Sprint 1 - Team Kraken

End Goal: Create a website for a food bank to manage food bank events

Hello World Example (initial project standup): https://e3aeaiux2p.us-east-2.awsapprunner.com/
GitHub repository: https://github.com/sluke817/4320_Project

Initial Requirements

User:

- Create user account
- Register/unregister for an event
- Add event to Google Calendar
- Create/join/view group
- Guest account navigation
- Specify my availability on a weekly basis
- Recommend events to me based on my availability
- Declare that you cannot make your shift/find a volunteer to take your shift
- Take someone else's shift

Employee:

- Create admin account
- Create/delete an event
- Configure event details
- View groups
- Manage attendees (delete duplicate profiles, etc)

Both:

- Login/logout
- Change account information
- View event/attendees/details
- Calendar display or list display

Create an Event Use Case

Title: Facilitate the ability to create and configure a volunteering event

Description: We want to allow (only) volunteer coordinators to create volunteering events. This would involve clicking a button that opens up a prompt to create the event. This prompt would ask questions about the event details such as the event title, a description, a date, minimum volunteers required, etc. Most of these details would be required to create the event. Once all the required details are entered, the volunteer coordinator can hit a button to create the event. When the event is created, it is added to a list and persisted in a database as a record so that everyone can see it when they view the list of events.

Triggers: Click a button that starts the creation of an event

Actors: Volunteer coordinators
(Then after the event is created users can view)

Preconditions:

- The person creating the event is a volunteer coordinator
- There is enough space in the database to store the event record

Main Success Scenario: An event gets added to the list and added to the database.

Alternate Success Scenarios:

- Volunteer coordinator begins creating an event by entering details, but cancels the creation

Failed End Conditions:

- An event gets created in a way that the volunteer coordinator did not configure it, e.g. with the wrong details
- An event gets created even though a required detail was not specified

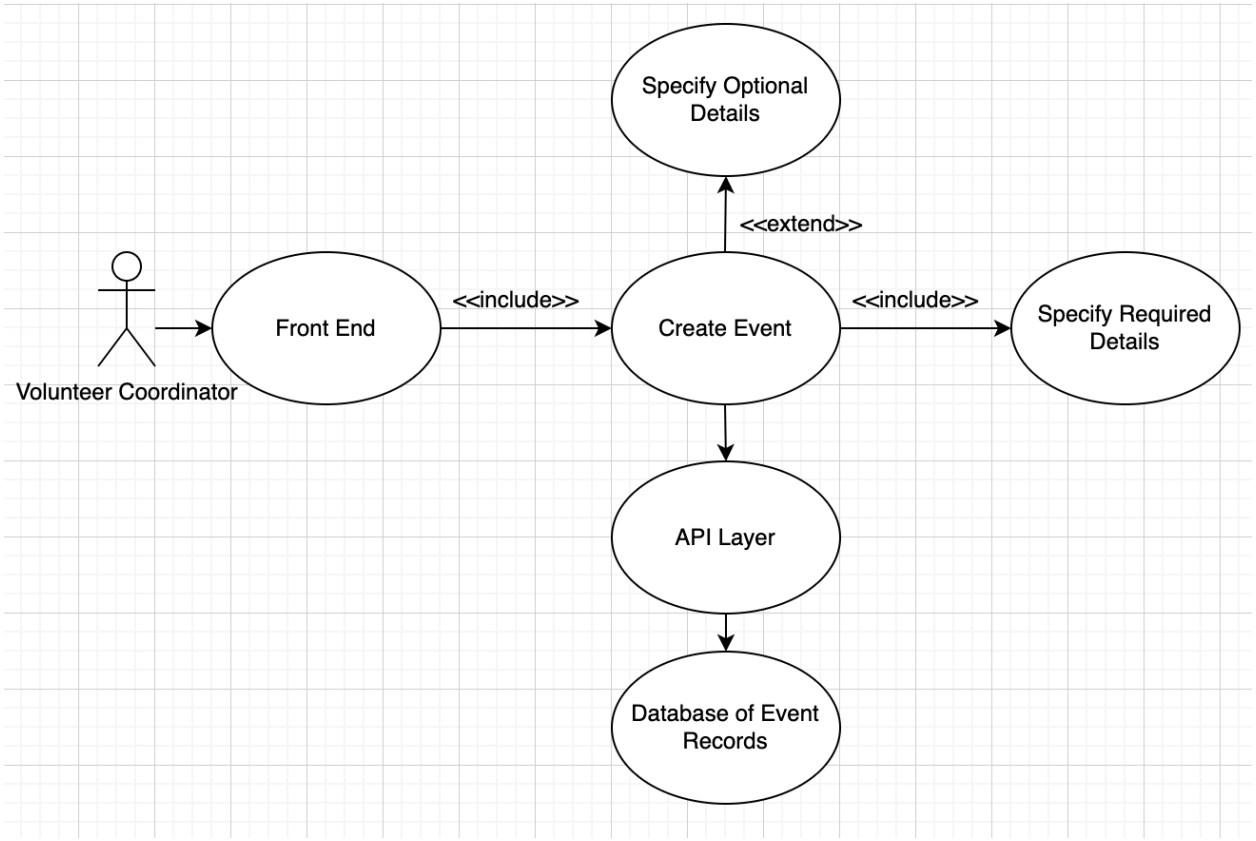
Extensions:

- Edit or delete an event after it has been created
- Specify additional details for an event
- Sort events
- Filter events

Steps of Execution (Requirements):

1. Log in as a volunteer coordinator
2. Click create event
3. Specify all the required details for the event
4. Click create event

Use case diagram:



Dependent Use Case: Login functionality (specifically be able to log in as a volunteer coordinator)

Register/Sign up For Event Use Case

Title: Provide a mechanism for signing up and viewing events

Description: We want to allow users (both volunteers and volunteer coordinators to view, and sign up for an event. To accomplish this, we will need a centralized database that contains the date and title for each event, the number of people we need, and the people that are currently signed up to volunteer for each event. Once we have this backend complete, we will need a front-end application that allows users to view the events by date that are still in need of volunteers, and then allow that volunteer to sign up for that event. This can be done through get/post APIs that the web application calls under the hood.

Triggers: A volunteer has free time and wishes to volunteer at an event.

Actors: Volunteers and Volunteer Coordinators

Preconditions: There are available events and there are volunteers that are willing to work at said events.

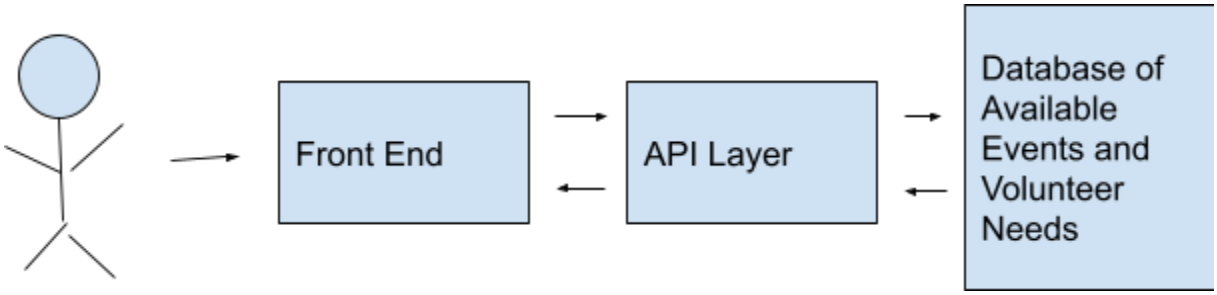
Main Success Scenario: Volunteers and Volunteer Coordinators are able to sign up for events, and that sign up is referenced quickly in the database.

Alternate Success Scenario: A volunteer or volunteer coordinator is able to view the events for a day, but decides not to sign up for whatever reason.

Failed End Condition: There are no volunteers that are willing to sign up for the events, there are no events to sign up for, the events are poorly named/described/scheduled as to make it difficult for a volunteer to choose an event.

Extensions: Produce a calendar of the events for the month, and the remaining slots that are available. I envision a volunteer coordinator being the predominant user of this extension.

Use Case Diagram:



Dependent Use Cases: Create/edit task, View and recommend an event based on availability

Filling an Empty Shift Use Case:

Title: Fill an Empty Shift

Description: We want volunteers to be able to fill empty shifts in the case that a volunteer who signed up is no longer able to attend their shift.

This would perhaps involve the application sending out an email to registered volunteers offering them the open shift? From there, a volunteer who wanted the shift could simply sign up for the shift as they would a usual shift.

Triggers: A volunteer drops a shift that they previously signed up for, leaving an empty shift.

Actors: Volunteers and Volunteer Coordinators

Preconditions: Events exist. There are volunteers registered in the database that can be contacted to fill the shift.

Main Success Scenario: Volunteers are contacted to fill the empty shift, and a volunteer willing to fill the shift is found. The found volunteer signs up for the shift.

Alternate Success Scenarios: A volunteer willing to fill the shift is found, but the volunteer doesn't know how to sign up for an event. The coordinator adds the volunteer to the event manually, perhaps?

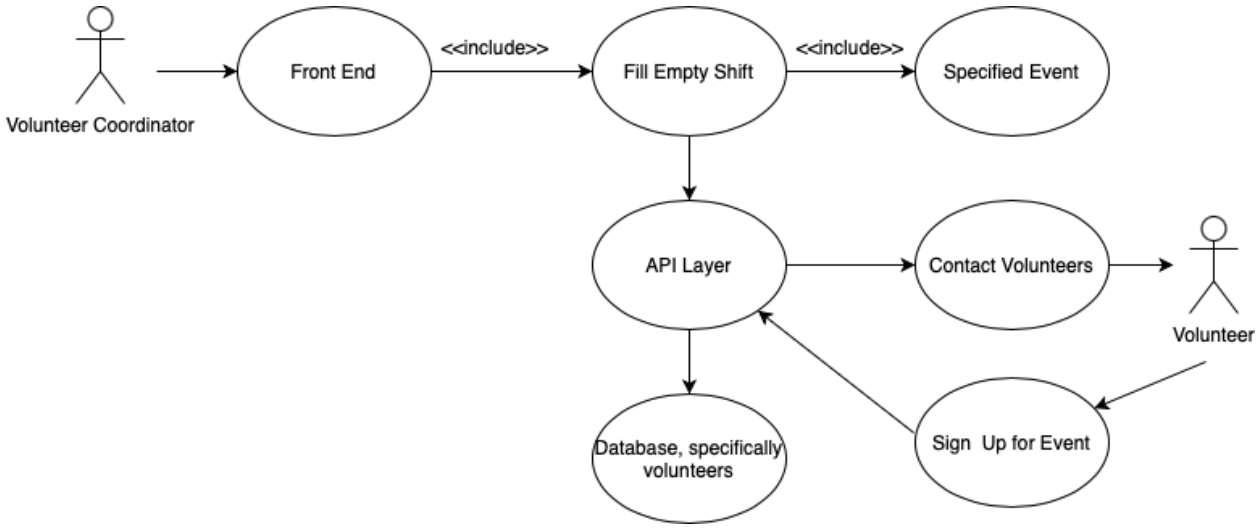
Failed End Condition: No volunteers are found, and the shift remains empty.

Extensions: The request sent out for volunteers to fill the empty shift allows volunteers to click a button that automatically signs them up for the specified shift.

Steps of Execution:

1. Log in as a volunteer coordinator
2. Select an event with at least one open shift
3. (Click a button) which sends a command to fill the open shift
4. Possible volunteers are contacted
5. Volunteer signs up for shift

Use Case Diagram:



Dependent Use Cases: Register/sign up for an event

Title: View recommended events based on availability

Description: In order to make registering for events more accessible to our users, we propose creating a feature that allows users to view recommended events based on their availability. After entering their availability, users will be able to view a list of recommended events that fit around their schedule.

Triggers: A volunteer has a busy schedule and is struggling to find events that match their availability.

Actors: Volunteers.

Preconditions: Volunteer has an account and entered their availability. There are available events on the site.

Main Success Scenario: Volunteer is recommended an event to join that fits around their schedule. Volunteer then registers for that event.

Alternate Success Scenario: Volunteer is notified that there are no upcoming events that match their availability, however, future events may be an option.

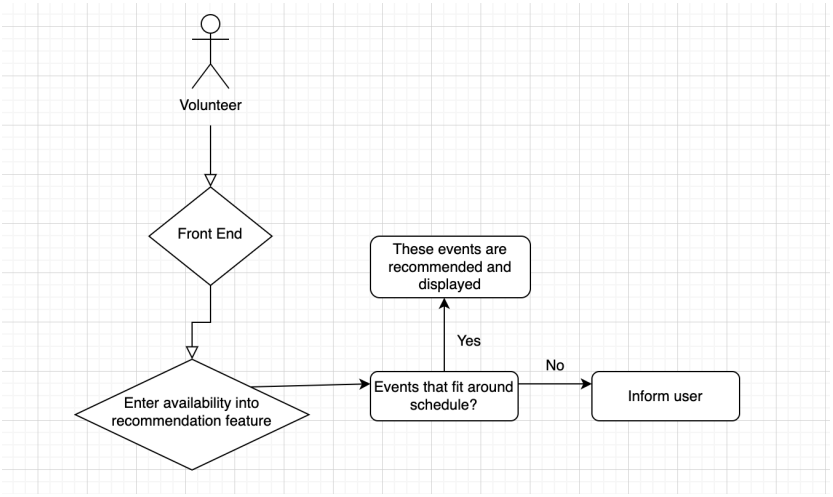
Failed End Condition: Volunteer is recommended an event that doesn't fit around their schedule.

Extensions: Organizers can see when the best times to host an event are to maximize the amount of volunteers.

Steps of Execution:

1. Log in as volunteer
2. Navigate to the recommendation feature
3. Enter availability
4. See recommendations

Use Case Diagram:



Dependent Use Cases: Volunteer can view events. Volunteer can enter availability.

Title: Schedule Volunteer Team Use Case

Description: A volunteer coordinator sees an open day for volunteering. They have lots of volunteers ready to volunteer, and want to schedule a group volunteer session at the food bank. They have the number of volunteers, their names, and emails ready in an acceptable file format. Instead of having to manually enter each of the volunteers one by one, they want to be able to schedule the whole team in the slot, as long as their team size fits in the time slot. They are able to navigate to the desired day of volunteering, and submit a request to the food bank to volunteer as a group to save time for each of the individual volunteers and the coordinator.

Triggers:

- A group leader has a group of volunteers and is ready to schedule them as a group for a desired day with the permitted group size.

Actors:

- The group leader
- Frontend event page
- Frontend signup
- Backend
- Food bank employee

Preconditions:

- The group leader has a day of service in mind
- The group leader has a list of volunteers with appropriate information
- Website has appropriate days listed

Success scenarios

- The group leader is able to schedule the group for an event/day
- The food bank is able to acknowledge the group for the day and knows that they have signed up

Alternate success scenarios:

- The group leader is able to cancel the group sign up
- The food bank is able to see the request and is able to deny the request
- The food bank is able to inform the group leader is able to cancel the event and inform the group leader

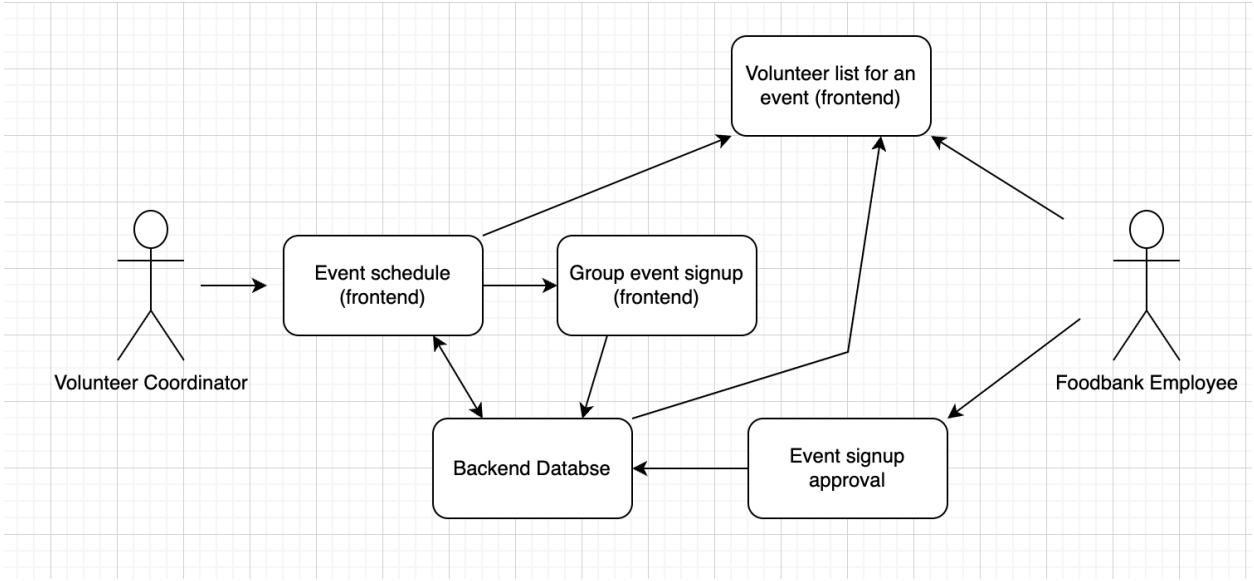
Failed end condition:

- The group leader is able to sign up for an event, but the food bank is unable to acknowledge their sign up
- The group leader believes that the group did not get signed up, but the food bank sees a group signed up (the group leader is uninformed of a successful sign up)

Steps of execution (requirements):

- User is able to access available events
- User is able to sign up as a group
- Food bank is able to see request
- Food bank is able to accept/deny request
- User is able to see food bank response

Use case diagram



Dependent (on) use cases:

- Sign up for an event (individual)
- View event

Drop a Shift Use Case:

Title: Drop a Shift

Description: Volunteers must be able to drop a shift in case a conflict arises or they find another shift they would rather sign up for. The process of dropping a shift would require the event to be removed from the specified volunteer's list of registered events, and it would require the database information for that event to be updated to reflect the empty slot. (This is crucial because a previously full shift would now have an open slot available for another volunteer to register.)

Triggers: Volunteer decides they can no longer attend a shift that they signed up for.

Actors: Volunteers

Preconditions: Volunteer is registered for an event.

Main Success Scenario: Volunteer is removed from the specified event/shift, and that shift gains one open "slot" for other volunteers to potentially sign up for.

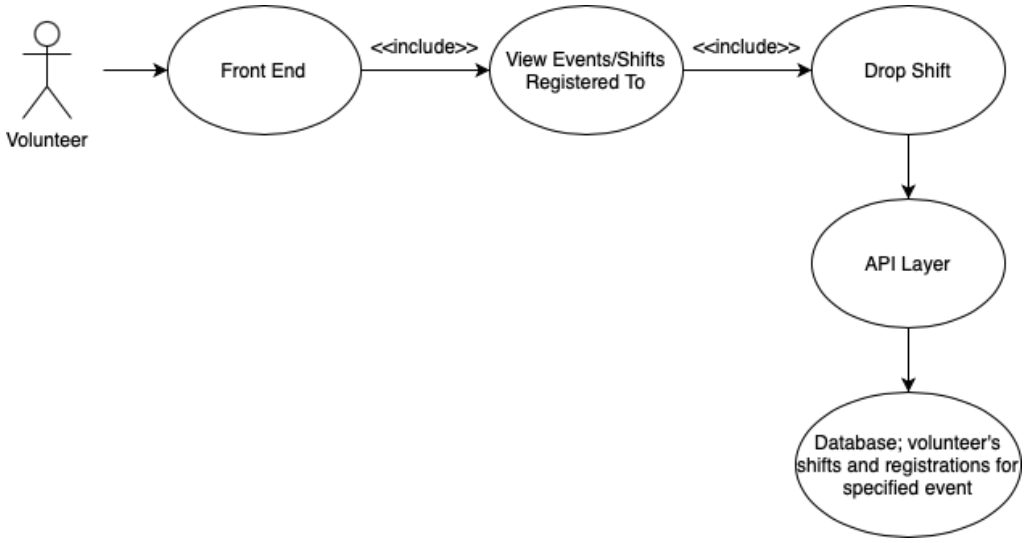
Alternate Success Scenarios: N/A
Failed End Conditions: Volunteer is not removed from the event.
-OR-
Volunteer is removed from the event, but the event information does not reflect the vacant shift.

Extensions: Volunteer coordinator can fill vacant shifts left by dropping a shift (as described in the "Fill Empty Shift" use case).

Steps of Execution:

1. User logs in as a volunteer.
2. Volunteer views events/shifts they are registered for.
3. Volunteer selects shift they no longer want.
4. Volunteer clicks a button to drop the shift. (A "red x" or "trash can")
5. The shift is removed from that volunteer's list of registered events in the database.
6. The volunteer is removed from the list of registered volunteers for that event in the database. The number of available spots for that event/shift in the database is incremented by 1.

Use Case Diagram:



Dependent Use Cases: N/A