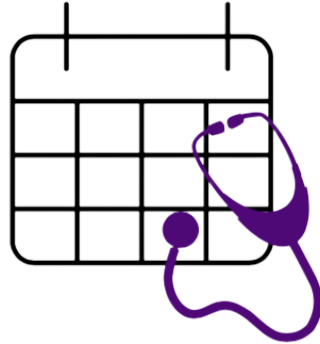

TCU & UNTHSC Volunteer Scheduler



Use Cases
Version 2.0

Revision History

Date	Version	Description	Author
09/30/2020	1.0	Defined Use Cases 1, 2 & 12	Maria Amoros
10/2/2020	1.0	Defined Use Cases 8 & 9	Peyton Freeman
10/2/2020	1.0	Defined Use Cases 6 & 7	Lydia Pape
10/3/2020	1.0	Defined Use Cases 3, 10 & 11	Jeshua Suarez-Lugo
10/3/2020	1.0	Defined Use Cases 4 & 5	Emerson Wolf
10/18/2020	1.1	Revised this document and fixed mistakes.	Maria Amoros
05/04/2021	2.0	Added a new use case on create reports for events.	Maria Amoros

Table of Contents

Use Case 1: Logs into account	4-5
Use Case 2: Changes account password	6-7
Use Case 3: Creates an Event	8-9
Use Case 4: Generates Report for One User	10-11
Use Case 5: Generates Report based on Event	12-13
Use Case 6: Browses for Events in Calendar	14-15
Use Case 7: Signs-up for Event	16-17
Use Case 8: Checks-in for an Event	18-19
Use Case 9: Checks-out For an Event	20-21
Use Case 10: Modifies an Event	22-23
Use Case 11: Cancels an Event	24-25
Use Case 12: Logs-Out	26-27
Use Case 13: Checks attendance for an event	28-29
Use Case 14: Checks for sign-up and waitlist for an event	30
Use Case 15: Adds a user	31
Use Case 16: Deletes a user	32

Use Case 1: Logs into account

Brief Description:

Students, Faculty and Administrators need to log into the TCU & UNTHSC School of Medicine Volunteer Scheduling System in order to use it, so they need to provide their username and password to the system when prompted.

Primary Actor:

- Students
- Faculty
- Administrators
- System

Level:

Sub function

Stakeholders and Interests:

- Students
- Faculty
- Administrators
- System
- TCU & UNTHSC School of Medicine

Preconditions:

Students, Faculty and Administrators already count with an account when logging into the system.

Postconditions:

- The user is able to use the system.
- The system shows functionalities to the user based on their privileges.
- If the user no longer wants to log-in, he/she will not be logged-in to the system.

Trigger:

The user wants to use the system.

Main Success Scenario:

1. The user is prompted by the System to enter username and password.
2. The user enters username and password.
3. The user validates the already entered username and password.
4. The system validates the username and password against its database.
5. The system retrieves the user's role and logs him/her into the system.

Extensions:

- 3a. User realizes he/she forgot log-in credentials.
 - 3a.1. User informs the system he/she forgot his/her password.

- 3a.2. The system prompts the user to indicate his/her TCU Email.
 - 3a.3. The user enters TCU email.
 - 3a.4. The system validates email exists and sends instructions to it on how to recover the account.
 - 4a.5. The user follows instructions and sets a new password.
 - 4a.6. The user wants to try to log-in again.
 - 4a.7. The use case returns to step 1 and continues.
- 4a. The user realizes he had a typo in his username and/or password.
- a 4a.1. The system indicates that there was an error and prompts the user to enter different user/password combination.
 - 4a.2. The use case returns to step 2 and continues.
- 4b. The System is down and cannot process any request.
- 4b.1. The system informs the user it is unable to process its request.
 - 4b.2. The use case ends.
- 4c. A non-system user without an account is trying to log-in, so its credentials cannot be validated.
- 4d.1. The use case ends.

Priority:

High

Secondary Actors:

None

Special Requirements:

- Password protection
- Usability

Open Issues:

None

Use Case 2: Changes account password

Brief Description:

Students, Faculty and Administrators want to change their account passwords.

Primary Actor:

- Students
- Faculty
- Administrators
- System

Level:

User level

Stakeholders and Interests:

- Students
- Faculty
- Administrators
- System
- TCU & UNTHSC School of Medicine

Preconditions:

User is already logged-in to the system.

Postconditions:

- User successfully changes the account password.
- User does not change password.

Trigger:

User wishes to change his/her account password.

Main Success Scenario:

1. User indicates he/she wants to change the password associated with his/her account.
2. The system prompts the user to indicate his/her current password.
3. The user enters the current password and submits it.
4. The system validates the current password.
5. System prompts the user to indicate the desired new password.
6. The user enters the new password he/she desires.
7. The system prompts the user to re-type the password.
8. The user re-types the desired new password.
9. The system validates that the new password and the re-type of it match.
10. The system stores the new password for this user.
11. The system indicates the user that he/she successfully changed his/her password.

Extensions:

- 1a. The user decides he/she does not want to change password.
 - 1a.1. The user indicates that he no longer wants to change his/her password.
 - 1a.2. The use case ends.
- 4a. The password entered by the user as his current password is incorrect.
 - 4a.1. The system lets the user know that the entered password is incorrect.
 - 4a.2. The use case returns to step # 3 and continues.
- 9a. The new password and its re-type does not match.
 - 9a.1. The system lets the user know that the two do not match.
 - 9a.2. The use case returns to step #5 and continues.

Priority:

Medium

Secondary Actors:

None.

Special Requirements:

- Password protection
- Usability

Open Issues:

None

Use Case 3: Creates an Event

Brief Description:

Admin wants to add a new event for Students/Faculty to sign up for.

Primary Actor:

- Admin
- System

Level:

Admin Level

Stakeholders and Interests:

- Students
- Faculty
- Partners
- System
- TCU School of Medicine

Preconditions:

Admin should be logged-in to the system.

Postconditions:

Events are created for Students/Faculty that meet criteria to sign up for.

Trigger:

Admin wants to create a new event.

Main Success Scenario:

1. Admin indicates they want to create a new Event
2. Admin indicates event details
3. System prompts Admin for Students groups the event will be visible to/permitted to sign up
4. Admin selects User groups permitted to sign up
5. System prompts Admin for Special Criteria (such as specific course completions)
6. Admin may include special criteria in the form of text
7. System displays New Event preview for final approval
8. Admin confirms Event
9. System adds new Event to database and displays Event for sign up to intended groups

Extensions:

8a. Admin wants to edit details

8a. 1. Admin selects "Edit Details" button

8a2. System "New Event" window keeps previously entered data, returns to Step

3 in Main Success Scenario.

Priority:

High

Secondary Actors:

None

Special Requirements:

- System has access to User Groups for intended users to have event visible

Open Issues:

None

Use Case 4: Generates Report for One User

Brief Description:

Admin wants to view the data on one user's volunteer work.

Primary Actor:

- Admin

Level:

Admin Level

Stakeholders and Interests:

- Students
- Faculty
- Partners
- System
- TCU School of Medicine

Preconditions:

Admin should be logged-in to the system.

Postconditions:

Admin view a file containing all of the relevant data for one user

Trigger:

Either

1. Admin generates a user's report off of a user's profile
2. Admin generates a user's report and then searches for the student

Main Success Scenario:

1. Admin views a user's profile
2. Admin generate a user's report
3. System gathers relevant data for the user
4. System condenses data into a readable, formatted report file
5. File is opened for admin

Extensions:

5a. Admin is allowed to download files.

Priority:

High

Secondary Actors:

None

Special Requirements:
None

Open Issues:
None

Use Case 5: Generates Report for an Event

Brief Description:

Admin wants to view the data on all users' volunteer work

Primary Actor:

- Admin

Level:

Admin Level

Stakeholders and Interests:

- Students
- Faculty
- Partners
- System
- TCU School of Medicine

Preconditions:

Admin should be logged-in to the system.

Postconditions:

Admin views a file containing the data of volunteer work for all users who attended an event.

Trigger:

Admin wants to create a report for an event.

Main Success Scenario:

1. Admin queries the system for a report of all volunteer users
2. System asks for specifics on which class(es) Admin wants to see data on.
3. Admin selects class (and potentially other) parameters to generate reports.
4. System collects relevant data.
5. System condenses data into a readable, formatted report file
6. File is opened for admin in a new window

Extensions:

- 6a. Admin is allowed to download files.

Priority:

Low

Secondary Actors:

None

Special Requirements:

None

Open Issues:

How to sort the data generated by the system

Use Case 6: Browses for Events in Calendar

Brief Description:

Student and Faculty Volunteers want to browse for events that they are interested in and that they can work into their schedules. They want to know about all the options available to them before committing to any certain event.

Primary Actor:

- Students
- Faculty volunteers

Level:

User level

Stakeholders and Interests:

- Students
- Faculty volunteers
- TCU & UNTHSC School of Medicine

Preconditions:

User is already logged into the system.

Postconditions:

User can see a list of available events organized by date.

Trigger:

User decides to browse for events.

Main Success Scenario:

1. User requests the calendar of available events either in week, month or day format.
2. The system fetches a list of events and their related information from the database.
3. The system displays the names, times, and general requirements of events to the user, organized by date and time in chronological order.
4. User views and possibly selects events at their leisure.

Extensions:

- 3a. The system received an empty list from the database.
 - 3a.1. The system informs the user that no events are currently available.
 - 3a.2. The use case ends.

Priority:

High

Secondary Actors:

None

Special Requirements:

The system has access to a database containing the events available for sign-up.

Open Issues:

None

Use Case 7: Signs-up for Event

Brief Description:

Student and Faculty Volunteers need to sign up for events to fill their required hours.

Primary Actor:

- Students
- Faculty volunteers

Level:

User level

Stakeholders and Interests:

- Students
- Faculty volunteers
- Partner Organizations
- System
- TCU & UNTHSC School of Medicine

Preconditions:

Users are already logged into the system and browsing events.

Postconditions:

- User successfully signs up for an event.
- Users do not sign up for an event.

Trigger:

User decides to sign up for a specific event.

Main Success Scenario:

1. User selects their chosen event.
2. The system displays the name and details of the event for the user to review.
3. User confirms that they want to sign up for this event.
4. The system consults the database and verifies that this user has not already signed up for an event with a conflicting scheduled time.
5. The system records the user's new commitment in the database.
6. The system confirms to the user that the sign-up process is complete.

Extensions:

- 3a. User chooses not to sign up for this event.
 - 3a.1. Users indicate that they will not sign up for the selected event.
 - 3a.2. The system ceases displaying the event's details and displays again the list of available events to the user.
 - 3a.3. The use case ends.
 - 4a. The system detects a time conflict with the user's chosen event.
 - 4a.1. The system informs the user of the conflict, and requires them to

cancel signing up for this new event.

4a.2. User accepts the failure.

4a.3. The system ceases showing the details of the chosen event and again displays the list of available events.

4a.4. The use case ends.

Priority:

High

Secondary Actors:

None

Special Requirements:

- Use case 7 has been implemented.
- The system has access to a database containing the events each user has already signed up for.

Open Issues:

None

Use Case 8: Checks-in for an Event

Brief Description:

Students need to check-in to events that they are attending.

Primary Actor:

- Students
- System

Level:

User Level

Stakeholders and Interests:

- Students
- System
- Faculty
- Administrators
- TCU & UNTHSC School of Medicine

Preconditions:

- An administrator has created the event
- The student has signed up for the event

Postconditions:

- The system registers the time that the user checked in

Trigger:

The user is attending an Event.

Main Success Scenario:

1. The user indicates that they have arrived at the scheduled event.
2. The system prompts the user to check in to the event.
3. The user accepts the prompt from the system.
4. The system records and stores the time of check-in.
5. The system informs the user that they have successfully checked in.

Extensions:

1a. The user has not been scheduled for the event:

The user is not scheduled for the event he/she is attempting to check in at.

1a2. The use case ends.

3a. The user does not want to check in.

3a1. The user indicates they no longer want to check in.

3a2. The system removes the prompt to check in.

3a3. The use case ends.

Priority:
High

Secondary Actors:
None

Special Requirements:
Connection to database in which system records time.

Open Issues:
None

Use Case 9: Checks-out For an Event

Primary Actor:

- Students
- System

Level:

User Level

Stakeholders and Interests:

- Students
- System
- Faculty
- Administrators
- TCU & UNTHSC School of Medicine

Preconditions:

- A user has checked in to a scheduled event.

Postconditions:

- The system registers the time that the user checked out
- The user is informed that they have successfully checked out

Trigger:

The user wants to leave an event.

Main Success Scenario:

1. The user indicates that they want to leave an event they have checked in to.
2. The system prompts the user to check out of the event.
3. The user accepts the prompt from the system.
4. The system records and stores the time of check-out.
5. The system informs the user that they have successfully checked out.

Extensions:

- 1a. The user has not checked in to the event:
 - . The user is not checked in to the event they are attempting to check out of.
- 1a2. The use case ends.
- 3a. The user does not want to check out:
 - 3a1. The user indicates they no longer want to check out.
 - 3a2. The system removes the prompt to check out.
 - 3a3. Use case ends.

Priority:

High

Secondary Actors:

None

Special Requirements:

Connection to database in which system records time.

Open Issues:

None

Use Case 10: Modifies an Event

Brief Description:

Admin wants to modify the Event details of selected Event

Primary Actor:

- Admin

Level:

User goal level

Stakeholders and Interests:

- Student
- Faculty
- Partner

Preconditions:

- Event has already been created and posted for sign ups
- Admin should be logged-in to the system.

Postconditions:

- Event details reflect new details
- Students previously signed up for event are notified of change

Trigger:

Admin indicates that he/she wishes to modify an event.

Main Success Scenario:

1. System pops up “Event Details” window
2. Admin enters new details for selected event
3. System prompts Admin for Students groups the event will be visible to/permitted to sign up
4. Admin selects User groups permitted to sign up
5. System prompts Admin for Special Criteria (such as specific course completions).
6. Admin may include special criteria in the form of text
7. System displays Event preview with modified details for final approval
8. Admin confirms Event
9. System sends modified event details to database
10. System notifies previously signed up Users of changes

Extensions:

- 1a. Event has already occurred
 - 1a. 1. System alerts Admin that Event has already occurred
 - 1a. 2. Admin confirms alert

- 1a. 3. System exits “Event Details” window
- 8a. The event's start time is in the pas
 - 8a. 1. System alerts Admin that Event is In Progress
 - 8a. 2. Admin confirms alert
 - 8a. 3. System exits “Event Details” window

Priority:

Medium

Secondary Actors:

None

Special Requirements:

Connection to Database for modification of Event listing

Open Issues:

- Current conceptual implementation may be limited in scope to future implementation

Use Case 11: Cancels an Event

Brief Description:

Admin cancels a previously scheduled Event

Primary Actor:

- Admin

Level:

User goal level

Stakeholders and Interests:

- Students
- Faculty
- Partners

Preconditions:

- Event exists in database
- Event has not already occurred
- Admin shall be logged-in
- Admin selects event to be modified

Postconditions:

- Event is removed from listings, database

Trigger:

Admin selects "Cancel Event" button

Main Success Scenario:

1. System prompts Admin to confirm cancelation of Event
2. Admin confirms cancelation with "Cancel Event" button
3. System removes event from listings, notifies affected (signed up) users of cancellation

Extensions:

3a. Event is in progress

3a. 1. System notifies Admin that event is in progress

3a. 2. Admin confirms notification

3a. 3. System exits

3b. Event has already occurred

3b. 1. System notifies Admin that event has already occurred

3b. 2. Admin confirmed notification

3b. 3. System exits

Priority:

High

Secondary Actors:

None

Special Requirements:

None

Open Issues:

None

Use Case 12: Logs-Out

Brief Description:

Students, Faculty and Administrators want to log-out from their accounts.

Primary Actor:

- Students
- Faculty
- Administrators
- System

Level:

User level

Stakeholders and Interests:

- Students
- Faculty
- Administrators
- System
- TCU & UNTHSC School of Medicine

Preconditions:

User is already logged-in to the system.

Postconditions:

- The user logs-out from his/her account.
- If the user changes his/her mind, he/she does not log-out from his/her account.

Trigger:

The user wants to log-out from his/her account.

Main Success Scenario:

1. The user indicates he/she wants to log-out from the account.
2. The system prompts the user to confirm whether he/she wants to log-out.
3. The user confirms he/she wants to log-out from the account.
4. The system successfully logs the user out.

Extensions:

- 3a. The user decides he/she no longer wishes to log out from the account.
 - 3a.1. The use case ends.

Priority:

Medium

Secondary Actors:

None.

Special Requirements:

None

Open Issues:

None

Use Case 11: Checks Attendance for an Event

Brief Description:

Admin checks an attendance for an event.

Primary Actor:

- Admin

Level:

User goal level

Stakeholders and Interests:

- Students
- Faculty
- Partners

Preconditions:

- Event exists in database
- Event already occurred
- Admin shall be logged-in
- Admin selects event to be checked

Postconditions:

- Admin checks for a the attendance of a past event

Trigger:

Admin selects "Attendance" button

Main Success Scenario:

1. Admin can see the list of users who attended a past event

Extensions:

- 1b. Event has not occurred
 - 1a. 1. Attendance will be zero.

Priority:

Medium

Secondary Actors:

None

Special Requirements:

None

Open Issues:
None

Use Case 12: Checks for sign-up and waitlist for an event

Brief Description:

Admin checks who has signed up and who is in the waitlist for an event

Primary Actor:

- Admin

Level:

User goal level

Stakeholders and Interests:

- Students
- Faculty
- Partners

Preconditions:

- Event exists in database
- Admin shall be logged-in
- Admin selects event to be checked

Postconditions:

- Admin checks who has signed up and who is in the waitlist for an event

Trigger:

Admin selects “Sign-up” button

Main Success Scenario:

- Admin can see the list of users who has signed up and who is in the waitlist for an event

Priority:

Medium

Secondary Actors:

None

Special Requirements:

None

Open Issues:

None

Use Case 13: Adds a user

Brief Description:

Admin wants to add a user

Primary Actor:

- Admin

Level:

User goal level

Stakeholders and Interests:

- Students
- Faculty
- Partners

Preconditions:

- Admin shall be logged-in
- Admin prepares excel file with user's information

Postconditions:

- User is now in the database.

Trigger:

Admin adds a file with the user's information and selects "Add"

Main Success Scenario:

- New user is now in the database.

Priority:

Medium

Secondary Actors:

None

Special Requirements:

None

Open Issues:

None

Use Case 14: Deletes a user

Brief Description:

Admin deletes a new user

Primary Actor:

- Admin

Level:

User goal level

Stakeholders and Interests:

- Students
- Faculty
- Partners

Preconditions:

- Admin shall be logged-in
- Admin prepares excel file with user's information

Postconditions:

- User is deleted from the database

Trigger:

Admin adds a file with the user's information and selects "Delete"

Main Success Scenario:

- New user is now in the database.

Priority:

Medium

Secondary Actors:

None

Special Requirements:

None

Open Issues:

None