



## TESTING DOCUMENT

*Breaking down barriers between carriers*

## Table of Contents

<b>1</b>	<b>REFERENCED DOCUMENTS .....</b>	<b>3</b>
<b>2</b>	<b>TEST CASES .....</b>	<b>3</b>
2.1	ACCOUNT INFORMATION .....	3
2.1.1	Login Information.....	3
4.2.1	Personal Information.....	3
2.1.2	Personal Settings.....	3
2.1.3	Privacy Settings .....	4
2.2	CLIENT APPLICATION .....	4
2.2.1	Internet Connectivity.....	4
2.2.2	Global Positioning System (GPS) Capability.....	4
2.2.3	Mobile Environment.....	4
2.2.4	SMS Capability .....	4
2.2.5	Contact List Access .....	4
2.2.6	Current GPS Location .....	5
2.2.7	Current Cell Tower Location .....	5
2.2.8	Google Maps Integration.....	5
2.2.9	Update Location .....	5
2.2.10	Fetch Buddy List .....	5
2.2.11	Buddy Manipulation .....	5
2.2.12	Buddy Location Request .....	6
2.2.13	Range Bar.....	6
2.2.14	Account Information Modification.....	6
2.3	WEB INTERFACE.....	6
2.3.1	Compatible Web Browser.....	6
2.3.2	Internet Connection .....	7
2.3.3	Google Maps.....	7
2.3.4	Fetch Buddy List .....	7

2.3.5	Buddy Manipulation .....	7
2.3.6	Buddy Location Request .....	7
2.3.7	Range Bar .....	8
2.3.8	Account Information Modification.....	8
<b>2.4</b>	<b>SERVER APPLICATION .....</b>	<b>8</b>
2.4.1	Microsoft Windows Server .....	8
2.4.2	Microsoft .NET Framework 3.5 .....	8
2.4.3	Microsoft SQL Server 2005 .....	8
2.4.4	Internet Connection .....	8
2.4.5	Account Updates .....	9
2.4.6	Alerts .....	9
2.4.7	Buddy in Range Alert .....	9
2.4.8	Stolen Phone Trak Alert.....	9
2.4.9	Location Request Alert .....	10
2.4.10	Received Location Alert.....	10
2.4.11	Friend Request Alert.....	10

## 1 Referenced Documents

The mEYetrak Requirements Document was used to create this test plan. Every test case is mapped to one or more requirements.

## 2 Test Cases

### 2.1 Account Information

#### 2.1.1 Login Information

The test objective is to verify that the account key is the user's email address and the password, created by the user, is at least 8-32 characters and contains at least one lower case letter, upper case letter, and number.

The test class is "Login Processing". The Qualification method is "Demonstration" realized by entering valid and invalid login information and verifying that the system responds correctly. The data recorded for this test is the examiner's signature that the system responds correctly.

#### 1.1.1 Personal Information

The test objective is to verify that the user's full name (comprised of their first name, last name, middle initial, and suffix), mobile phone number (used to run the client application), and most up-to-date GPS coordinates (including time of last location synchronization) are including as Personal Information.

The test class is "User Information". The qualification method is "Inspection" realized by examining user personal information and verifying that it contains the full name, mobile phone number, and latest GPS coordinates. The data recorded for this test is the examiner's signature for correct information.

#### 2.1.2 Personal Settings

The test objective is to verify that the user can specify a range radius, in miles, between themselves and a specific buddy or group, and set enable or disable the "Buddy in Range" and "Location Request" alerts. The default range radius is one (1) mile.

The test class is "User Settings". The qualification method is "Demonstration" realized by using an emulator, generating various range radii, and verifying that the system responds with the correct alerts, if enabled. The data recorded for this test is the examiner's signature for correct responses.

### 2.1.3 Privacy Settings

The test objective is to verify that the privacy options (“public”, “semi-private”, and “private”) are functioning correctly, and that each group is assigned a privacy setting that applies to all buddies in the group.

The test class is “Privacy Settings”. The qualification method is “Demonstration” realized by using an emulator, generating various group privacy options, and verifying that the system responds correctly. The data recorded for this test is the examiner’s signature for correct responses.

## 2.2 Client Application

### 2.2.1 Internet Connectivity

The test objective is to verify that the client phone has an internet connection.

The test class is “Client Environment”. The qualification method is “Inspection” realized by examining the client phone and verifying that it has an internet connection. The data recorded for this test is the examiner’s signature for available internet connection.

### 2.2.2 Global Positioning System (GPS) Capability

The test objective is to verify that the client phone has a Global Positioning System (GPS).

The test class is “Client Environment”. The qualification method is “Inspection” realized by examining the client phone and verifying that it has a Global Positioning System. The data recorded for this test is the examiner’s signature for GPS capability.

### 2.2.3 Mobile Environment

The test objective is to verify that the client phone runs Windows Mobile 6.

The test class is “Client Environment”. The qualification method is “Inspection” realized by examining the client phone and verifying that it runs Windows Mobile 6. The data recorded for this test is the examiner’s signature for correct mobile environment.

### 2.2.4 SMS Capability

The test objective is to verify that the client phone has SMS capabilities.

The test class is “Client Environment”. The qualification method is “Inspection” realized by examining the client phone and verifying that it has SMS. The data recorded for this test is the examiner’s signature for available SMS capabilities.

### 2.2.5 Contact List Access

The test objective is to verify that the client can access the contact list on the host.

The test class is “Client Functionality”. The qualification method is “Demonstration” realized by utilizing the client to access the contact list. The data recorded for this test is the examiner’s signature for correct access.

#### **2.2.6 Current GPS Location**

The test objective is to verify that the client can access the host’s GPS chip.

The test class is “Client Functionality”. The qualification method is “Demonstration” realized by utilizing the client to access the GPS chip. The data recorded for this test is the examiner’s signature for correct access.

#### **2.2.7 Current Cell Tower Location**

The test objective is to verify that the client can access the cell tower triangulation function of the host.

The test class is “Client Functionality”. The qualification method is “Demonstration” realized by utilizing the client to access the cell tower triangulation function. The data recorded for this test is the examiner’s signature for correct access.

#### **2.2.8 Google Maps Integration**

The test objective is to verify that the client can integrate with Google Maps.

The test class is “Client Functionality”. The qualification method is “Demonstration” realized by utilizing the client to access and integrate with Google Maps. The data recorded for this test is the examiner’s signature for correct access.

#### **2.2.9 Update Location**

The test objective is to verify that the client can update its location on the server.

The test class is “Client Functionality”. The qualification method is “Demonstration” realized by utilizing the client update its location on the server. The data recorded for this test is the examiner’s signature for a correct update.

#### **2.2.10 Fetch Buddy List**

The test objective is to verify that the client can retrieve the buddy list and corresponding locations from the server.

The test class is “Client Functionality”. The qualification method is “Demonstration” realized by utilizing the client to retrieve the buddy list and locations from the server. The data recorded for this test is the examiner’s signature for correct retrieval.

#### **2.2.11 Buddy Manipulation**

The test objective is to verify that the client can add a buddy (via phone number or email address), remove a buddy, and confirm a buddy request.

The test class is “Client Functionality”. The qualification method is “Demonstration” realized by a three-part test. The first part is realized by generating an “Add Buddy” using a phone number, confirming addition, and generating a “Remove Buddy”, and confirming removal. The second part is the same process; however the buddy is added via an email address. The third part is realized by generating a “Confirm Buddy” request and verifying that the client responds correctly based on a confirmation or rejection. The data recorded for this test is the examiner’s signature for correct buddy addition and removal, and for correct response on a “Confirm Buddy” request.

#### **2.2.12 Buddy Location Request**

The test objective is to verify that the client can request the location of a buddy.

The test class is “Client Functionality”. The qualification method is “Demonstration” realized by utilizing the client to request a buddy’s location and verifying that the buddy client responds correctly. The data recorded for this test is the examiner’s signature for correct response.

#### **2.2.13 Range Bar**

The test objective is to verify that the client displays buddies within the range radius in a graphical, linear form, ordered by distance.

The test class is “Client Functionality”. The qualification method is “Demonstration” realized by generating buddies within range radius of the client and verifying that the client displays correct information including, but not limited to, name, number, and picture. The data recorded for this test is the examiner’s signature for correct display.

#### **2.2.14 Account Information Modification**

The test objective is to verify that the client can modify all information settings in the Account Information Requirements section with the exception of email address and location.

The test class is “Client Functionality”. The qualification method is “Demonstration” realized by utilizing the client to modify information settings and verifying that only those (allowable) settings were changed. The data recorded for this test is the examiner’s signature for correct modifications.

### **2.3 Web Interface**

#### **2.3.1 Compatible Web Browser**

The test objective is to verify that the web browser used by the client is Firefox 3 or Internet Explorer 7.

The test class is “Web Interface Environment”. The qualification method is “Inspection” realized by verifying that the web browser used with the client is Firefox 3 or Internet Explorer 7. The data recorded for this test is the examiner’s signature for acceptable browsers.

### 2.3.2 Internet Connection

The test objective is to verify that the user has an internet connection.

The test class is “Web Interface Environment”. The qualification method is “Inspection” realized by verifying an internet connection. The data recorded for this test is the examiner’s signature for a connection.

### 2.3.3 Google Maps

The test objective is to verify that the website integrates with Google Maps through the available API.

The test class is “Web Interface Functionality”. The qualification method is “Demonstration” realized by verifying that the website integrates with Google Maps correctly. The data recorded for this test is the examiner’s signature for correct integration.

### 2.3.4 Fetch Buddy List

The test objective is to verify that the web interface can retrieve the buddy list and corresponding locations from the server.

The test class is “Client Functionality”. The qualification method is “Demonstration” realized by utilizing the web interface to retrieve the buddy list and locations from the server. The data recorded for this test is the examiner’s signature for correct retrieval.

### 2.3.5 Buddy Manipulation

The test objective is to verify that the web interface supports add a buddy (via phone number or email address), remove a buddy, and confirm a buddy request.

The test class is “Web Interface Functionality”. The qualification method is “Demonstration” realized by a three-part test. The first part is realized by generating an “Add Buddy” using a phone number, confirming addition, and generating a “Remove Buddy”, and confirming removal. The second part is the same process; however the buddy is added via an email address. The third part is realized by generating a “Confirm Buddy” request and verifying that the web interface responds correctly based on a confirmation or rejection. The data recorded for this test is the examiner’s signature for correct buddy addition and removal, and for correct response on a “Confirm Buddy” request.

### 2.3.6 Buddy Location Request

The test objective is to verify that the website supports request buddy location requests.

The test class is “Web Interface Functionality”. The qualification method is “Demonstration” realized by utilizing the website to request a buddy’s location and verifying that the buddy client responds correctly. The data recorded for this test is the examiner’s signature for correct response.



### 2.3.7 Range Bar

The test objective is to verify that the website displays buddies within the range radius in a graphical, linear form, ordered by distance.

The test class is “Web Interface Functionality”. The qualification method is “Demonstration” realized by generating buddies within range radius of the client and verifying that the website displays correct information including, but not limited to, name, number, and picture. The data recorded for this test is the examiner’s signature for correct display.

### 2.3.8 Account Information Modification

The test objective is to verify that the website can modify all information settings in the Account Information Requirements section with the exception of email address and location.

The test class is “Web Interface Functionality”. The qualification method is “Demonstration” realized by utilizing the website to modify information settings and verifying that only those (allowable) settings were changed. The data recorded for this test is the examiner’s signature for correct modifications.

## 2.4 Server Application

### 2.4.1 Microsoft Windows Server

The test objective is to verify that the server has Windows Server (preferably 2008) installed on it.

The test class is “Server Environment”. The qualification method is “Inspection” realized by verifying the server has Windows installed on it. The data recorded for this test is the examiner’s signature for Windows Server.

### 2.4.2 Microsoft .NET Framework 3.5

The test objective is to verify that the server has Microsoft .NET Framework 3.5 installed on it.

The test class is “Server Environment”. The qualification method is “Inspection” realized by verifying the server has Microsoft .NET Framework 3.5 installed on it. The data recorded for this test is the examiner’s signature for the .NET Framework.

### 2.4.3 Microsoft SQL Server 2005

The test objective is to verify that the server Microsoft SQL Server 2005 installed to handle the database needs.

The test class is “Server Environment”. The qualification method is “Inspection” realized by verifying the server has Microsoft SQL Server 2005 installed on it. The data recorded for this test is the examiner’s signature for the SQL Server.

### 2.4.4 Internet Connection

The test objective is to verify that the server has internet connection.

The test class is “Server Environment”. The qualification method is “Inspection” realized by verifying the server is successfully connected to the internet. The data recorded for this test is the examiner’s signature for a connection.

#### 2.4.5 Account Updates

The test objective is to verify that the server handles account updates (including GPS location) generated by the client application and the web interface.

The test class is “Server Functionality”. The qualification method is “Demonstration” realized by generating account updates via the client application and the web interface and verifying that the server processes the information correctly. The data recorded for this test is the examiner’s signature for correct processing.

#### 2.4.6 Alerts

The test objective is to verify that the server handles alerts based on set conditions and sends it to any and all open sessions an account has (client application or web interface).

The test class is “Alert Processing”. The qualification method is “Demonstration” realized by generating alerts via an emulator and verifying that the server sends the alerts to the account’s open sessions. The data recorded for this test is the examiner’s signature for correct alert processing.

#### 2.4.7 Buddy in Range Alert

The test objective is to verify that the server initiates a “Buddy in Range” alert (with the correct information included) when the alert is enabled on the user’s settings and a buddy’s GPS location is within the range radius of the user’s GPS location.

The test class is “Alert Processing”. The qualification method is “Demonstration” realized by generating a “Buddy in Range” alert via an emulator and verifying that the server sends the alerts to the account’s open sessions with the following information: 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. The data recorded for this test is the examiner’s signature for correct alert processing.

#### 2.4.8 Stolen Phone Trak Alert

The test objective is to verify that the server initiates a “Stolen Phone Trak” alert (with the correct information included) and sends it to the client application, if one is open, or added to a queue (where it stays up to a week) until a client session opens.

The test class is “Alert Processing”. The qualification method is “Demonstration” realized by generating a “Stolen Phone Trak” alert via an emulator and verifying that the server sends the alerts to the account’s client session if one is open, or to a queue (for up to a week) if one is not, with the following information: 1) alert type; 2) timestamp; 3) user. The data recorded for this test is the examiner’s signature for correct alert processing.

#### 2.4.9 Location Request Alert

The test objective is to verify that the server initiates a “Location Request” alert (with the correct information included) when a buddy requests the user’s location and when the Location Request Alert is enabled in the user’s settings.

The test class is “Alert Processing”. The qualification method is “Demonstration” realized by generating a “Location Request” alert via an emulator and verifying that the server sends the alerts to the account’s open sessions with the following information: 1) alert type; 2) timestamp; 3) requesting user; 4) target user. The data recorded for this test is the examiner’s signature for correct alert processing.

#### 2.4.10 Received Location Alert

The test objective is to verify that the server initiates a “Received Location” alert (with the correct information included) when a buddy sends their location to the user and when the Received Location Alert is enabled in the user’s settings.

The test class is “Alert Processing”. The qualification method is “Demonstration” realized by generating a “Received Location” alert via an emulator and verifying that the server sends the alerts to the account’s open sessions with the following information: 1) alert type; 2) timestamp; 3) sending user; 4) receiving user; 5) GPS location of the sending user. The data recorded for this test is the examiner’s signature for correct alert processing.

#### 2.4.11 Friend Request Alert

The test objective is to verify that the server initiates a “Friend Request” alert (with the correct information included) when user A sends user B a friend request. If user B is not available at the time of alert creation, the alert is added to a queue until the user becomes available.

The test class is “Alert Processing”. The qualification method is “Demonstration” realized by generating a “Friend Request” alert via an emulator and verifying that the server sends the alerts to the account’s open sessions with the following information: 1) alert type; 2) timestamp; 3) user A; 4) user B. The data recorded for this test is the examiner’s signature for correct alert processing.