 *Internet Connection*

Revision: (09.28.2008)

Description: An internet connection is required to communicate between clients and the server

Risk: Without an internet connection, there will be no means of sending the data back and forth, crippling the service.

Priority: 🌍🌍🌍🌍🌍 (5)

2.5.2 Functional Requirements

2.5.2.1 *Account Updates*

Revision: 1.0 (09.28.2008)

Description: The server application should handle updates made by the client application and the web interface (this includes GPS location made silently by the client application).

Risk: If this requirement is not met, then the user will not be able to make updates to their account, and the location of the user will never be updated.

Priority: 🌍🌍🌍🌍🌍 (5)

2.5.2.2 *Alerts*

Revision: 1.0 (09.28.2008)

Description: Alerts are triggered based on the specified conditions. When an alert is triggered, it is sent to any and all open sessions that an account has (client application or web interface).
Risk: If this requirement is not met then notifications will not be sent to the client application or the web interface when a user is logged in. Stolen Phone Trak, Buddy in Range, and Location Requests would not be functional.
Priority: 🌍🌍🌍🌍 (4)

~~2.5.2.2.1~~2.5.2.3 *Buddy in Range Alert*

Revision: 1.0 (09.28.2008)
Description: This alert is triggered under the following condition(s): 1) A buddy has a GPS location that is less than the range radius from the user's GPS location; and 2) The Buddy in Range Alert is enabled under the user's settings. Information included in the alert: 1) Alert type; 2) Timestamp; 3)User receiving the alert; 4) The user (buddy) in range; 5) The user specified range radius for that buddy.
Risk: If this requirement is not met then a user will never be notified if a friend is in range, or may be alerted even if the user has chosen to ignore these alerts by changing the account setting.
Priority: 🌍🌍🌍🌍 (4)

~~2.5.2.2.2~~2.5.2.4 *Stolen Phone Trak Alert*

Revision: 1.0 (09.28.2008)
Description: Information included in the alert: 1) Alert type; 2) Timestamp; and 3) User. This alert has the following exception(s): 1) This alert is only sent to the client application; 2) If the client application does not have an active session open, then the alert is added to a queue and is sent when the client application opens a session; 3) The alert will grow stale after 1 week.
Risk: If this requirement is not met, then the Stolen Phone Trak function will not work.
Priority: 🌍🌍🌍🌍 (4)

~~2.5.2.2.3~~2.5.2.5 *Location Request Alert*

Revision: 1.0 (09.28.2008)
Description: This alert is triggered under the following condition(s): 1) A buddy requests the user's location; and 2) the Location Request Alert is enabled in the user's settings. Information included in the alert: 1) Alert type; 2) Timestamp; 3) Requesting user; and 4) Target user.
Risk: If this requirement is not met, then a user will not be able to send or receive a location request alert.
Priority: 🌍🌍🌍🌍 (4)

~~2.5.2.2.4~~2.5.2.6 *Received Location Alert*

Revision: 1.0 (09.28.2008)
Description: This alert is triggered under the following condition(s): 1) A buddy sends their location to the user; and 2) The Received Location Alert is enabled in the user's settings. Information included in the alert: 1) Alert type; 2) Timestamp; 3) Sending user; 4) Receiving user; and 5) GPS location of the sending user.
Risk: If this requirement is not met, then a user will not be able to receive a location sent by a buddy.
Priority: 🌍🌍🌍🌍 (4)

~~2.5.2.2.5~~2.5.2.7 *Friend Request Alert*

Revision: 1.0 (09.28.2008)

Description: This alert is triggered under the following condition(s): 1) User A sends User B a friend request. Information included in the alert: 1) Alert type; 2) Timestamp; 3) User A; 4) User B. This alert has the following exception(s): If User B is not available at the time of alert creation; the alert is added to a queue until the user becomes available.

Risk: If this requirement is not met, then a user will not be able to request friends.

Priority: 🌍🌍🌍🌍 (4)

2.5.3 Non-functional Requirements

2.5.3.1 *Inter-service Communicability*

Revision: 1.0 (09.28.2008)

Description: The server application needs the ability to communicate with the web service, the database, and client applications.

Risk: If the server is unable to communicate with these entities, then nothing will work.

Priority: 🌍🌍🌍🌍🌍 (5)

2.5.3.2 *Efficient*

Revision: (09.28.2008)

Description: The server should run quickly and efficiently, not hindering user requests due to poor implementation

Risk: If the server does not use resources properly, it will run slowly and cause bottlenecks and instability ~~in~~ throughout the system~~, completely messing everything up~~.

Priority: 🌍🌍 🌍🌍 🌍 (5)

2.5.3.3 *Easily Maintainable*

Revision: (09.28.2008)

Description: The server should be easy to use and maintain. Settings should be self-explanatory and obvious to any administrator to update or change.

Risk: If the server does not allow for changes to be made easily, simple changes can lead to increased downtime, ruining client experience.

Priority: 🌍🌍 🌍 (3)

2.5.3.4 *Easily Scalable*

Revision: (09.28.2008)

Description: The server should allow updates and new features to be easily added to itself without a complete reworking of the source's structure.

Risk: If the server doesn't allow for it to be added onto or expanded, then crucial time will be wasted trying to implement new features in a non-standard way.

Priority: 🌍🌍 🌍🌍 (4) ~~– because it's *that* important~~

2.6 Legal Requirements

2.6.1 Internet Connectivity

While most phones have 802.11 Wireless capabilities, some older phones do not come standard with this feature. As a result, mEYEtrak will communicate with the server through an available data plan, which is a paid feature amongst most carriers. It is highly recommended that a user has an Unlimited

Data Plan to ensure they are not charged extra fees from their service provider. Neither mEYEtrak nor its creators are responsible for the cost incurred.

2.6.2 SMS

At times, mEYEtrak will need to communicate with the server via the phone's Short Message Service. An example of this usage is the verification process. With the varying Text Messaging plans available, neither mEYEtrak nor its creators are responsible for the cost incurred.

2.6.3 Privacy

mEYEtrak and its creators have taken every precaution to ensure that users' privacy is maintained. However, there is always the possibility that mEYEtrak and its features can be abused. In turn, neither mEYEtrak nor its creators are responsible for the potential consequences of such actions.