Voluntunities
CS 411W Gold Software Requirements
Lab 2 Section 3 v3
11/06/17

Table of Contents

3 Specific Requirements 2
3.1 Functional Requirements 2
  3.1.1 User Profiles 2
  3.1.2 Opportunities 3
  3.1.3 Virtual Bulletin Board 4
  3.1.4 Servers and Database 5
3.2 Performance Requirements 6
  3.2.1 Google Maps Api 6
  3.2.2 Servers 6
3 Specific Requirements
3.1 Functional Requirements
3.1.1 User Profiles (*Breon Day*)

3.1.1.1 Voluntunities shall allow individuals and organizations to create new volunteer or opportunity account profiles, using a unique username and email address using the following relevant information:

- Unique username
- Email address
- Hashed and salted password
- Location
- Time of creation
- Time last updated
- Stored photos
- Description
- User Preferences
- Tagged skills

3.1.1.2 Voluntunities shall allow individuals and organizations to log in to their created profiles by inputting the account’s unique username, or email address, and the paired password. The following prerequisites will be required:

- The user has an account created from 3.1.1.1.
- The user correctly spells the username or email address created from 3.1.1.1.
- The user correctly spells the paired password created from 3.1.1.1.

3.1.1.3 Voluntunities shall not allow individuals or organizations to log in to created profiles when input with an incorrect username, or email address, or the paired password created from 3.1.1.1.

3.1.1.4 Voluntunities shall provide access to view user accounts.

3.1.1.5 Voluntunities shall provide the ability for the account owner to restrict account viewing to the owner only.

3.1.1.6 Voluntunities shall provide access to edit and update the account profile of the registered account owner. The following functionality will be provided/required:

A. User chooses to edit account profile.
B. User is shown editable version of account.
C. User removes moves or adds components to the profile.
D. User saves or cancels out of finished edits.
E. User is sent back to un-editable version of account profile.

3.1.1.7 Voluntunities shall provide the ability for users to register profiles under organizations.
3.1.2 Opportunities (Joshua Stallard)

3.1.2.1 Voluntunities shall allow users to post a volunteer opportunity composed of:
- Title
- Description
- Time/Date
- Location
- Items or skills required
- Lifetime of opportunity

3.1.2.2 Voluntunities shall allow users to post a charity opportunity composed of:
- Title
- Description
- Progress meter
- Goal
- Lifetime
- Time of creation

3.1.2.3 Voluntunities shall allow users to delete an existing opportunity or charity.

3.1.2.4 Voluntunities shall allow users to restructure/transfer management of an existing charity or opportunity.

3.1.2.5 Voluntunities shall allow users to store information about completed or deleted opportunities to the user profiles mentioned from 3.1.1.1.
3.1.3 Virtual Bulletin Board (Matthew Streett)

3.1.3.1 Voluntunities shall allow users to select an opportunity.

3.1.3.2 Voluntunities shall allow users to browse a list of available opportunities.

3.1.3.3 Voluntunities shall allow users to sort opportunities.

   3.1.3.3.1 Voluntunities shall allow users to sort opportunities by time and date by selecting a time frame for the opportunity.

   3.1.3.3.2 Voluntunities shall allow users to sort opportunities by desired distance in miles.

   3.1.3.3.3 Voluntunities shall allow users to sort opportunities from a list of tags.

3.1.3.4 Voluntunities shall allow users to search the available opportunities by keywords.

3.1.3.5 Voluntunities shall allow users to search the available opportunities by tags.

3.1.3.6 Voluntunities shall allow users to see nearby opportunities. The following functionality shall be provided.

   A. Map display through the use of Google Maps API.
   B. User profile information from 3.1.1.1.
3.1.4 Servers and Database (Jamie gastrich and Ali Shaikh)

3.1.4.1 Server

3.1.4.1.1 Voluntunities shall use MS ASP.NET Core MVC(Web API) to provide RESTful API client for all applications. The following requests/calls will be supported.

3.1.4.1.1.1 Authorize
- Params: Username,Password
- Return: Timed Authorization Token

3.1.4.1.1.2 User
- Params: Username
- Return: JSON with public User Info

3.1.4.1.1.3 Organization
- Params: Organization Name
- Return: JSON with public Organization Info

3.1.4.1.1.4 Opportunity
- Params: Opportunity ID
- Return: JSON with public Opportunity Info

3.1.4.1.1.5 Edit - Provides Edit capability for each of the above in the following format, saves
- {Type}/{ID}
- Params: Datatype, ID, JSON with updated info
- Return: JSON with new saved info

3.1.4.1.2 Voluntunities shall communicate with the Database to access relevant information from 3.1.4.1.1 using Entity Framework.

3.1.4.2 Database

3.1.4.2.1 Voluntunities shall use MS SQL database to store all the necessary information.

3.1.4.2.1.1 User Table
The User table shall hold the information of the application’s registered users. Each entry shall represent a single individual user.

3.1.4.2.1.2 Opportunities Table
The Opportunities table shall hold the information of the application’s opportunities, past present and future. Each entry shall represent a single opportunity.

3.1.4.2.1.3 Organizations Table
The Organizations table shall hold the information of the application’s Organizations. Each entry shall represent a single Organization.

3.1.4.2.1.4 Location Table
The Location table shall hold the location information about the users, the opportunities, and the organizations. Each entry shall represent a single location.
3.2 Performance Requirements

3.2.1 Google Maps API

The Google Maps API shall meet the following performance requirements:
1. Capable of returning opportunity locations within 1 to 50 miles.
2. Capable of receiving and handling search requests from all active users numbering less than 10,000, searching for an opportunity at any given time.

3.2.2 Server

The server and database must be able to meet the following requirements:
1. The server must handle simultaneous requests from at least 100 separate users, including
   a. Authorization Tokens
   b. View generation
   c. Accessing and updating database
2. The database must not lock all records during
   a. Database retrieval
   b. Database updates
3. Have 99.99% uptime.