

EXHIBIT C

Mark Zuckerberg
2734 Harvard Yard Mail Center
Cambridge, MA 02138
914.646.8593

StreetFax Back-End Technical Specification

Non-technical Explanation:

To make the specification more readable, I will give an overview of the functionality of the system described by the set of scripts and applications below.

The first section deals with logon and security. The first script makes sure that the interface through which users log into the system is completely secure, using the latest methods of commercial cryptography. It will verify if the user has the privileges to enter the system, and if they do, it admits them. The second script protects the system from being entered through a page other than the logon page. If the only security was at the logon screen, then a person could just go to a different page and bypass the logon completely. This script will ensure that the user is logged in before it grants access to any page on the site. The third script in this section will allow the site administrators to create and edit usernames and account information. This is critical for customers who do not create their accounts through the site.

The second section deals with e-commerce. Much of the discussion here is about the different options of registration with SSL and VeriSign. The scripts we develop will use these technologies to perform secure e-commerce transactions. It will allow users to register on a subscription and individual-use basis. All actions performed by users in this section of the site will be logged in a database and can be used to generate reports later on.

The third section deals with searching the database of images. The user will enter two streets to search for their intersection, and then a script we write will search the database. If multiple intersections are found, the user will be shown a list with the towns and states to choose from. If, after reviewing the choices, there is still no an accurate match, the user will be routed to another page which will ask them to specify a state and city for a more detailed search. From their selection, the user is taken to another page with the list of intersections matching their search in the specified area. The user can choose an intersection to search for images. If only a single intersection is returned from the original search, then the images from that intersection are automatically brought up. The images will be displayed in alphabetical order with any signs at the top of the page above the photographs of the intersections. The script that retrieves the images from the database will also construct the necessary sentence describing the image from information in the database. If at any point no intersection is found, the user will be taken to a site that asks them if they want to pay to have those images acquired for them within 24 hours, this screen will appear as a half page with the nearest possible matches above it. They can search these pictures by clicking on any one of the intersections to

view it, and then by navigating back and selecting the next intersection. There will also be a script to get the number of images matching an intersection.

The fourth section deals with saving and retrieving adjuster comments and layout settings from the database. This will allow adjusters to write comments and retrieve them later on. They will also be able to specify a layout for the screen, but we have not designed the possible layouts yet so no further explanation can be given at this point. A sample outline has been received consisting of 4 different layout types.

The fifth, sixth, and seventh sections deal with the very back-end implementation and architecture of the server and database. The database will be robust enough to be able to efficiently search the collection of images and be able to store them in a number of conventions, although the convention of "direction+street1+street2" will be accepted for now. The server will be equipped with the latest and most flexible web server, database server, and a solid hosting package. It will also have a built in interface for retrieving intersections from a given area and editing their database entries.

The eighth section deals with the client/photographer interface. Once a client requests an image, the photographer whose territory the intersection is in is notified that they need to take the picture. The dispatcher is also notified that the photograph needs to be taken by a new entry on the open requests page. This open requests page will give information on when the request was made, who made the request, and what intersection was requested. After 40 minutes, if the photographer assigned to that territory has not responded to the notification and confirmed its receipt, the specific request on the open requests page will turn a different color. When the photographer does respond to this initial request, the open requests page will display this and an email will be automatically generated and sent back to the adjuster informing them that their request has been received and accepted by the nearest photographer, and that their photos will be provided guaranteed. At selected intervals, if the photographer has not yet responded or uploaded the images, the dispatcher is again notified that the photographs still need to be taken. The system will continue to email the photographer and update the open requests page with the time remaining and status on the situation until the images are uploaded to the system. On the street, once the images have been taken, the photographer can upload them to the server, which will require the photographer to sign in to their homepage prior to upload. From here, the photographer must enter the request's claim number, which will be automatically generated with each new request. After the appropriate claim number has been entered, the server uploads the pictures to the site and updates the open requests page by graying the area. At this point, a link appears and the dispatcher has five minutes to approve or reject the photographs before the server automatically notifies the adjuster through email that their pictures are on site. There will be a link in the email that will allow the adjuster to link directly to the page which has his or her images. If the dispatcher rejects the pictures, then no notice is given to the adjuster and the process is taken over manually.

The ninth and tenth sections deal with allotting time for administrative discussions and algorithm planning.

Database Structure and Scripting:

1. Logon Scripts

- a. A script to validate logon information by referencing the user information table in the database. Validation includes taking the username and password and making sure they match and that the account is still valid. This script will also acquire the privileges of the user to ensure that only searches they are entitled to will be performed. It will use special security measures and encryption to maintain the integrity of users' passwords. This script will also log all user logons in a separate table in the database. Estimated Development Time: 1 day.
- b. A script to check for the logon key on all other privileged parts of the site. This script will protect against people bypassing the logon screen and accessing user-only parts of the site. EDT: ½ day.
- c. A script to add users and edit users. EDT: ½ day.

2. Financial Scripts

- a. A script to interface with the credit card validation component. This script will use security measures like secure socket layer (SSL) to ensure the integrity of the transactions. Two different scripts will need to be developed to deal with subscription and individual use transactions. All transactions will be logged in a separate table in the database. *Note:* The credit card validation component and SSL certificate will carry additional costs. There are several options for e-commerce, and these can be discussed later. EDT: 4 days.

3. Search Scripts

- a. A script to interface with the database of images, which is given two street names. If no such intersection exists, then the names of intersections which are similar to the one entered are returned and the search can be repeated. If multiple intersections match the street names entered, then a new page comes up which asks the user to specify a state or city. From the selection, if there are still multiple matches, another page comes up that lets the user choose which intersection they want to retrieve images of. All searches will be logged, along with which user performed them, in a separate table in the database. EDT: 3 days.
- b. A script to retrieve images from the database, given a unique intersection in some state. The script will display all images found in the database along with a sentence describing the image, its orientation, and the street it was taken from. If no images are found, it will ask the user if they would like those images to be acquired. If so, a notice will be sent to an email address or list of email addresses. Either way, the fact that no results came up will be recorded in a separate table in the database along with the user's response to acquiring the images. EDT: 3 days.
- c. A subscript to reference the main script and return how many intersections and/or images match a given set of intersections. EDT: ¼ day.

- d. A subscript to include speed limits in the capacity of the search engine. This script will return images of the speed limits in the queried regions using the keywords described. A full explanation of all keywords will be needed to write this script. EDT: 2 days.
4. Adjuster Preference Scripts
- a. A script to store adjusters' comments about images from their queries in the database. This script will take the comments that adjusters enter about the images and it will save the comments in the database for later retrieval when the same images are viewed. EDT: ½ day.
 - b. A script to query adjusters' comments about images from the database. This script will return and display the current adjuster's saved comments about the currently viewed images in the allotted area for marking notes. EDT: ½ day.
 - c. A script to store adjusters' viewing preferences about the layout of the screen. Several different layout options will be available (they are not specified here but they will be in the final specification), and this script will save each adjuster's settings in the database. EDT: ½ day.
 - d. A script to query adjusters' viewing preferences from the database and incorporate these settings into the display. The script will apply the given display settings into the screen format. As mentioned in 4.c, the different settings available to the adjusters will be specified in the final version of the spec. EDT: 1½ days.
 - e. A script that allows Back End Administrators to view & edit all intersections within a given city through a drop down menu bar.
5. Database Architecture
- a. An application to build the appropriate table structure from the image files in the system. This program will make an entry in the main table in the database for each image file in the database. This is the conversion necessary to get from the raw image collection to the organized database we will use in the site. The program will be robust enough to handle images within different directory structures as long as the same naming convention is used throughout the file system. This will allow images of a given intersection to be stored in a separate folder, or in a conglomerate folder with images from other intersections, as long as each image follows the same naming convention. For this version of the specification, the file naming convention in use unilaterally will be "directional+street1+street2". EDT: 3½ days.
 - b. Configuration of an efficient table structure in the database. This construction will provide the framework for the database so that it can be populated with all of the necessary data fields. EDT: 2 days.

Server Specification and Applications:

- 1. Package Research
 - a. As we have not yet decided how to host the final application, some research must be done to figure out the best option in this area. It seems doubtful that we will be able to find a non-specific contract agreement with some well-known hosting company that will meet our specifications for access requirements and applications that need to be run. Therefore, it might seem

R
112

best to purchase a server and take care of hosting ourselves. We will look into all of these options. EDT: 1½ days.

2. Server Configuration

- a. Stable Internet configuration. EDT: 1 day.
- b. Apache web server configuration. EDT: ¾ day.
- c. PHP pre-processor configuration. EDT: ¾ day.
- d. MySQL configuration and phpMyAdmin interface installation. EDT: ¾ day.
- e. Photographer access configuration. EDT: ¾ day.

3. Photographer Application

- a. A web interface between the photographers and the dispatcher. A script that will automate sending emails to client, photographer, and dispatcher. The client will be emailed upon receipt of their request and again once the request has been completed to tell them that their images are ready. The correct photographer will be contacted to take the images, and then will be warned on regular intervals if the photographs have not been taken and uploaded to the server. An email will be sent to the photographer to confirm the upload of the images when the job is done. An email will be sent to the dispatcher to notify him of the request, and warning emails will be sent on regular intervals to tell the dispatcher if the request has not yet been filled. A confirmation email will be sent to the dispatcher to confirm that the request has been filled. *Note:* This is a major change from the original specification. The original photographer application was discarded and this one replaced it for the initial development. EDT: 4 days.

Internal and External Administrivia:

1. Communication

- a. Correspondence through emails and phone with you about progress, questions, and other general information exchange. EDT: 1 day.
- b. Correspondence and meetings with my team about progress, development, and other general information exchange. EDT: 1 day.
- c. Quality Assurance of all work done by my team. EDT: 3 days.

Algorithm Design:

1. Planning and Design

- a. Architecting the main algorithms in the search engine, database population, file naming, and general site security. EDT: 2 days.

Enhancements (Not Included):

1. Police Reports and Building Inspection

- a. These will essentially require new sections of the database and new database population algorithms. It seems to make sense to save these aspects of the project until later unless they are absolutely necessary in the first version of StreetFax.

2. **Highlighted Maps**
 - a. This feature will be difficult to implement since it will require someone to go through the database and add information to all of the old entries. At this point, that does not seem like an economical use of resources. We can try to implement this enhancement later on, perhaps using a different algorithm.

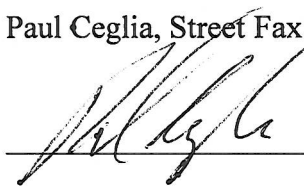
3. **Automated Database Applications**
 - a. It was a little unclear to me what sort of automated database cleanup you wanted, but I definitely see room for redundant entry filtering, data linking, and priority sorting to help increase the efficiency of the system. This can come after the basic development.
 - b. Scripts that query the database to find results from a specific photographer and that email photographers when photographs have not been submitted on time. This functionality also seems less important in getting the system up and running.

4. **Robust Photographer Interface**
 - a. This feature will take the form of a powerful custom server application that photographers can log into to use dynamic functionality within the assignment and request systems. It will also provide the photographers with extra tools for batch uploading and perhaps editing of their images.

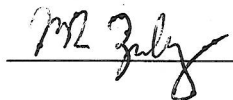
5. **Anti-Hacker System**
 - a. An additional system to ensure the security of the server and maintain the integrity of the information inside. Since some of the data, namely credit card numbers and passwords, is sensitive, this extra functionality is highly recommended in the long term.

This specification will be approved with appropriate signatures below.

Paul Ceglia, Street Fax

 4/28/03

Mark Zuckerberg

 04.28.03