

What if the recipient need change? he/she would like another help different from the first one or after receiving the first one? Changes are (currently) done by contacting an administrator by email or phone. Another (future) way could be to provide the recipient with a personal link and password/PIN to make changes.

|  |  |  |
| --- | --- | --- |
| **Use case name**:Fill a recipient’s form | **ID**: UC-1 | **Priority**: Medium |
| **Actor**: Potential recipient | | |
| **Description**: A potential recipient solicits help by filling out the recipient’s form available at any Elsk’s city website | | |
| **Trigger**: The potential recipient decides to become a recipient  **Type**:   * External * Temporal | | |
| **Preconditions**:  The recipient’s form is available at the local website  The potential recipient is not yet registered as a recipient of ***that*** Elsk’s city | | |
| **Basic flow:**  The person enters the URL of an Elsk city website in the browser  The person goes to the recipient registration section  The person fills out the form and submits it  The person receives a confirmation email | | |

|  |  |  |
| --- | --- | --- |
| **Use case name**: Fill a volunteer’s form | **ID**: UC-2 | **Priority**: Medium |
| **Actor**: Potential volunteer/helper | | |
| **Description**: A potential volunteer fills out the volunteer’s form available at any Elsk’s city website and submits it. | | |
| **Trigger**: The potential volunteer decides to become a volunteer/helper  **Type**:   * External * Temporal | | |
| **Preconditions**:  The volunteer’s form is available at the local website  The potential volunteer helper is not yet registered as a volunteer of ***that*** Elsk’s city | | |
| **Basic flow:**  The person enters the Url of an Elsk city website in the browser  The person goes to the volunteer registration section  The person fills out the form and submits it | | |

|  |  |  |
| --- | --- | --- |
| **Use case name**:Register a recipient to an Elsk’s city | **ID**: UC-3 | **Priority**: High |
| **Actor**: Elsk’s city website | | |
| **Description**: A recipient registration request is sent from an Elsk’s city website. The system satisfied the request. | | |
| **Trigger**: A potential recipient visits an Elsk, fills out the recipient form and submits it.  **Type**:   * External * Temporal | | |
| **Preconditions**:  The recipient’s form is correctly filled out and submitted from Elsk’s city website.  The person is a new recipient for that city.  The form is correctly filled. | | |
| **Basic flow:**  **1.0 register a recipient to Elsk’s city**   1. Elsk city website sends recipient registration request via *the use case UC-1:Fill a recipient’s form* 2. The system identifies the city making the request 3. The system checks that the data is valid 4. The system confirms the register request 5. The system notifies the Elsk’s city website that the registration is accepted and completed using a thank-you URL. 6. The system sends to the new recipient a confirmation email   *Invalid data*:   * missing form mandatory field * username/email already exists in the system * information entered in form field does comply with the require format. E.g malformed email. | | |
| **Alternative flow:**  **A.1 The system cannot identify the city** [branch 2]   1. The request is dismissed.   **A.2 The system cannot validate the data** [branch 3]   1. The system describes which data entered was invalid and sends it together with suggestions for entering valid data to the Elsk’s city website 2. The Elsk’s city website prompt the potential recipient to re-enter the invalid data for submission 3. Repeat the steps of the Normal flow until the data is valid. | | |

|  |  |  |
| --- | --- | --- |
| **Use case name**: Register a volunteer to an Elsk’s city | **ID**: UC-4 | **Priority**: High |
| **Actor**: Elsk’s city website | | |
| **Description**: A volunteer registration request is sent from an Elsk’s city website. The system satisfied the request. | | |
| **Trigger**: A potential volunteer visits an Elsk, fills out the helper form and submits it.  **Type**:   * External * Temporal | | |
| **Preconditions**:  The volunteer’s form is correctly filled out and submitted from Elsk’s city website.  The person is a new volunteer for that city.  The form is correctly filled. | | |
| **Basic flow:**  **1.0 register a volunteer to Elsk’s city**   1. Elsk city website sends volunteer registration request via *the use case UC-2:Fill a volunteer’s form* 2. The system identifies the city making the request 3. The system checks that the data is valid 4. The system accepts the registration request 5. The system sends a notification of successful registration to Elsk’s city website through a thank-you URL. 6. The system sends to the newly registered volunteer a confirmation email. | | |
| **Alternative flow:**  **A.1 The system cannot identify the city** [branch 2]   1. The request is dismissed.   **A.2 The system cannot validate the data** [branch 3]   1. The system returns an error and describes which data entered was invalid and sends it to the Elsk’s city website, together with suggestions for entering valid data The Elsk’s city website prompt the potential recipient to re-enter the invalid data for submission 2. Repeat the steps of the Normal flow until the data is valid. | | |

|  |  |  |
| --- | --- | --- |
| **Use case name**: Generate a city schedule for a Help-Event(should we say event instead of period of time? For extension, an event can be calendar based i.e defined within a period of time, may have a location, etc.) | **ID**: UC-5 | **Priority**: High |
| **Actors**: Local Administrator, Global Administrator | | |
| **Description**: This use case describes how an administrator(local/global) produces the schedule for city for a Help-Event. | | |
| **Trigger**: the administrator decides to generate the schedule  **Type**:   * External * Temporal | | |
| **Preconditions**:  There are registered recipients for that city  There are registered volunteers for that city  All the recipients are assigned to category(Green, Yellow or Red)  All the volunteers are assigned to category to which they can provide help(Green, Yellow or Red)  The kind of help wanted or provided respectively by a recipient or a volunteer, is identified so that a match can be possible or not.  The “help time frame” of city for which the schedule has to be generated, is created. (see UC-13) | | |
| **Basic flow:**  **1.0 generate schedule for a help-event**   1. The system asks the administrator to set Date of the help-event 2. The system asks the administrator to specify the time-period 3. The system verifies that all the recipients have been assigned a category 4. The system verifies that all the volunteers have been assigned a category 5. The system checks that all the recipients and volunteers have been assigned to an item of the listOfHelpTypes. Under the new system, this list can be pre-established and updated as the need arises. 6. The system generates the schedule for the concerned help-event based on three attributes namely, category, the availability, and the kind of help provided or sought | | |
| **Alternative flow:**  **A.1 The system found that there is at one recipient with no category** [branch 3]   1. A recipient with no category, is assigned one via *use case UC-6: Assign a city recipient to category*. 2. Repeat step 1 for any other recipient without a category   **A.2 The system found that there is at one volunteer with no category** [branch 4]   1. A volunteer with no category, is assigned one via *use case UC-7 Assign a city volunteer to category*. 2. Repeat step 1 for any other volunteer without a category | | |

|  |  |  |
| --- | --- | --- |
| **Use case name**: Assign a city recipient to category | **ID**: UC-6 | **Priority**: High |
| **Actor**: Local Administrator, Global Administrator | | |
| **Description**: this use case describes how a local/global administrator assigns a recipient of a city to category | | |
| **Trigger**: the administrator decides to generate the schedule or the administrator just decides this assignment to happen  **Type**:   * External * Temporal | | |
| **Preconditions**:  The recipient has registered for that city | | |
| **Basic flow:**  **The steps to assigned a recipient a category** | | |
| **Alternative flow:** | | |

|  |  |  |
| --- | --- | --- |
| **Use case name**: Assign a city volunteer to category | **ID**: UC-7 | **Priority**: High |
| **Actor**: Local Administrator, Global Administrator | | |
| **Description**: this use case describes how a local/global administrator assigns a volunteer of a city to category | | |
| **Trigger**: the administrator decides to generate the schedule or he/she just decide this assignment to happen after the registration  **Type**:   * External * Temporal | | |
| **Preconditions**:  The volunteer has registered for that city | | |
| **Basic flow:**  **The steps to assigned a volunteer/helper a category** | | |
| **Alternative flow:** | | |

|  |  |  |
| --- | --- | --- |
| **Use case name**: Identify the kind of help provided or solicited by a participant (R/V) | **ID**: UC-8 | **Priority**: High |
| **Actor**: Local Administrator, Global Administrator | | |
| **Description**: this use case describes how a local/global administrator identifies the kind of help provided or solicited by a participant (R/V). This may eventually lead to the creation or the extension of listOfHelpTypes. | | |
| **Trigger**: the administrator decides to generate the schedule or he/she just decide this identification to happen after the registration  **Type**:   * External * Temporal | | |
| **Preconditions**:  The volunteer/recipient has registered for a city  They did not choose one of the predefined option from the listOfHelpTypes (May be an the other option with description), probably the “other” option followed by a “description” from the registration form. | | |
| **Basic flow:**  **The steps to find out which volunteers can help a given participant.**   * **In the actual system, this is done by reading what is inputted in corresponding form field by both the recipient and volunteer. The administrator can then decide what matches are possibles based on that particular attributes**   1.0 The administrator read the description provided by the recipient/volunteer   1. He/she may contact the person for more information (phone call or email) if necessary 2. The administrator finds that the request can be satisfied through exiting option and decides to assign him/her an existing option of the listOfHelpTypes | | |
| **Alternative flow:**  **A.1 The administrator cannot find an existing option that corresponds to the request of the recipient/volunteer** [branch 2]   1. The administrator decide to create a new option to the listOfHelpTypes.   **A.2** The administrator cancels the operation if recipient / volunteer doesn’t respond or suspends it until they respond. [branch 1] | | |

|  |  |  |
| --- | --- | --- |
| **Use case name**: Send email to a participant | **ID**: UC-9 | **Priority**: High |
| **Actor**: Local Administrator, Global Administrator | | |
| **Description**: this use case describes how a local/global administrator sends an email to a participant. A participant is either a recipient or a volunteer. | | |
| **Trigger**: the administrator want to contact a participant to either confirm, follow-up or evaluate the “city love week”.  **Type**:   * External * Temporal (The evaluation period is due) | | |
| **Preconditions**:  The participant (R/V) has registered for that city or has visited / has been visited | | |
| **Basic flow:**  **1.0 This use case starts with the local administrator accesses the system feature to view all the participants**   1. The System displays the list of participants 2. The local administrator identifies which participant he/she wants to email and request the System to send an email. 3. The System displays that participant account information and prompts the administrator to enter the message. 4. The Administrator enters the desired information and confirms its request. (Alternative flow 1.1) 5. The System validates the information, store the information in the participant account and sends an email to the participant. It then notifies the administrator that the email has been sent. 6. The use case ends | | |
| **Alternative flow:**  **1.1 Add the participant to an email existing campaign. The administration subscribes the participant to campaign rather than sending an email directly. [branch 4]**   1. (*See UC-10:* Subscribe a participant to a campaign). | | |

|  |  |  |
| --- | --- | --- |
| **Use case name**: Subscribe a participant to a campaign | **ID**: UC-10 | **Priority**: High |
| **Actor**: Local Administrator/Global Administrator, MailChimp | | |
| **Description**: this use case describes how a local/global administrator subscribe a participant to campaign available in MailChimp | | |
| **Trigger**: the administrator want to email a participant to either confirm, follow-up or evaluate the “city love week”.  **Type**:   * External * Temporal (The evaluation period is due) | | |
| **Preconditions**:  The participant (R/V) has registered for that city or has visited / has been visited | | |
| **Basic flow:**  **1.0 The administration subscribes the user to campaign (from Mailchimp) rather than sending an email directly.**   1. The Administrator specifies a campaign list and requests the system to add the participant to that list. 2. The System validates the request and notifies the Administrator that the participant has been added. 3. The use ends | | |

|  |  |  |
| --- | --- | --- |
| **Use case name**: Manage Local Administrator | **ID**: UC-11 | **Priority**: High |
| **Actor**: Global Administrator | | |
| **Description**: this use case describes how the global administrator create, update or delete a local administrator from the system. | | |
| **Trigger**: A new local administrator need to be created/appointed, or existing one need to leave or change need to done in an existing administrator profile like reassignment to another city.  **Type**:   * External * Temporal | | |
| **Preconditions**:  The Global Administrator must be logged in | | |
| **Basic flow:**  **1.0 Create Local Administrator Account. This use case starts with the Global administrator accesses the system feature to manage local administrator accounts**   1. The System displays the local administrator’s accounts currently stored in the system. 2. The *Global Administrator* selects the desired local administrator’s account to update (Alternative flow 1.1) 3. The System displays the selected administrator’s account information currently stored in the system. 4. The *Global Administrator* enters the desired local administrator’s account information values and requests that the system saves the entered values. 5. The System validates and store the values in the local administrator’s account. It then notifies the *Global Administrator* that the local administrator’s account has been updated. 6. The use case ends | | |
| **Alternative flow:**  **1.1 New Local Administrator Account Creation. The Global Administrator requests to create a new account** [branch 2]   1. The System prompts the *Global Administrator* to fill the form to create a new local administrator’s account information. 2. The *Global Administrator* fills out the form and submits it. 3. The System validates the data and saves into a new administration account. 4. The System then notifies *Global Administrator* that a new local administrator account has been created and sends an email the new local administrator to creates the login details (*See UC-17: Create login details*). 5. The use ends | | |

|  |  |  |
| --- | --- | --- |
| **Use case name**: Manage help time frame | **ID**: UC-12 | **Priority**: High |
| **Actor**: Local Administrator, Global Administrator | | |
| **Description**: this use case describes how a local/global administrator manages help time frame for a city. It allows the update of the daily time period within which visits must be done. | | |
| **Trigger**: the local administrator want to schedule *“Love week” event.*  **Type**:   * External * Temporal | | |
| **Preconditions**:  The Administrator is signed in. | | |
| **Basic flow:**  **1.0 This use case starts with the local administrator accesses the system feature to manage the daily time period**   1. The System displays the visit time period for each of week currently stored in the system. 2. The *Administrator* enters the desired time values and requests the System to save the changes 3. The System validates and saves the values. It then notifies the Administrator that the help visit time period has been updated. 4. The use case ends | | |
| **Alternative flow:**  **1.1 The Administrator cancels request**   1. At any time, the Administrator may choose to cancel the help time period update. At which point, the processing is discontinued, the help time period remains unchanged, and the administrator is notified that the help time period management request has been cancelled. | | |

|  |  |  |
| --- | --- | --- |
| **Use case name**: Create help time frame | **ID**: UC-13 | **Priority**: High |
| **Actor**: Local Administrator, Global Administrator | | |
| **Description**: this use case describes how a local/global administrator creates help time frame for a city in which visit must be done. Example includes every second friday of a month, a whole week or few day in particular month of the year, etc. This help time frame varies depending of city and has to be created before the schedule can be generated. | | |
| **Trigger**: the local administrator want to schedule *“Love week” event.*  **Type**:   * External * Temporal | | |
| **Preconditions**:  The Administrator is signed in. | | |
| **Basic flow:**  **1.0 This use case starts with the local administrator accesses the system feature to specify the daily time period within which the activities will take place**   1. The System prompts the Administrator to set the times allowed for starting and ending respectively the activities for the first the day of the week. 2. The *Administrator* enters the desired time values 3. Steps 1 to 2 are repeated for each day of the week. 4. The *Administrator* requests the system to saves the entered values 5. The System validates and saves the data. 6. The use case ends | | |
| **Alternative flow:** | | |

|  |  |  |
| --- | --- | --- |
| **Use case name**: Create login details | **ID**: UC-14 | **Priority**: High |
| **Actor**: Local Administrator, Global Administrator, volunteer | | |
| **Description**: this use case describes how a local/global administrator or a volunteer creates its login details, username and password. | | |
| **Trigger**: the actor need to login into the system to accomplish certain task.  **Type**:   * External * Temporal | | |
| **Preconditions**:  None | | |
| **Basic flow:**  **1.0 This use case starts with the *User* being invited to register through a link with a validity period sent to his/her email account**   1. The *User* visits the login creation link 2. The System validates the link and prompts the User with the login form to fill 3. The *User* fills out the form, entering username and password and submits it 4. The System validates the information submitted and saves it to the User Account information 5. The use case ends   Invalid *login creation* link:   * Link not created by the System * The validity period of link is expired | | |
| **Alternative flow:** | | |

|  |  |  |
| --- | --- | --- |
| **Use case name**: Login | **ID**: UC-15 | **Priority**: High |
| **Actor**: Local Administrator, Global Administrator, Volunteer | | |
| **Description**: this use case describes how a local/global administrator or a volunteer logs in the system. | | |
| **Trigger**: the User want to log in.  **Type**:   * External * Temporal | | |
| **Preconditions**:  The User has a login credential | | |
| **Basic flow:**  **1.0 This use case starts when the *User* (Admin, Volunteer) accesses the sign in feature of the system.**   1. The system prompts the *User* for his/her username and password. 2. The *User* enters his/her username and password. 3. The system validates the entered information, making sure that the entered username and password are valid for one user account in the system, and that the required password is entered for the entered username. 4. The *Use*r is signed in. The system displays a message indicating that the user is signed in. 5. The use case ends. | | |
| **Alternative flow:**  **1.1 User Fails Authentication** [branch 3]   1. If the *User* forgot his/her username or password, the System will display an “invalid credential” message. 2. The System also displays a password forgot link that will require the user to input his/her email address provided when the account was created. 3. The system prompts the User to re-enter the valid information or to click on the “forgot password link”    1. The User decides to re-enter username and password. The Basic Flow continues where the *User* enters new information (see step 2 of the Basic Flow).    2. The User clicks on the password forgot link. Provided that the email address associated with his/her account is answered correctly, A password reset link is emailed to that email address. | | |

|  |  |  |
| --- | --- | --- |
| **Use case name**: Update availability | **ID**: UC-16 | **Priority**: Low |
| **Actor**: Volunteer | | |
| **Description**: this use case describes how a volunteer change its availability for paying visit. This can also be done by an administrator on his/her behave after receiving phone call or email. The updated availability can lead to “not available” which is equivalent to cancellation. Depending on the “cancellation policy”, a cancellation may merely lead an email notification to the recipient that the host has cancelled his visit in worse case scenario. The best scenario include | | |
| **Trigger**: the volunteer want to change its availability.  **Type**:   * External * Temporal | | |
| **Preconditions**:  The volunteer has signed in the system to make the change | | |
| **Basic flow:**  **1.0 This use case starts when the User accesses the system feature that enables him/her to update this/her availability that is maintained in the User's account.**   1. The System displays the User availability information currently stored for the User. 2. The User enters the desired availability information values and requests that the system saves the entered values. 3. The system validates the entered User availability information. 4. The values for the User availability information are stored in the User’s account. The system notifies the User that the account has been updated. 5. The use case ends | | |
| **Alternative flow:**  **1.1 User Cancels Availability Update Request**   1. At any time, the User may choose to cancel the availability update. At which point, the processing is discontinued, the user availability remains unchanged, and the user is notified that the availability update request has been cancelled.   **1.2 User Enters Invalid Availability Information**   1. If during Update availability, the system determines that the User entered invalid availability information, the following occurs: 2. The system describes which entered data was invalid and presents the User with suggestions for entering valid data. 3. The system prompts the User to re-enter the invalid information. 4. The User re-enters the information and the system re-validates it. 5. If valid information is entered, the availability information is stored. 6. If invalid information is entered, the Entered Information is Invalid alternative flow is executed again. This continues until the User enters valid information, or chooses Cancel (see the User Cancels Account Management Request alternative flow).   Invalid User Account information:   * Missing information items * Username already exists in the system * User Account information entered does not comply to its definition in the glossary * Not well formed email address * Offending words in any part of the User Account information | | |

|  |  |  |
| --- | --- | --- |
| **Use case name**: View city report | **ID**: UC-17 | **Priority**: Low |
| **Actor**: Local Administrator, Global Administrator | | |
| **Description**: this use case describes how a local/global administrator consults the report generated by the system. (The report to be generated not yet been precised) | | |
| **Trigger**: the administrator want to get an overview/statistics of what is happening or has happened  **Type**:   * External * Temporal | | |
| **Preconditions**:  *Depending on the nature of the report, this will vary* | | |
| **Basic flow:** | | |

|  |  |  |
| --- | --- | --- |
| **Use case name**: View global report | **ID**: UC-18 | **Priority**: Low |
| **Actor**: Global Administrator | | |
| **Description**: this use case describes how a local/global administrator consults the report generated by the system for comparative assessment of all cities. (The report to be generated not yet been precised) | | |
| **Trigger**: the administrator want to get an overview/statistics of what is happening or has happened in all cities.  **Type**:   * External * Temporal | | |
| **Preconditions**:  *Depending on the nature of the report, this will vary* | | |
| **Basic flow:** | | |

### New use cases

++ Helpers that do not show up (Though scenario to manage, suggest taking a proactive approach by given the volunteer the possibility to cancel earlier enough (a week before), by including a cancellation/update availability link in the general email notification or sending a complete new email “update availability” . choice may be guided by the cost implication. This can be done before the first initial assignment)

++ Helpers that cancel (can happen through phone call, email or use case “update availability” that lead to non availability or cancellation)

++ Helper reminder notifications few days or hours before the meeting

++ Membership to Elsk movement (create an elsk city)

++ Business process description diagram.

When are volunteers notified where and who to visit? (1 day, 2 day before the visit!!!, etc.)

* Answer useful for UC-16: “update availability” to see whether the schedule need to be regenerated as a whole or with a new constraint “keep as much as possible the existing assignment”
* Sometimes, this can happen 5 days before and at “Alborg” the email specify a meeting point where they will be communicated where to go
* A recipient can be assigned different volunteers like a team
* In some cities, the email sent just tell the recipient the venue to help the recipient
* The general reminders email

**Email notifications**

At signup —> Confirmation (signup ends 1 day before 1st event date)

2 weeks before first event date —> Individual reminder

5 days before first event date —> General info and reminder (recipients and helpers)

1 day before every event date —> Individual reminder (recipients and helpers) incl. location

10 days after event date —> Survey (recipients and helpers)

**Help Category should be automated**

Rules for assignment

By default only Green should be assigned

If a recipient makes a request that requires special needs like computer which is more difficult to match, then it should be assigned category yellow

Check rules on Pjece om elsk pdf file on the share drive

* **Green:** recipient has the know-how (help types which does not require special volunteer abilities) AND has the tools
* **Yellow:** recipient lacks the know-how (help types which requires special volunteer abilities) OR lacks the tools
* **Red:** recipient requires special care (manually assigned) - i.e. from description box

------

The user with red category can be manually assigned. The suggestion now is to make a suggestion-dialogue-box of which recipient should be red based on whether they filled out the input field “any special comment of request form”. This suggestion-dialogue-box might be shown only once, the first time the schedule is run.

---

Amend the EER diagram to reflect the fact that a help type should have an attribute for whether it requires special ability - add a table and also a form for creating a new help-type with a field related to the special ability.