Stefan Strek Mayoral Campaign Website

Project Documentation

Table of Contents

[1. Introduction 2](#_Toc445635694)

[A. Purpose 2](#_Toc445635695)

[B. Project Scope 2](#_Toc445635696)

[C. Definitions and Acronyms 2](#_Toc445635697)

[D. Responsibilities 2](#_Toc445635698)

[1. Project Team 2](#_Toc445635699)

[2. Project Sponsor (Brian Bird) 2](#_Toc445635700)

[3. Client 2](#_Toc445635701)

[2. Requirements 3](#_Toc445635702)

[A. System Functionality 3](#_Toc445635703)

[B. Deliverables 3](#_Toc445635704)

[1. Project Design Document 3](#_Toc445635705)

[2. Working Website 3](#_Toc445635706)

[3. Data Design 4](#_Toc445635707)

[A. Data Dictionary 4](#_Toc445635708)

[1. Tables 4](#_Toc445635709)

[B. Database Schema 6](#_Toc445635710)

[4. System Design 7](#_Toc445635711)

[A. Sequence Diagrams 7](#_Toc445635712)

[B. Use Case Diagrams 9](#_Toc445635713)

[C. Use Case Scenarios 11](#_Toc445635714)

[D. Sitemap 16](#_Toc445635715)

[Approval and Authority to Proceed 17](#_Toc445635716)

# Introduction

## Purpose

Our client, Stefan Strek, wants a website for his Eugene mayoral campaign. He wants part of the website right away, so he can start taking campaign donations. This initial site will include a home page featuring the three main positions he is running on and a photo of Stefan. He wants the website to have a red, white, and blue theme because he is targeting conservative voters.

## Project Scope

The scope of this project is a website which will enable the Stefan Strek Mayoral Campaign to provide information to constituents and interested voters, as well functionality allowing donors to place donations, and volunteers to sign up and specify activities they are interested in.

## Definitions and Acronyms

* Article – A news or other article related to the campaign
* Campaign – The Stefan Strek Mayoral Campaign
* Event – A campaign event
* Donation – A financial contribution to the campaign
* Donor – An individual who has created or wishes to create a Donation
* Issue – A determined political issue for which a position has been specified by the candidate
* Position – A political position on an issue
* System – The Stefan Strek Mayoral website
* Volunteer – An individual who uses the System to volunteer for the campaign

## Responsibilities

### Project Team

The project team is responsible for designing a functioning system, producing documentation describing the design, implementing the system and thoroughly testing it prior to publishing.

### Project Sponsor (Brian Bird)

The project sponsor is responsible for approving design changes and providing high level support regarding project issues and developments.

### Client

The client (Stefan Strek) is responsible for providing feedback regarding the development of the System, as well as specifying which payment API should be implemented to allow Donations to be made electronically.

# 2. Requirements

## System Functionality

1. Voters and interested parties to access the site securely through the Web
2. Voters and interested parties should be able to view a calendar of current and upcoming Events
3. Voters and interested parties should be able to view Issues and Positions the candidate holds
4. Voters and interested parties should be able to view a list of Articles involving the mayoral candidate
5. Donors should be able to make Donations and safely and electronically pay for them
6. Volunteers should be able to sign up and specify the activities they are interested in
7. Administrators should be able to view records of Donors, Donations, and Volunteers
8. Administrators should be able to create, edit and delete certain portions of site content, including Articles, Issues and Positions, and Events
9. Donor, Donation, and Volunteer information should be safely stored and easily retrievable

## Deliverables

### Project Design Document

1. Project Scope
2. Database Schema (ERD)
3. Data Dictionary
4. Sequence Diagrams
5. Use Case Scenarios
6. Site Map

### Working Website

# 3. Data Design

## Data Dictionary

### Tables

1. Articles
   1. Fields:

* ArticleID int
* Title varchar
* Body varchar
* Url varchar
* Image varchar

1. Donors
   1. Fields:

* DonorID int
* FirstName varchar
* LastName varchar
* Address1 varchar
* Address2 varchar
* City varchar
* State varchar
* Zipcode varchar
* Phone varchar
* Email varchar
* Employer varchar
* Position varchar
* EmpStatus varchar

1. Donations
   1. Fields:

* DonationID int
* DonorID int
* Amount decimal
* Date datetime
* StatusFlag bit
* Confirmation varchar

1. Events:
   1. Fields:

* EventID int
* Name varchar
* Location varchar
* Description varchar
* Date datetime

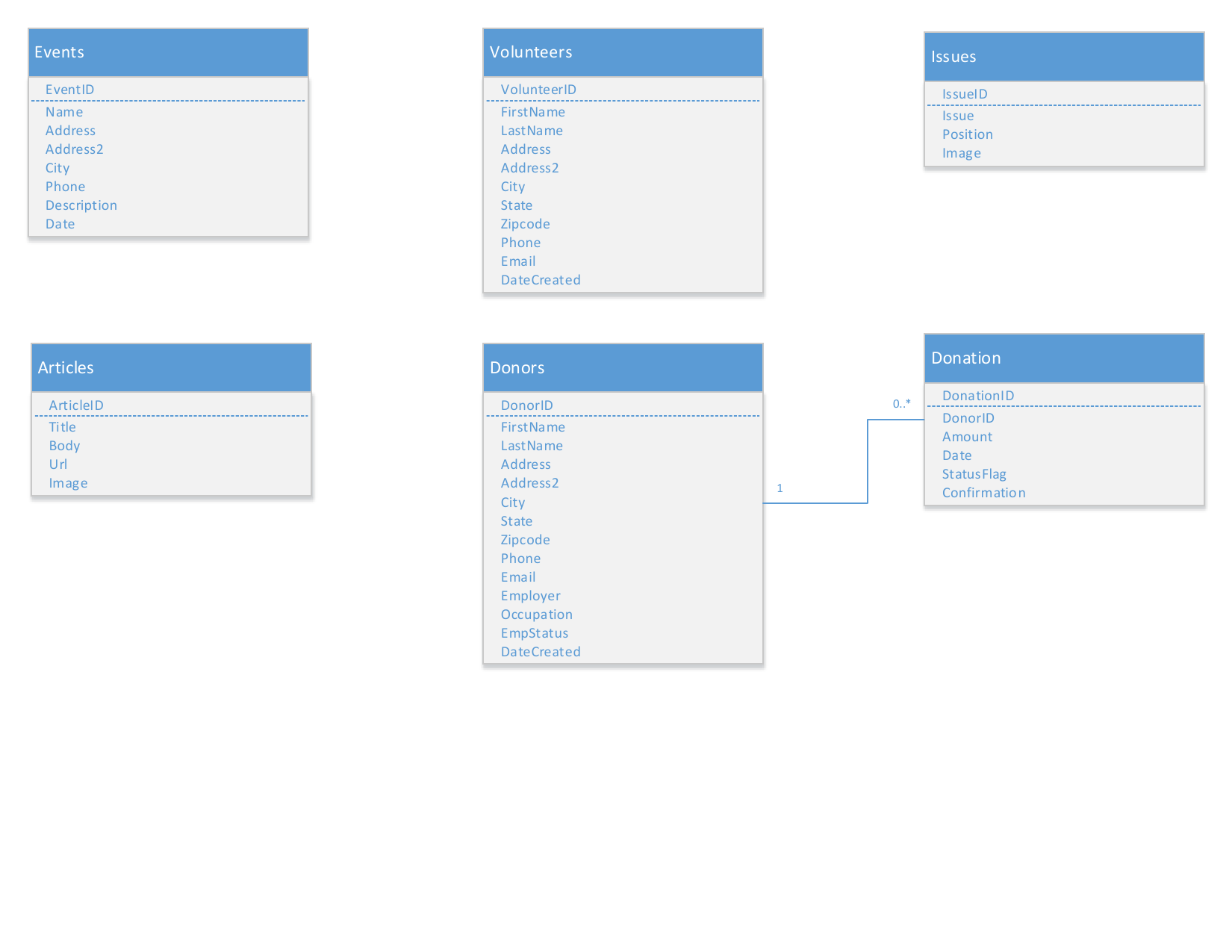
1. Issues
   1. Fields:

* IssueID int
* Issue varchar
* Position varchar
* Image varchar

1. Volunteers
   1. Fields:

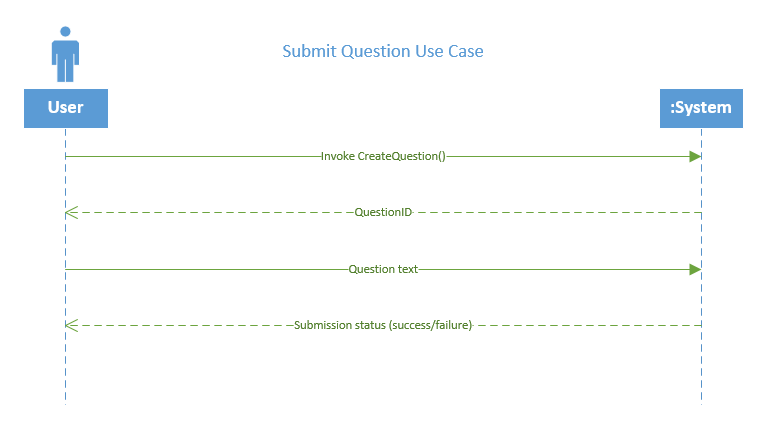
* VolunteerID int
* FirstName varchar
* LastName varchar
* Address1 varchar
* Address2 varchar
* City varchar
* State varchar
* Zipcode varchar
* Phone varchar
* Email varchar
* <Bit columns for volunteer activities>

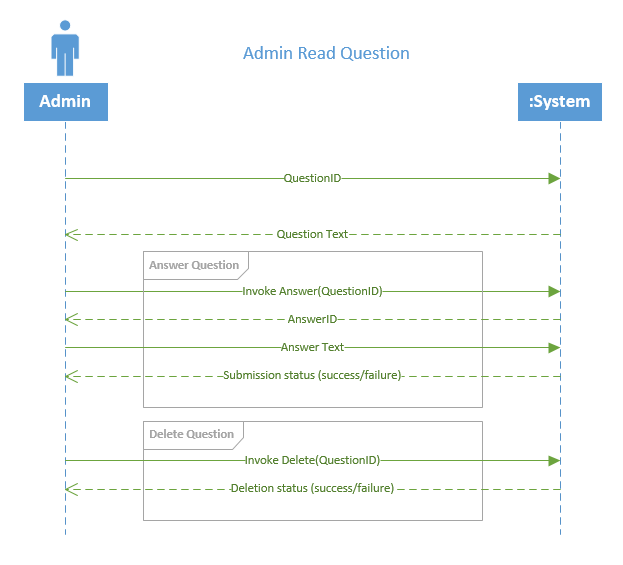
## Database Schema



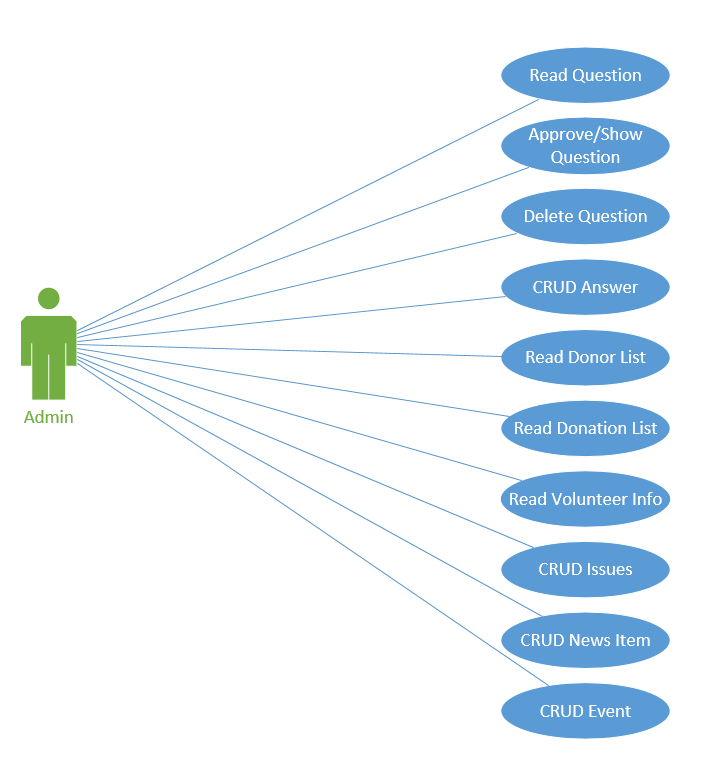
# 4. System Design

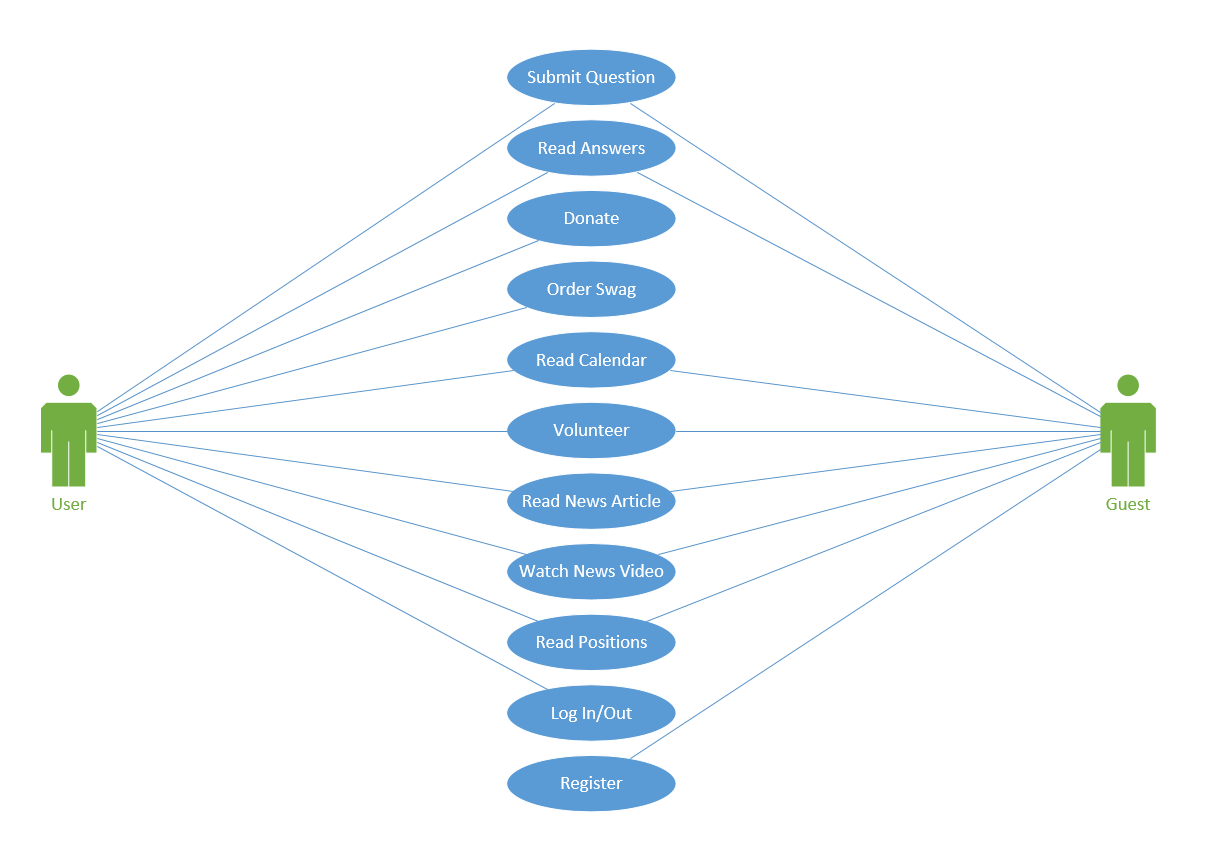
## Sequence Diagrams





## Use Case Diagrams





## Use Case Scenarios

|  |  |
| --- | --- |
| Use case name: | Read question |
| Scenario: | Read submitted question |
| Triggering event: | Admin selects to read a submitted question |
| Brief description: | Admin selects to read a submitted question from list of questions |
| Actors: | Admin |
| Related use cases: | Answer Question, Delete Question, Submit Question |
| Stakeholders: | Admin |
| Preconditions: | Question must be submitted; admin must be logged in |
| Post conditions: | Question must be displayed |
| Flow of activities: | |  |  | | --- | --- | | Actor | System | | 1. Admin requests to view submitted question 2. Admin selects Answer, Delete, or Save For Later | * 1. System retrieves and displays question   2. System prompts for action   2.1 System calls method for selected use case | |
| Exception conditions: | 1.1 Requested message does not exist in database |

|  |  |
| --- | --- |
| Use case name: | Answer Question |
| Scenario: | Answer read question |
| Triggering event: | Admin selects to answer a read question |
| Brief description: | Admin reads a user submitted question and selects to answer it |
| Actors: | Admin |
| Related use cases: | Read Question, Submit Question |
| Stakeholders: | Admin |
| Preconditions: | Question must be retrieved and displayed; admin must be logged in |
| Post conditions: | Answer must be saved; question must be flagged as answered |
| Flow of activities: | |  |  | | --- | --- | | Actor | System | | 1. Admin selects to answer question 2. Admin enters answer 3. Admin selects Submit | * 1. System creates answer object   2. System prompts for answer text   3.1 System records answer and displays success or failure | |
| Exception conditions: | 3.1 Inability to insert record |

|  |  |
| --- | --- |
| Use case name: | Delete Question |
| Scenario: | Delete read question |
| Triggering event: | Admin selects to delete a read question |
| Brief description: | Admin reads a user submitted question and selects to delete it |
| Actors: | Admin |
| Related use cases: | Read Question, Submit Question |
| Stakeholders: | Admin |
| Preconditions: | Question must be retried and displayed; admin must be logged in |
| Post conditions: | Question must be deleted |
| Flow of activities: | |  |  | | --- | --- | | Actor | System | | 1. Admin selects to delete question 2. User selects Delete | 1.1 System prompts are you sure?  2.1 System deletes question record | |
| Exception conditions: | 2.1 Inability to delete record |

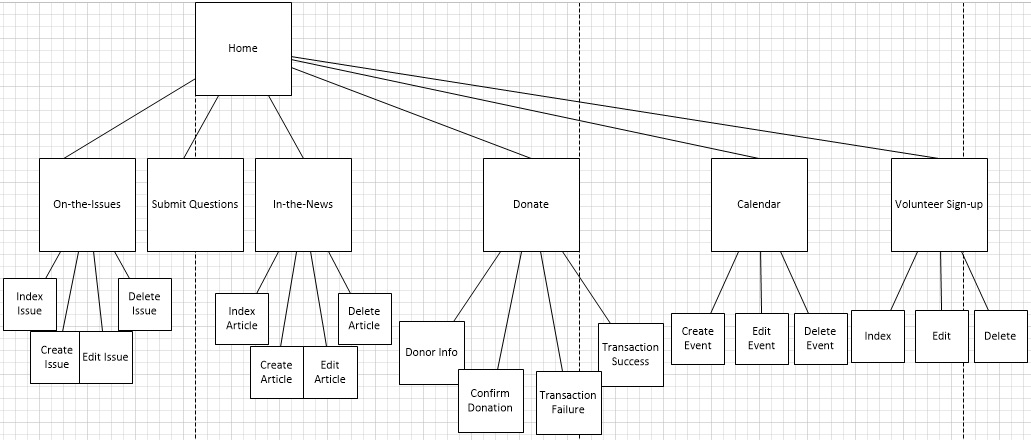
|  |  |
| --- | --- |
| Use case name: | Create Event |
| Scenario: | Create calendar event |
| Triggering event: | Admin selects to create a calendar event |
| Brief description: | Admin wants to add an event to the calendar |
| Actors: | Admin |
| Related use cases: | Display Event, Update Event, Delete Event |
| Stakeholders: | Admin |
| Preconditions: | None |
| Post conditions: | New event must be inserted into the event table |
| Flow of activities: | |  |  | | --- | --- | | Actor | System | | 1. Admin selects to create a new event 2. Admin enters event info 3. Admin selects save event | * 1. System creates new event object   2. System prompts for event info   3. System inserts event into event table   4. System displays insert success or failure | |
| Exception conditions: | 3.1 Inability to insert event record |

|  |  |
| --- | --- |
| Use case name: | Donate |
| Scenario: | Donate money |
| Triggering event: | Visitor to site selects donate |
| Brief description: | Site visitor wants to donate money to campaign |
| Actors: | Donor |
| Related use cases: | Register, Login |
| Stakeholders: | Admin, Donor |
| Preconditions: | None |
| Post conditions: | If donation is successful, donor’s contact info, employment status, employer’s name, donation amount, and date must be recorded |
| Flow of activities: | |  |  | | --- | --- | | Actor | System | | 1. Donor selects to donate 2. Donor selects or enters donation amount. If donor not logged in, donor logs in or provides necessary info | * 1. System prompts for donation amount   2. If donor not logged in, system prompts for donor to either login, register or provide contact info, employment status, and employer’s name   2.1 If new donor created, system creates and inserts donor  2.2 System redirects to third party payment processor  2.3 System receives success or failure of payment from payment processor  2.4 If payment successful, system inserts donation  2.5 System displays thank you or oops message to donor | |
| Exception conditions: | 2.1 Inability to insert donor  2.3 Inability to interface with third party payment processor  2.4 Inability to insert donation |

|  |  |
| --- | --- |
| Use case name: | Volunteer |
| Scenario: | Volunteer time |
| Triggering event: | Visitor to site selects to volunteer |
| Brief description: | Site visitor wants to sign up to volunteer |
| Actors: | Volunteer |
| Related use cases: | Register, Login |
| Stakeholders: | Volunteer, Admin |
| Preconditions: | None |
| Post conditions: | Volunteer table must be updated or new volunteer inserted |
| Flow of activities: | |  |  | | --- | --- | | Actor | System | | 1. Person selects they want to volunteer 2. If person not logged in, they login, register, or enter contact info 3. Volunteer selects and submits what activities they want to volunteer for | * 1. If person not logged in, system prompts for login, register, or contact info   2.1 System retrieves or creates volunteer object   * 1. System displays current status of volunteer object for volunteer to edit   2. System inserts or updates volunteer record   3. System displays thank you message | |
| Exception conditions: | 2.1 Inability to create volunteer record  3.1 Inability to insert or update volunteer record |

|  |  |
| --- | --- |
| Use case name: | Submit Question |
| Scenario: | Submit a question |
| Triggering event: | Visitor select to submit a question |
| Brief description: | Visitor wants to ask the candidate a question |
| Actors: | Visitor, Admin |
| Related use cases: | Answer Question, Delete Question |
| Stakeholders: | Visitor, Admin |
| Preconditions: | None |
| Post conditions: | Question must be inserted into question table |
| Flow of activities: | |  |  | | --- | --- | | Actor | System | | 1. Visitor selects they want to submit a question 2. Visitor enters questions text and selects submit question | * 1. System creates a new question object   2. System prompts for question text   3. System inserts question into question table   4. System displays thank you message and success or failure of insert | |
| Exception conditions: | 3.1 Inability to insert question into question table |

## Sitemap



# Approval and Authority to Proceed

We approve the project as described above, and authorize the team to proceed.

|  |  |  |
| --- | --- | --- |
| Name | Title | Date |
|  |  |  |
|  |  |  |
|  |  |  |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |
| Approved By |  |  | Date |  | Approved By |  |  | Date |