CIS 4327 – IS Senior Project I

Pre-Semester Team Project Assignment

Due on **August 22 (The first day of the class)**

Submission will be GitHub Repository or other appropriate means of sharing the source codebase.

Volunteer Management System

Introduction

The Volunteer Management System (VMS) is a web-based application that allows a nonprofit organization to collect volunteers' information, track their participation in activities, and produce reports on their contributions. The purpose of this document is to define the high-level requirements and features of a portion (only administrative features) of the Volunteer Management System (VMS).

This document describes two use case specifications relevant to the administrative features of the VMS. Use case specification provides a detailed description of the high-level requirements of a feature. Requirements are typically described as interactions between the user and the system as a series of user actions and system events. We will learn more about the use cases in the fall semester.

Details of Manage Volunteer Profile Information and Manage Opportunities use cases are provided in this document. Read the user interactions and system actions described in the use cases to understand the administrative feature needs of the VMS. Work as a team to develop administrative features of the VMS. Use the web development platform (ASP.NET or PHP or Javascript stack) selected as a team to develop these features. Make decisions as a team to develop these two features. You are allowed to make necessary assumptions on missing information about the requirements and the VMS.

Objectives and Expectations of the Pre-Semester Assignment

The objectives of the assignment are two-fold. The first objective is to provide you a sample set of requirements that is much closer to what you will face in the senior project class for practicing your web development programming skills. Below two use case specifications were taken from one of the previous senior project student submissions. The second is to allow you to self-assess your individual and team capabilities to develop working software. Mentors will use the codebase to assess your team's capabilities as well.

As mentioned in my previous communications, in the senior project course, you will learn about the software development lifecycle processes. Web development programming skills are a prerequisite for the course. As the class involves real-world projects and real commitments, prerequisite skills will be important for delivering working and useful products to your client. Therefore, you must practice and improve programming skills in the remaining few weeks before the fall semester starts.

I don't expect your solution to be perfect, but demonstrate the team's capability. Try to put in at a minimum of 5 hrs per week to make progress on this assignment. Do your best to communicate, divide up work, and work as a team (as much as possible).

Use Case 1: Manage Volunteer Profile Information

Brief Description

This use case shows the steps involved when an administrator manages the profile information of volunteers.

1. Basic Flow of Events

1.1. Administrator logs on

The use case begins when an administrator enters the following user credentials:
-Username
-Password
The system stores this information in the session so that the administrator only can see information appropriate to them.

1.2. System displays administrator page

The system displays a selection of actions to perform. The following actions relevant to this use case are displayed: Manage Volunteers and Manage Opportunities.

1.3. Administrator chooses Manage Volunteers option

The administrator selects the option of going to the area where he/she can manage the volunteers.

1.4. System displays a summary view of volunteers based on the default filter

The system shows a list of the volunteers along with possible actions the administrator can take for each of them. The relevant actions are the following:

a. Change Volunteer Filter
b. Search Volunteers
c. View Volunteer Opportunity Matches
d. Edit Volunteer
e. Add New Volunteer

The volunteers displayed at this step are shown based on the system's default filter of criteria for which categories of volunteers show up.

1.5. Administrator Chooses Option to Change Volunteer Filter

The administrator selects the option to Change Volunteer Filter.

1.6. System Displays List of Filters

The list of filters includes the following:

a. Approved/Pending Approval (volunteers in both approval statuses are included)
b. Approved
c. Pending Approval
d. Disapproved
e. Inactive
f. All

1.7. The administrator chooses the particular filter to use.

1.8. The system displays an updated list of the volunteers based on the administrator's choice.

1.9. Use Case Ends

The use case ends.

2. Alternative Flows

2.1. Search Volunteers

1. The administrator chooses Search Volunteers option the basic flow step 1.4 (System Displays Administrator Page).
2. The administrator specifies the volunteer to search for.
3. The system displays list of volunteers matching the search criteria.
4. The use case ends.

2.2. Volunteer Not Found

1. This alternative flow starts after step 1 of alternative flow 2.1 (Administrator Specifies Volunteer to Search For), when the system finds no volunteer matching the search criteria
2. System shows a message informing the administrator that no volunteers matched the search criteria.
3. The use case ends.

2.3. View Opportunity Matches

1. The administrator performs at the Basic Flow and Alternative Flow 2.1 (Search Volunteers) as needed to navigate to a list that includes the volunteer for which he/she wants to perform the action.
2. The administrator chooses the View Volunteer Opportunity Matches option to see which opportunities the system has matched with the volunteer.
3. The system displays the list of opportunities it has matched with the volunteer.
4. The use case ends.

2.4. Edit Volunteer Profile

1. Administrator performs the Basic Flow and Alternative Flow 2.1 (Search Volunteers) as needed to navigate to a list that includes the volunteer for which he/she wants to perform the action.
2. Administrator selects the Edit Volunteer option.
3. System displays the volunteer's information to the administrator in a format that can be edited. This information is as follows:
    a. First Name
    b. Last Name
    c. Username

    d. Password
    e. Centers where the volunteer prefers to work
    f. Skills/Interests
    g. Availability times
    h. Address
    i. Home/work/cell phone numbers
    j. Email
    k. Educational Background
    l. Current Licenses
    m. Emergency Contact Name
    n. Emergency Contact Home/Work Phone
    o. Emergency Contact Email
    p. Emergency Contact Address
    q. Whether there is a copy of volunteer's drivers license on file
    r. Whether there is a copy of volunteer's social security card on file
    s. Volunteer's Approval Status
4. Administrator makes changes to the volunteer's information as needed.
5. Administrator selects option to update the volunteer's information by saving his/her changes.
6. System ensures that the updated information is valid.
7. If the information is valid, the system saves the updated information.
8. System removes the page showing the volunteer's information and goes back to where the administrator was before he/she selected the Edit Volunteer option.
9. The use case ends.

2.5. Add a New Volunteer

1. The administrator performs the Basic Flow and Alternative Flow 2.1 (Search Volunteers) as needed to navigate to a list that includes the volunteer for which he/she wants to perform the action.
2. The administrator selects the Add New Volunteer option.
3. System displays a section in which the administrator can enter various aspects of volunteer information. Information includes:
    a. First Name
    b. Last Name
    c. Username
    d. Password
    e. Centers where the volunteer prefers to work
    f. Skills/Interests
    g. Availability times
    h. Address

i. Home/work/cell phone numbers
j. Email
k. Educational Background
l. Current Licenses
m. Emergency Contact Name
n. Emergency Contact Home/Work Phone
o. Emergency Contact Email
p. Emergency Contact Address
q. Whether there is a copy of volunteer’s drivers license on file
r. Whether there is a copy of volunteer’s social security card on file
s. Volunteer’s Approval Status

4. Administrator adds the volunteer data.
5. Administrator selects option to create the new volunteer by saving his/her changes.
6. System ensures that the information is valid.
7. If the information is valid, the system creates the new volunteer.
8. System removes the page showing the volunteer’s information and goes back to where the administrator was before he/she selected the option to create the new volunteer.
9. The use case ends.

2.6. Handle Cancelled Add Volunteer Action

If, at any step, the administrator selects the option to cancel the addition of the new volunteer:

1. The system clears any changes made by the administrator and does not save them.
2. The system removes the page for adding the new volunteer and goes back to where the administrator was before he/she selected the option to create the new volunteer.
3. The use case ends.

2.7. Handle Cancelled Edit Volunteer Action

If, at any step, the administrator selects the option to cancel the action of editing the volunteer’s profile:

1. The system clears any changes made by the administrator and does not save them.
2. The system removes the page for editing the volunteer’s profile and goes back to where the administrator was before he/she selected the option to edit the volunteer information.
3. The use case ends.

2.8. Unauthenticated Administrator

If, at basic flow step 1.1 (Administrator Logs On), the system finds that the login credentials the administrator has provided are invalid:

1. The system refuses to log the administrator into the system.
2. The system displays a message to the administrator indicating that his/her credentials were not valid, along with the option to re-enter them.
3. Process resumes at step 1.1 (Administrator Logs On).

2.9. Invalid Volunteer Data Input

If, at any step, the system finds that the administrator has provided invalid volunteer data input:

1. The system refuses to save the added/updated volunteer information.
2. The system displays an indication to the administrator that he/she has entered invalid information.
3. Process goes back to step where invalid data was entered and continues from there.

2.10. Quit Session

1. At any point, the user decides to quit the session unexpectedly.
2. The system discards any unsaved changes the user has made.
3. The use case ends.

Use Case 2: Manage Opportunities

1. Basic Flow

1.1. Administrator logs on

The use case begins when an administrator enters the following user credentials:
-Username
-Password

The system stores this information in the session so that the administrator only can see information appropriate to them.

1.2. System displays administrator page

System displays a selection of actions to perform. The following actions relevant to this use case are displayed: Manage Opportunities.

1.3. Administrator chooses Manage Opportunities option

Administrator selects the option of going to the area where administrator can manage opportunities.

1.4. System displays summary view of opportunities based on the default filter

System shows a list of the opportunities along with possible actions the administrator can perform:

a. Change Opportunity Filter
b. Search Opportunity
c. View Volunteer Opportunity Matches
d. Edit Opportunity
e. Add New Opportunity
f. Delete Opportunity

The opportunities that are displayed at this step are shown based on the system's default filter of criteria.

1.5. Administrator Chooses Option to Change Filter

The administrator selects the option to change current opportunity filter.

1.6. System Displays List of Filters

The list of filters includes the following:

a. Most Recent (60 days)
b. By Center

1.7. The administrator chooses the particular filter to use.

1.8. The system displays an updated list of the opportunities based on the administrator's choice.

1.9. Use Case Ends

The use case ends.

2. Alternative Flow

2.1. Search Opportunities

1. The administrator chooses Search Opportunities option after the end of step 1.4 of the Basic Flow.
2. The administrator types in keyword or name of opportunity.
3. The system updates the opportunities that are displayed to only include those that match the search criteria.
4. The use case ends.

2.2. Opportunity Not Found

1. This alternative flow starts after step 1 of the 2.1 alternative flow (Search Opportunities).
2. The system updates the opportunities that are displayed to include no opportunities, since none were found that matched the administrator's criteria.
3. System shows a message informing the administrator that no opportunities matched the search criteria.
4. The use case ends.

2.3. View Opportunity Matches

1. The administrator performs the Alternative Flow 2.1 (Search Opportunities) as needed to navigate to a list that includes the opportunities for which he/she wants to perform the action.
2. The administrator chooses the View Volunteer Opportunity Matches option to see which volunteer the system has matched with the opportunity.
3. The system displays the list of volunteers it has matched with the opportunity.

4. The use case ends.

2.4. Change Opportunity Filer

1. The administrator performs 1.4 of basic flow.
2. Choose different filter option.
3. System display refreshes corresponding to selected option.

2.5. Edit Opportunity

1. Selects edit from table next to opportunity.
2. System display window to edit information.

2.6. Administrator edits opportunity fails

Edit failure can occur for mistyped information this can happen after alternative flow 2.5. User is asked to reenter information.

2.7. Add New Opportunity

1. Selects to add new opportunity.
2. System display window to add information.

2.8. Delete Opportunity

1. Selects delete from table next to opportunity.
2. Data on page refreshes to correspond.

General system notes

The system must be designed in an easily-usable manner and must handle errors or wrong inputs appropriately.