**Template:** send message

|  |  |
| --- | --- |
| **Title:** | **send message** |
| **Description:** | **A way for the Actors within the system to communicate with each other, based on the client demands.** |
| **Actors:** | Volunteer, school manager, regional manager, Admin |
| **Preconditions:** | The Actors are logged into the system. |
| **Postconditions:** | The receiving actor has a message writing for him in the message system window.  He can access the contents of his message via pressing on the message alert from the app or opening the app and navigating to the message system window, where the message will appear. |
| **Basic flow:** | An actor clicks on the "send message button" leading him to the messaging system window.  In this window he must fill 2 boxes: the destination of his message (a person's name from a list) and the content of his message.  when these 2 requirements are met, and the Actor presses the "send" button, the system will sent an alert to the designated Actor, notifying him there is a message waiting for him in the system. |
| **Alternative Flows:** | 1) The Actor sending the message may attach a file to the message  2) The Actor receiving the message may reply to a message.  The reply will contain a tag mentioning its a reply and including the replied message contents. |
| **Exception Flows:** | The sending Actor doesn’t fill one of the 2 main boxes and hits send.  The system will pop an error message notifying him that these fields must be filled. |

**Template:** show activities

|  |  |
| --- | --- |
| **Title:** | **show activities** |
| **Description:** | An all-purpose main feed, containing posts on special events or ordinary day to day updates regarding a volunteer's place of volunteering. |
| **Actors:** | Volunteer, school manager, regional manager, Admin |
| **Preconditions:** | The Actors are logged into the system and the volunteer is following after his volunteer place. |
| **Postconditions:** | for 1: the post is added to the database and other users can see and interact with it.  for 4: all the responses are collected and shown to the post owner and admin.  for 5: the event is added to the user's calendar |
| **Basic flow:** | An actor selects the show activity tab, the system loads his connected posts to a feed of post's that he can scrolls through. |
| **Alternative Flows:** | 1) A manager who opened the activity tab, pressed the "add a new post" button. a new window pops up that lets him fill the contents of his posts and its properties (selecting post's type, input fields, images, documents...)  the post is added to the database and every actor with that follows this post's author will see it in their post feed.  2) a manager that uploaded a post selects the edit button on his post. the system opens the fill post contents tab with the fields already filled with the current values of the post and lets him edit it.  3) the admin opens the activity tab. it will choose a region of which he wants to see its feed. this is so the admin's feed won't be overwhelmed and unorganized. |
| **Exception Flows:** | the Actor doesn’t have internet connection when requesting the feed. the system will alert him of this error and the feed will be empty.  4) an actor chooses to fill an input field for a post. precondition for this flow: the post was not uploaded from this user.  5) an actor clicks the button to add to calendar.  the system captures the input and formats it with all the replies to this input field. |

**Template:** Volunteers sign-up

|  |  |
| --- | --- |
| **Title:** | **Volunteers sign-up** |
| **Description:** | volunteers sign up to the system and are associated with a school manager and a volunteering place. |
| **Actors:** | volunteer, school manager |
| **Preconditions:** | The school manager is signed up to the system and there are volunteer places in the database. |
| **Postconditions:** | his user is signed up to the system, associated with at least one volunteer place and a school manager to see events on the main feed.  a notification is sent to the school manager about a new user to verify it is actually a volunteer and erase him as a volunteer user if he is an intruder (or entered the wrong school code). the system will notify the intruder if his disassociated with the school manager. |
| **Basic flow:** | A user enters the system and is welcomed with the opening login screen. he presses the sign-up button, and the system opens a secondary screen for sign up.  in this screen, he enters his name, a password and a pre-generated code of his school manager. the system associates the volunteer with the school manager he entered the code for and opens the edit volunteer place window.  in the enter volunteer place window the volunteer must choose at least one volunteer place to be associated. after completing this task, he is rerouted to the main screen after login. |
| **Alternative Flows:** | There are no alternative flows. |
| **Exception Flows:** | 1) the user enter's an invalid school manager code, or a weak password. the system notifies him about this error and asks to fix his input.  2) the user doesn’t select a volunteer place. the system notifies him about this error and asks to choose again. |