|  |
| --- |
| Project Group 7 |
| VolunteerNow |
| CEN4010 Principles of Software Engineering, Summer Semester 2023 https://github.com/JonathanG77/CEN-4010-Project Version: 1.0|6/20/2023 |

|  |
| --- |
| Jonathan Garcia jggarcia2020@fau.edu Ernesto Benavides ebenavides2016@fau.edu Jacob Campbell campbellJ2019@fau.edu Tivvon Cruickshank tcruickshank2018@fau.edu Kareen Godder kgodder2020@fau.edu  6-20-2023 |

Table of Contents:

[Executive Summary 2](#_Toc138183092)

[Competitive Analysis 3](#_Toc138183093)

[Data Definition 6](#_Toc138183094)

[Use Cases 7](#_Toc138183095)

[List of Functional Requirements 16](#_Toc138183096)

[List of Non-Functional Requirements 19](#_Toc138183097)

[High-level System Architecture 21](#_Toc138183098)

[Team Roles 23](#_Toc138183099)

[Checklist 24](#_Toc138183100)

[Revision History 25](#_Toc138183101)

# Executive Summary

At VolunteerNow, we are designing a powerful system that functions as an all-in-one service that will benefit both potential future volunteers and the organizations that depend on them. Volunteers will have access to our powerful search that will allow them to connect with thousands of organizations across the country and be able to filter by their skills and personal preferences which maximize the possibility of finding the right organization to make a difference. Once a suitable opportunity is found, volunteers will be able to become members and apply right from the opportunity page. Even if a volunteer can’t find an opportunity to their preferences, they have the unique ability to make a post with their skills and availability included for organizations to see.

Organizations will have access to many features that will make the normally tough task of finding suitable volunteers a breeze. Organizations will be able to browse the collection of posts made by volunteers to be able to find suitable candidates and send messages to them directly through our system. There will also be the feature to send announcements and notifications through our system to all volunteers associated with that specific organization. Members representing the organization will be able to track volunteer hours, assign them to events, and more directly through our system. This allows important volunteer organizations to spend less time managing and more time doing what matters, helping people. All sorts of organizations will benefit from this, ranging from global non-profits to local community groups and everyone in between. This is only possible at VolunteerNow, where we turn talk into action.

# Competitive Analysis

Research into some of the various competitor volunteer management sites out there has identified key features that our planned site features are either on the same level as or improve upon the implementation of our competitors. Since the site is planned to be both a way for volunteers to search for volunteer jobs to apply for and a place for organizations to post listings to acquire volunteers, we did not take into consideration competitors that only offer volunteer management systems. With that in mind, the following table will compare five different main features between the [site] website and our competitors. Each category will be rated on a scale from 1 to 10 with a one indicating that the feature is poorly implemented on the site and a ten indicating that the feature has an excellent implementation and presence on the site.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **VolunteerNow** | **VolunteerMatch** | **MyVolunteerPage** | **JustServe** | **International Volunteer HQ** |
| **Design** | 7 | 6 | 4 | 5 | 8 |
| **Navigation** | 8 | 7 | 3 | 7 | 8 |
| **Searching** | 8 | 8 | 2 | 8 | 7 |
| **Content** | 8 | 7 | 4 | 6 | 10 |
| **Communication** | 7 | 5 | 5 | 4 | 3 |

VolunteerMatch(<https://www.volunteermatch.org/>):

Design is simple but effective, having the search bar front and center with results based on your last search and a few images. Navigation is covered by the header bar with clearly labeled sections. Search allows filtering by group, skills needed, and cause of the opportunity and displays both opportunities and organizations. The opportunity listings go into detail about important factors such as location, skills needed, and information about the organization. Site passes on your email and phone number when applying which allows companies to contact the volunteer.

MyVolunteerPage (<https://app.betterimpact.com/Login/Volunteer>):

The design is overall bland with only a few shades of gray and text boxes to greet visitors with profile overview pages having some color once logged in. There are no navigation buttons or headers without being logged in. Once logged in, there is a navigation header. The search bar on the homepage does not let you filter and gives inconsistent results. The content is dependent on the organization as some of them have blank pages when searched for and opportunities don’t show unless signed in and with an organization. There is the option to send and receive emails from administrators as well as text messages.

JustServe(<https://www.justserve.org/>):

Design is simple but effective with the search bar on top of the home page and various images and recommended opportunities as it scrolls down. The site has simple navigation that works between the header bar and defined buttons. Searches can be done with a variety of different inputs are able to be filtered by various categories such as skills, schedule, interests. Opportunities have details about the skills needed, location, dates, and contact info. Communication with organizations is mainly done through email.

International Volunteer HQ(<https://www.volunteerhq.org/>):

Design strikes a good balance between simple and cluttered with a good variety of pictures and different sections showing off awards, featured trips, reviews, etc. All buttons, drop down menus, and options clearly state their purpose and where they link to. The search uses two selector menus instead of a text-box input field, but there is still filtering for parameters such as accommodations and locations. Opportunity listings go in depth with the details for the opportunity going over travel accommodations, daily schedule, housing, price, requirements, and more. Communication is lacking since there is one company that handles all of the opportunities and thus there are only emails or messages from employees.

Advantages:

Our site allows all potential volunteers, regardless of if they are registered or not to browse for opportunities using our advanced search that allows filtering by skills, location, type of job, and more. The clean interface and fast search make finding potential opportunities a breeze that can be freely catered to the user’s preference with opportunities that have all of the important details such as schedule, skills, and location clearly listed in the description. Once a volunteer is ready to sign up, they’ll find that everything is done through the site so organizations will be able to send messages and reminders directly through the site, so all communications are handled in one place.

# Data Definition

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Meaning | Usage | Comment |
| volunteer | actor | Use case scenarios | Person looking to volunteer |
| user | actor | Use case scenarios | Definition of volunteer |
| member | actor | Use case scenarios | Volunteer registered in the system |
| Non-member | actor | Use case scenarios | Volunteer not registered in the system |
| organization | data | Searchable place | Represents the place where a member will volunteer |
| post | data | Searchable member | Represents the member’s information shared in the system |
| account | data | Use case scenarios | Store volunteer’s information |
| Skills | data | Use case scenarios | Store volunteer’s skills |
| availability | data | Use case scenarios | Store volunteer’s availability |
| Search | service | Site user service | Allow user to find organizations based on their skills |
| navigator | Service | Site user service | Allow visitor to navigate the web page |
| Volunteer list | service | Site user service | Allow organizations to find volunteers. |
| Organization list | services | Site user service | Allow volunteer to find organization |
| filter | Service | Site user service | Help user be more specific on their search |
| Log in | service | Site user service | Allow user to create a post or to find an organization whose needs matches the volunteer’s skills. |
| Web site | User interface | User interface | Front end display for user interaction |
| Home page | User interface | User interface | First page that appears to user. |

# Use Cases

**Use Case- Registration**

The volunteer: our user, comes to the home page and proceeds to press on the registration tab. The system follows up by showing all the required information the user needs to enter in order to create an account.

**1.     Description:**

Use case describes the process of how a user registers to the volunteer management system.

**2.     Actors:**

2.1 User

2.2 System

**3.     Preconditions:**

3.1 User has good internet connection

3.2 System is available

**4.     Primary flow of events:**

    4.1. User arrives on home page

              4.2. User tabs on the registration button (Use case Registration)

              4.3. User gets moved to a registration display

              4.4. User enters name

                   4.4.1. If numbers or non-letter characters are entered go to alternative flow 5.2

              4.5. User enters phone number

      4.5.1. If non-number characters are entered go to alternative flow 5.2

              4.6. User enters e-mail

                    4.6.1. If e-mail doesn’t exist or is not registered go to alternate flow 5.3

              4.7. System saves User’s information, and an account is created.

              4.8. Terminate Use Case: Registration

**5.     Alternate Flows of events:**

**5.1 User enters wrong format on name bar**

                    If in step 4 user enters numbers or non-letter characters on the name bar.

1.     Web page notifies user that characters aren’t allowed.

2.     Return to step 4.4

**5.2 User enters wrong format on phone number bar**

If in step 4 user enters letter characters on the number bar.

1.     Web page notifies user that characters aren’t allowed.

2.     Return to step 4.5

**5.3 User enters a non-registered email**

 If in step 4 user enters a non-registered email on the e-mail bar.

1.     Web page notifies the user that the email doesn’t exist.

2.     Return to step 4.6

**Use Case – User’s Profile**

After the registration process, the user will be able to add their skills and availability into their profile. The system will update their profile with this information.

**1.     Description:**

This use case allows the registered user to add a skill and availability, so it is shown in the user’s profile.

**2.     Actors:**

2.1 User

2.2 System

**3.     Preconditions:**

3.1 User has good internet connection

3.2 System is available

3.3 User has an active account

3.4 User is logged into the system

**4.     Primary flow of events:**

4.1 User has created an account (Use case Registration)

4.2 System checks if user is logged in

4.2.1 If user isn’t logged in go to alternative flow 5.1

4.3 User fills the information regarding their skills

4.4 User fills the time available to volunteer.

4.5 System saves skills and availability into the user's profile.

4.6 Terminate Use case – User’s Profile

**5.     Alternative Flows of events:**

**5.1. User is not logged in from (Primary Flow of events 4.2.1)**

           5.1.1 System ask user to enter login information

                5.1.1.1 If user doesn’t have an account, go to alternative flow 5.2

           5.1.2 User enter their login information

           5.1.3 System validates login information

                5.1.3.1 If login information is wrong or doesn’t exist, return to 5.1.1

           5.1.4 End alternative Flow 5.1- return to primary flow step 4.3

**5.2 User doesn’t have an account on the system**

           5.2.1 Return to Use case- registration

           5.2.2 End alternative flow 5.2 return to primary flow, step 4.3

           5.2.3 System end use case- User’s profile

**Use case – Search**

User come to the page to search for an organization they can volunteer into, after an account is created the user will be able to login and post their skills and availability on the system to find organizations which are looking for volunteers that fits those skills and the time needed.

**1.     Description:**

Use case describes the process of how a user searches for an organization based on their skills and time availability.

**2.     Actors:**

2.1 User

2.2 System

**3.     Preconditions:**

3.1 User has good internet connection

3.2 System is available

3.3 User has an active account

3.4 User is logged into the system

3.5 User has filled their skills and time availability on the user’s profile.

**4.     Primary flow of events:**

4.1 User has created an account (use case- registration)

4.2 User has filled in the information in the profile. (Use case- User’s profile)

4.3 User enters search criteria based on user’s availability and skills.

4.4 Systems receives request and proceeds to search for an organization.

4.5 If the systems finds a match will display the organization(s)

4.5.1      If the system doesn’t find a match go to alternative flow 5.1

4.5.2      If the system states there are no organizations available, go to alternative flow 5.2.

4.6 User selects organization.

4.7 System saves the volunteer request volunteer

**5.     Alternative flow of events:**

5.1 System doesn’t find a matching organization for user

If the web page doesn’t find an organization matching the user’s skills or availability, the user must modify availability or skills. (Primary flow 4.5.1)

5.1.1      Web page notifies user there are no matching organizations.

5.1.2      User proceeds to change their skills and availability (use case – User’s profile)

5.1.3      Return to primary flow events 4.4

**5.2. System shows there are no organizations available and will allow user to make a post.**

                     If the web page displays no organizations at all, the user will be allowed to create a post with their profile.

                   5.2.1 Web pages notifies users there is no organizations available

                   5.2.2 Web pages notifies users the option to create a post

                   5.2.3 User creates a post (use case- Volunteer post), return to 4.7

                   5.2.4 System terminate use case- search.

**Use case – Post**

User comes to the page to search for an organization they can volunteer into, after no finding any available organization or no organizations at all. System will allow user to create a post that will be visible to any organization looking for volunteers.

**1.     Description:**

Use case describes the process of how a user creates a post with their profile information and list in on the system to be visible to organizations.

**2.     Actors:**

2.1 User

2.2 System

**3.     Preconditions:**

3.1 User has good internet connection

3.2 System is available

3.3 User has an active account

3.4 User is logged into the system

3.5 User has filled their skills and time availability on the user’s profile.

3.6 There are no organizations available for user or none at all.

**4.     Primary flow of events:**

4.1 User has created an account (use case- registration)

4.2 User has filled in the information in the profile. (Use case- User’s profile)

4.3 User enters search criteria based on user’s availability and skills.

4.4 System couldn’t find any organization for user or there is none available

4.5 User is offered the option to create a post

4.5.1 If user don’t want to create a post go to alternative flow 5.1

4.6 User fills post with their profile information and post it

4.7 System saves post into the web page

4.8 Organizations can find volunteers’ posts at any time.

**5.     Alternative Flow of events:**

5.1 If user doesn’t want to create a post

If the user doesn’t want to create a post, they will be allowed to go back and modify their profile or exit web page.

5.1.1      After the user declines to create a post system will offer the option to modify users’ profile. Return to case- user’s profile

5.1.1.1 If user doesn’t want to modify profile, user may exit the web page

                   5.1.2.   User modify profile and return to use case- Search

                   5.1.3 System terminate case - Post

# List of Functional Requirements

1. User Registration and Profile Management:

  Volunteers that are interested in volunteering and joining are able to join by creating accounts and providing necessary information such as their name, skills they would like to use, and their contact information. After creating their profiles/accounts, the volunteers should be able to edit and modify their online profiles at any given time after signing up or add any details.

1. Volunteer Opportunity Listing:

  There will be a list of volunteer opportunities by category that are available together with information on all of the different events. There will be information on dates, times, qualifications needed, as well as the number of volunteers needed for the event.

1. Volunteer Opportunity Sign-up:

Volunteers will have the option to look through a volunteer opportunity list and choose volunteer activities and events that interests them. They will have many options to choose from, and they can have their skills in mind when choosing an opportunity so it can fit their abilities. The volunteers can then sign up for the tasks or activities they can participate in (with their availability in mind) and get confirmation for their involvement.

1. Task Assignment and Tracking:

 Ensure effective use of volunteer resources and information by enabling administrators or organizers to allocate volunteers to certain tasks and events in these volunteer opportunities by tracking their involvement in activities and tasks.

1. Volunteer Hour Tracking:

 Volunteers will have a way to record and monitor their volunteering schedule, as well as start and end times. By having these details on their account, they can create statements and records on their schedules and hours volunteering. They can use that information for their own personal use, or in case they need to show their volunteering schedule and volunteering hours.

1. Communication and Notifications:

There will be use of messaging and communication tools for the purpose of communication between different administrators and organizers, as well as between them and the volunteers.

1. Reporting and Analytics:

There will be a tool to produce reports on the volunteer’s participation, volunteering hours, and other indicators on their participation that can be used for the reports. It will be used to analyze the volunteers' work and assist the organizations by providing this important information and analytics that they can use for their advantage.

1. Administrator Dashboard and Management:

Provide administrators and organizers with a dashboard where they may manage the events, tasks, volunteer opportunities and personal information of the volunteers. The administrators and organizers will be offered choices for the configurations that compose the system, the management of users and their accounts, and administrative controls.

1. Integration and Customization:

To improve functionality, there will be integration with other programs or resources, such as communication and messaging tools. There will also be the option for system customization in order to meet the requirements of organizations and their brand. Organizations will have the opportunity to update their requirements based on the organization's needs.

# List of Non-Functional Requirements

1. Performance
   1. The system must be responsive and work within a reasonable amount of time to engage our audience. Unit testing must be performed to keep track of efficiency and the load up time of the landing page should be no more than 8 seconds.
   2. Monitor the system’s resources such as memory and CPU to keep track of usage. The entire system should be no more than 5 GB with 1.5 GB allocated towards persistent volume.
2. Usability
   1. The system will be built with a responsive design to engage end users across different devices such as mobile and desktop.
   2. The interface must be simple and engaging to interact with volunteers and organizers to use the system productively. It should be directed towards productively and end users completing their purpose.
   3. Must be able to understand or troubleshoot any area of the system within a reasonable time from proper and concise documentation throughout the different parts of the project. The same should be reflected for end users and there includes help or “tip” sections at the more complex parts of the system.
   4. Must be accessible to the top 3 languages in North America
3. Accessibility
   1. Should be portable to run on the 4 most popular web browsers.
4. Expected Load:
   1. Incorporate load testing identity performance limitations.
   2. For now, the system will be able to handle 10 users at a time.
5. Security Requirements:
   1. The system needs to have secure user authentication and authorization functionalities. These methods may include encryption techniques, role-based access and monitoring of data transmissions. Specifically input validation and sanitization to protect against SQL injections.
   2. System dependencies must be kept organized and updated to reduce security threats.
6. Storage:
   1. Allocated space for persistent volume is 5 GB.
7. Availability:
   1. The system must be available 100% of the time during business hours otherwise it should be available at least 70% of the time.
8. Reliability:
   1. If a page or request from the system fails there needs to be a way to mitigate around the issue and return to the index page.
   2. If the server fails, debug output needs to be recorded and the last few requests should be captured.
   3. If there is a system failure, a job to correct it should not run longer than 10 minutes.
9. Scalability:
   1. The system will be relatively small and run directly on a single server.

# High-level System Architecture

1. Programming Languages:
   * Backend: Python
   * Frontend: JavaScript, HTML, CSS
2. Frameworks and Libraries:
   * Backend: Flask (Python Micro web framework)
   * Frontend: React.js (JavaScript library), Bootstrap (CSS)
3. Database Management System:
   * MongoDB: A no SQL document-oriented database program.
4. Authentication and Authorization:
   * User Models and routing within the Flask server
5. Mapping and Location Services:
   * Google Maps API: Integrating maps and geolocation features to display event locations and directions.
6. Version Control and Collaboration:
   * Git: A distributed version control system for managing codebase and facilitating collaboration among team members.
   * GitHub: GitHub is a code hosting platform for version control and collaboration
   * Trello: Project management space for assigning tasks and visualizing workflows among team members.
7. Deployment and Hosting:
   * Heroku: A cloud-based platform for deploying and hosting web applications.
   * Amazon Web Services (AWS): Utilizing AWS services like Amazon S3 for storing static assets or media files.
8. Licensing:
   * Django: Released under the Django license (BSD-3-Clause).
   * React.js: Licensed under the MIT License.
   * Flask: Licensed under the BSD license

# Team Roles

**Team Leader, Back End Developer**:

* Jonathan Garcia

**Front End Developers:**

* Ernesto Benavides
* Jacob Campbell
* Kareen Godder

**Back End Developers:**

* Tivvon Cruickshank

# Checklist

1. Team decided on basic means of communications: Done
2. Team found a time slot to meet outside of the class: In Progress
3. Front and back-end team leads chosen: In Progress
4. GitHub master chosen: Done
5. Team ready and able to use the chosen back and front-end frameworks: Done
6. Skills of each team member defined and known to all: In Progress
7. Team lead ensured that all team members read the final M1 and  
   agree/understand it before submission: In Progress

Trello Link: <https://trello.com/invite/b/O4ixT6Yk/ATTI0b714da185aa6f3b75ea034abf092c3aAA7665C7/cen-4010-project>

# Revision History

* Version 1.0 – Initial Documentation Completed; 6/20/2023