**VRoot Functional Requirements**

**Version 1.0**

**1/24/2012**

Dan Strom

Joe Isaacson

Steve Hunt

Contents

[Glossary 3](#_Toc346747049)

[Introduction 4](#_Toc346747050)

[Functional Requirements 5](#_Toc346747051)

[Profile Tools 5](#_Toc346747052)

[Volunteer Functionality 5](#_Toc346747053)

[Manager/Administrator Functionality 5](#_Toc346747054)

[Organizational Tools 6](#_Toc346747055)

[Documents 6](#_Toc346747056)

[Events 6](#_Toc346747057)

[Schedules 8](#_Toc346747058)

[Data Sharing 8](#_Toc346747059)

[Communication 8](#_Toc346747060)

[Message Boards 8](#_Toc346747061)

[Events 8](#_Toc346747062)

[Direct Messaging 8](#_Toc346747063)

[Non-Functional Requirements 9](#_Toc346747064)

[User Facing 9](#_Toc346747065)

[User Experience 9](#_Toc346747066)

[Performance 9](#_Toc346747067)

[Back-End 9](#_Toc346747068)

[Environmental 9](#_Toc346747069)

[Robustness 9](#_Toc346747070)

[Security 10](#_Toc346747071)

Glossary

* Administrator – An individual involved with TR given the highest permissions in the software. This individual is expected to be at a high level of leadership or a trusted volunteer specifically tasked to administer the software.
* Event – A specific action taken by TR in which volunteers can participate.
* Manager – An individual involved with TR given permissions above volunteers but below administrators. This individual is expected to undertake a leadership role.
* Permissions – The ability granted to each class of users (e.g., volunteers) to view or edit pieces of the software.
* Region – Geographical region assumed to be based on FEMA regions.
* TR – Team Rubicon
* User – Any user of the software
* Volunteer – An individual involved in TR given the lowest level of permissions in the software.

Introduction

This requirements document has been prepared to elucidate the elements of a software tool to aid the national and international efforts of Team Rubicon and its volunteers. The first part of most software engineering projects is to translate client (e.g., TR) needs into a specific set of requirements that a software tool will perform to solve the client’s problem. To be effective these requirements should be reviewed and approved by both all parties before tasks, schedule and technical demonstrations are set to organize the software development. Once a requirements document is finalized, changes to the software’s functionality are expected to be minimal, as major unplanned changes initiated later in the development cycle are typically more difficult to implement and time consuming. Significant upgrades to the functionality specified in this document, possibly due to changes to TR itself, will be recorded (see user feedback), reviewed and implemented in future requirements documents and associated software releases.

Explained within this document are two categories of requirements: functional and non-functional. Functional requirements are specific actions the software will perform. Described in detail below are several sub-categories of functional requirements. General profile tools for volunteers and managers/administrators, organizational tools for document sharing, event administration, and scheduling and clean, concise tools for communication are all essential components for streamlining TR’s workflow. Non-functional requirements describe broader actions taken into account during the implementation of functional requirements. These include user interface considerations, loading performance expectations, and security requirements.

At all stages of development, the software described below will be open sourced and freely available to TR. It is expected that a reasonable level of trust, confidentiality and professional courtesy will be maintained between TR and the authors regarding the content of this document. This document should not be distributed to other parties by TR under any circumstances without prior coordination and understanding.

Functional Requirements

# Profile Tools

## Volunteer Functionality

### Registration

Before using any of the tools provided by the system, a user must register to create an account. The software will provide a form for users to fill out in order to register. Required registration information will include name, contact information, and address. Users will be assigned to a region based on their address. In addition to the form, a web service will be provided to allow submissions from registration forms on other websites. Once registered, the user will have gain access to a small, configurable subset of features until their registration is approved by a manager. If a user has not completely filled out the minimum required set of information their registration will be flagged as invalid or incomplete.

### Waivers

Due to the nature of volunteer work, Team Rubicon may require volunteers to sign waivers, codes of conduct, or other documents. The software will provide an interface to handle these electronically. Users will be able to view these documents, print them out and upload signed copies.

### Volunteer Profile

Each user will have an editable profile page viewable by Team Rubicon. Users will be able to list any special skills, training, or experience that might be useful to a volunteer effort. They will also be able to upload files such as DD214 or training certification images, viewable by TR managers or administrators.

## Manager/Administrator Functionality

### Registration Approval

Managers will be able to approve newly registered users. Unapproved volunteers will only have access to the most basic functionality of the software.

### Volunteer Search

The software will provide a method to search for volunteers based on location, skills, certifications, or any other pertinent information in the system. Search results will include each volunteer’s name, contact information, and a link to their profile.

### Volunteer Evaluation

Managers will be able to save evaluations of individual volunteers. These evaluations will be invisible to volunteers and only viewable by TR managers. Evaluations allow organizers to store information about whether or not to work with a specific volunteer again and for what types of work the volunteer is best suited.

# Organizational Tools

## Documents

The software will provide a repository for any documents Team Rubicon might want to distribute to its volunteers. Team Rubicon will be able to upload documents, define who they can be viewed by, and depending on the type, make changes within the software. The software will also allow the creation of a directory structure to organize these files. Each document will have assigned parameters that determine its visibility to users:

### Regional Visibility

Each document will be assigned to a region. Unless given special permission, volunteers will only be able to access documents from the same region. Managers may view documents from all regions. In addition, there will be a “National” region for non-region specific documents. National documents will be accessible to all users, and likewise, national users (i.e., managers and administrators) will have access to documents from every region.

### Permission Level Visibility

Each document will have a minimum permission level to view or edit. This can be used to keep certain documents from going out to every single volunteer, and ensure that only specific people are able to edit documents. Documents can be set to be either viewable or editable by volunteers, managers, and/or administrators. Since TR is likely to have a hierarchy more fine-grained than 3 basic categories, the software will support the creation of permission levels within the categories. Document viewing and editing permission within the groups can also require a minimum permission level.

## Events

In order to keep track of volunteering efforts, Team Rubicon will be able to create events in the system. These will help to organize and track volunteering. Events will be created with a description, time, and location. Once created, events will be searchable by volunteers.

### Data Feeds

The software will allow Team Rubicon to add connections to external data feeds. These configurable feeds will be able to draw data from any public online data source with a standard feed interface. These feeds can be added to events or event tags. Data feeds on these pages can be configured with search terms to help provide more relevant information.

### Event Tags

Team Rubicon will be able to create event tags to help with organization. An event tag is a category describing an event (e.g., Katrina). These tags can describe the type of work being done, issues the event is working to fix, or the larger occurrence that the event is responding to. Any number of applicable event tags can be added to an event. Users will be able to search and browse events by tags.

### Application

Volunteers will be able to apply to participate in events. Team Rubicon will then be able to view the profiles of all applying volunteers to check skills, experience, and evaluations of the volunteers. TR managers can subsequently approve or reject the request to participate.

Team Rubicon may also run user searches to find qualified volunteers and invite candidates to participate. In this case, it is up to the volunteer to accept or reject the invitation.

Once approved on both ends, related events will automatically be added to a user’s schedule.

### Communication

Team Rubicon will have an easy way to communicate (WARNO, SITREPS etc.) with all volunteers approved for a particular event.

### Organization

Event managers will have a tool to help decide on how to group users for carpooling. The software will organize volunteers by geographic location and/or route and provide a grouping suggestion (personnel list with contact information).

### Hour Tracking

Once underway, the system will allow volunteers to “punch in” and “punch out” from their events. These start and end times can be used to track volunteering hours for each volunteer and the total for the event. Team Rubicon can configure whether or not the times need verification.

These hours can also be aggregated over event tags to easily calculate the total hours worked.

### Expenses

After returning from an event, volunteers will be able to use the software to report expenses. They will be allowed to upload images of receipts or other documents to verify their claims.

Managers will be able to approve these reports in-system and view or download expense reports for individual volunteers or whole events.

## Schedules

The software will provide built-in schedules to keep track of events. Managers will also be provided an interface to create meetings or training sessions and invite registered volunteers to these events.

As many people use external calendar tools to manage their schedules, there will be an interface to export schedule data to universal calendar formats.

## Data Sharing

The software will support the sharing of specific and aggregate data with external organizations that has been previously stored in its database.

# Communication

## Message Boards

The software will provide the ability to create message boards for communication. Managers will be able to create boards, add users and moderate content. This will allow groups of users to make non-urgent communications inside the system instead of using long chains of emails. Users will be able to decide how often message boards send notification emails. Potential options include: notify on every message, manager messages only, or periodic digest emails. Users with permission will be able to force important posts to override these settings and notify users on the board of the post by email. The software will support the transmission/receipt of text messages from event specified distribution lists.

## Events

Each event will automatically have a message board containing each user involved with the event. Each event will automatically have an event email and text message distribution list.

## Direct Messaging

Users will be able to directly send messages to one or more other users within the system.

### User Feedback

There will be a user feedback tab built into every functional window that will allow the transmission of text and images to collect software product feedback.

Non-Functional Requirements

# User Facing

## User Experience

As a web-based application, the software will be run in a web browser. The interface will have a strong focus on usability and intuitive structure. This will be done by using a familiar layout and providing mouse-over help text wherever possible. There will be online user guide information available on each user page.

## Performance

Most pertinent functions will be expected to run quickly, without waiting more than the standard time to load and render a page in a web browser. Functions which are abnormally intensive must indicate to the user that the delay is due to the computation time of the process rather than an error or slow service.

The software must be able to concurrently serve a large number of users simultaneously. While normal usage may be relatively easy on resources, any site can as part of its normal usage cycle see large spikes in activity. The software should be elastic enough to continue to run in these peak usage scenarios without too noticeable a performance hit.

# Back-End

## Environmental

The application will be developed using a web scripting language on a server connecting to a relational database. Preference parameter files will maintain a standard setup at initiation.

## Robustness

Many functions of the software will require users to fill out forms. Entries into these forms will be verified on submissions. Fields with easily recognizable formats (email addresses, phone numbers, etc.) will have to follow expected patterns, and required fields must be filled out. Submissions with fields that fail this basic testing will be rejected and the failing fields will be highlighted for the users to fix and resubmit.

Due to the fact that much of this data will likely be inserted into a database and later retrieved and displayed on another webpage in the system, any values in those fields that may cause unexpected behavior in the database or page rendering will be properly sanitized.

## Security

As a web application, the software will be exposed to the larger internet population. This makes it vulnerable to various forms of attack. The main focus of security will be the protection of personally identifiable information contained in the database. Any communication of personally identifiable information will be done over HTTPS to keep all data encrypted while in transit. The software will be robust enough to prevent malicious users from using the software to interact with the database in any way other than intended. The database should also prevent direct connection from outside the software. The software will not store any information that can be highly damaging, such as for donations, third party applications (e.g. PayPal, Google Checkout, etc.) will be leveraged whenever possible.