TCU & UNTHSC Volunteer Scheduler

Glossary

Version 2.0

Revision History

Date	Version	Description	Author
03/10/2020	1.0	Defined Purposes, Scope, References, Overview.	Maria Amoros
04/10/2020	1.0	Defined CSV Document, Volunteer and Administrator.	Maria Amoros
18/10/2020	1.1	Added terms Canvas, Events and Partners	Maria Amoros
15/04/2021	1.2	Added 11 new terms.	Lydia Pape
04/05/2021	1.2	Added 3 new terms, updated a few old ones, updated Table of Contents	Lydia Pape
04/05/2021	2.0	Added terms of Heroku and Dynos	Maria Amoros

Table of Contents

Glossary

1. Introduction

This Glossary provides important definitions for the TCU & UNTHSC School of Medicine Volunteer Scheduler.

1.1 Purpose

The purpose is to define a definition for words that could have many as well as to provide clarification where ambiguities can occur.

1.2 Scope

The scope for this glossary is to provide definitions to terms associated with the TCU & UNTHSC School of Medicine Volunteer Scheduler project, its deliverable documents, and any important vocabulary that stakeholders need to be aware of.

1.3 References

Canvas interface information -

1.4 Overview

The glossary consists of important terms for the TCU & UNTHSC School of Medicine Volunteer Scheduler project, which includes terms related to technology, academia as well as medicine.

2. Definitions

2.1 Administrator

Person in charge of managing the volunteer scheduling website, especially the creation of events, modifying events' information, cancelling of events, monitoring of user's volunteer hours and creation of reports.

2.2 Canvas

A course management system that supports online learning and teaching in the TCU & UNTHSC School of Medicine where faculty and administrators can also make announcements for targeted students. Early planning of the system in development included considerations of the possibility of interacting with Canvas.

2.3 CSV File (.csv)

A comma separated value file containing a list of data.

2.4 Event

An act of a student, faculty or staff of the TCU & UNTHSC School of Medicine who freely gives time and labour for community service.

2.5 Partner Organization (or just Partner or Organization)

Clinics, shelters or any other institution or organization who offers volunteer opportunities for students, faculty and/or staff to participate in.

2.6 Volunteer

Person who freely chooses to contribute with his time, effort, and expertise to take part in an event or completion of a task, and the definition of a volunteer includes students, faculty and staff from the TCU & UNTHSC School of Medicine. It can also include administrators.

2.7 User

Any person who uses the system, including volunteers and administrators.

2.8 Excel

The MS program currently used by administrators to track student volunteer hours and events. This program will continue to be used for reports of student and event activity; the system will generate reports in the .xlsx file format so that they can be opened in Excel.

2.9 Report

A document containing summarized information and details about volunteer activities as tracked by the system. There are several different types; a report can be on a particular (volunteer) user, a particular event, all users of a certain class (students graduating in the same year), or general information on everything.

2.10 API document

A table of information on the HTTP requests used for communication between the front and back ends of the system. This table is located in the Developer's Manual and contains a row for each type of request, with details on how the front end makes the request and how the back end responds. This is meant to help developers working on the system, particularly for coordination between front and back end developers.

2.11 WAR

WAR stands for Weekly Activity Report. Each member of the developer team is required to log their work on the project each week in this document.

2.12 Test Case

An activity that a developer or user (depending) could perform to gather information on the functionality of a specific part of the system. The dev team will ask the client and earliest users to perform test cases by attempting to use different parts of the system and reporting whether or not they work as expected.

2.13 Front end

The part of the system responsible for presenting each user with a graphical interface to use the system in a browser. The front end must allow a user to log in, present them with appropriate information as queried, and receive information from the user when necessary. The front end passes information both ways between the user and the back end. It is coded mostly in Javascipt with the Vue framework.

2.14 Back end

The part of the system responsible for working with information and performing most logical operations. The back end exchanges information with the front end via HTTP request and response, and also exchanges information with the database using Hibernate. It is coded in Java with the Spring framework.

2.15 Database

The part of the system responsible for storing all information important to the system, including the users (but not their passwords), events, and all waitlist/sign-up/attendance information. It can be queried using

SQL. The back end uses Hibernate SQL to store new information, update existing information, delete information, and query information in the database.

2.16 Waitlist

List of (volunteer) users who want to attend a certain event, but cannot sign up because the event is already at full capacity. The system enforces that users who attempt to sign up for a full event get put on the event's waitlist instead. Waitlists are prioritized in the backend by timestamp. If a user cancels their sign-up for the event, the user on the waitlist with the oldest timestamp is automatically pulled from the waitlist and signed up instead. It is usually split into two lists, between faculty and non-faculty users.

2.17 Sign-up, sign-up list

Sign-up (sometimes written as signup) describes the status of any user who has marked that they plan to attend a certain event. Each event has given capacity fields, limiting how many users can sign up for that event. A user can cancel their sign-up, meaning they no longer plan to attend the event. A sign-up list is a list of users signed up to a particular event. It is usually split into two lists, between faculty and non-faculty users.

2.18 Capacity

The number of users allowed to sign up for a certain event. Each event actually has two capacities: a faculty capacity and a non-faculty capacity. If enough non-faculty are signed up that the event has reached its non-faculty capacity, but the event has not reached its faculty capacity, then a faculty user can still sign up.

2.19 Attendance (or Attendance record)

The occurrence of a (volunteer) user showing up at an event and volunteering for a certain number of hours. Users can log their attendances in the system by checking in and checking out for the event with the appropriate times. After an attendance has been completed (check-in and check-out have both happened), the back end automatically calculates the number of hours and adds that number to the user's total hour count. (An attendance record can be modified afterward by checking in or out again with a different time.) (After an attendance record has been created, the user may no longer cancel their sign-up for that event.)

2.20 Firebase

A third-party service which handles security and authentication of all users in the system. Firebase has its own records of users including emails and passwords. These records are separate from the Database (see 2.15) User records, which include the email and some other information, but not the password. For the purpose of authentication, Firebase records are matched to the corresponding Database records using the common email. See the Software Requirements Specification and Developer Manual for more information on how Firebase works with the system.

2.21 Deployment

The act of making the software "live" on the internet to be accessed by all users. This needs to be done to make changes to the software (updates, bug fixes, etc.) take effect in the final product. It is set up to happen automatically when changes are committed to the main Github repository.

2.22 Github, Github repository

Github is the online version control tool used by the dev team to collaborate and track changes to all parts of the software. A Github repository is a container for a body of software, containing many files, and capable of tracking changes across multiple "branches" with different versions as necessary. The software's frontend code resides in the “Volunteer-App” repository while the backend code, and frontend build, go in the “Server-Volunteer-App” repository. The latter is deployed publicly as the final product.

2.23 Heroku

Heroku is a platform as a service (PaaS) that enables developers to build, run, and operate applications entirely in the cloud.

2.24 Dynos

Dynos are isolated, virtualized Linux containers that are designed to execute code based on a user-specified command. Your app can scale to any specified number of dynos based on its resource demands.