

KU-FS-2021Q4 - Final Project: Staying Alert

Data Structures

PROJECT NAME: _____ Staying Alert
PROJECT START: _____ June 8, 2022
TEAM NAME: _____ S.J.M.
TEAM MEMBERS: _____ Stephen Svetlovics, James Ingram II and Manuel "Manny" Muro
CREATED BY: _____ Manuel Muro
CREATED ON: _____ June, 8, 2022
LAST UPDATED BY: _____ Manuel Muro
LAST UPDATED ON: _____ June 19, 2022 @ 11:02 (PST)

PROJECT DESCRIPTION

The purpose behind the "Staying Alert" project was to serve as a software version of the "Medical Alert" bracelets and necklaces, used in the past, to warn others that the person wearing the medical alert item as having a medical condition that should the wearer need help that information would be taken into consideration and be made readily available. However, what if someone does not have a medical condition, but is involved in a situation that other people should be notified that a given person needs help, think amnesia or dementia, but there are probably other such conditions where getting in touch with others, such as a lost child, connecting with others can be extremely helpful. As we brainstormed, we also saw the need beyond just medical conditions, such as people going on a date or sleep-over, so we identified two types of "Alerts": 1) Assistance and 2) Check-In Alerts. The Assistance type alerts will cover medical conditions and general unplanned needs for assistance, such that there will only be one assistance alert per person. Check-In alerts can be multiple such that the people to do a date based check-in may be different than an emergency based alerts.

Depending on the nature of the alert(s), it might be desirable for a user of the "Staying Alert" system to set up alerts for other people such as for their children or elderly parents. For this reason, as the user of the system adds people via their account, they also need to indicate if the person that they are adding is a "self" entry. While a user can enter several people, there can only be one person tagged as "self". This individual should also set up an assistance alert for themselves, so that such assistance information can be used for their benefit as well; ideally via an emergency alert button that is accessible on their smartphone even if the phone is locked.

Aside from the option to access the emergency assistance alert information from a person's smart, and really independent of the smartphone itself, for each person a unique QSR code will be generated that contains the URL information to access the emergency assistance information for the person needing assistance, but underneath or around the QSR code will be a human readable version of the URL, so that if someone else has a device that can access the internet they can access the assistance alert information, and the key point being without needing to log into the application. On this page, the person providing assistance will be presented with a form that will allow them to provide: 1) their name 2) their contact information and 3) a message as to why assistance is needed. That information will then be sent to the person, or group of persons, listed as being the contact(ees) for the person in need of assistance.

Check-in alerts will be more involved, with information relating to the duration and frequency of check-ins related activities. Additionally, unlike assistance alerts, check-in alerts can be activated and deactivated as they may not always be needed.

DATA STRUCTURE ANALYSIS

Based on the project's objectives and the kind of information that can be anticipated as needing to be tracked, using basic normalization concepts of data to eliminate and data duplication and for more efficient storage of the data, the number of data object arrays will be six and they will contain the following type of information:

- 1) User Account Data
- 2) Person Data
- 3) Group Data
- 4) Assistance Data
- 5) Check-In Data
- 6) Transactions Data

The fields, or attributes, associated with each object array will be listed in the sections that follow, along with a description of the data, including if it is a required field, as well as indication of where the field data will be coming from: a) the user, b) the database or c) the application. While the user data fields will obviously require manual entry from the user, the database will only be providing unique IDs for each item in a given object array and each array will have a field with the name "**id**" that will contain the aforementioned unique ID for each item in the array. The application will automatically, and properly, populate the date and time fields as well as ensuring the proper cross correlation ID field is populated with the correct ID value for any ID related field, which will be denoted with: **<field_name>_id**, as such fields will not be automatically managed by the database. Following each field will be the source of how the field gets its data as denoted in parenthesis following the field name, using the following keys: **USR**, **DB** and **APP**, which correspond appropriately with: *User*, *Database* and *Application*.

ATTENTION: All fields are text/string based fields and are optional, i.e. they can be left blank, unless specified otherwise, with Boolean fields defaulting to false rather than being left blank. The actual coding name for the specific data array is shown in parenthesis, in quotes and is bolded.

User Account Data ("**user_account**")

The User Account Data array will contain information related to the user including: name (both the system profile name and real life name), contact information, security access keys for the user's account as well date and time information related to when the account was created and last modified. The fields for the "user_account" array are as follow:

- 1) **id** (DB/APP) -required, the cross-reference field for this "id" will be "**user_id**" or something closely similar.
- 2) **user_name** (USR) - required
- 3) **first_name** (USR) - required
- 4) **last_name** (USR) - required
- 5) **email_addr** (USR) - required
- 6) **createdAt** (DB) - Date type, which includes date and time information.
- 7) **updatedAt** (DB) - Date type, which includes date and time information.
- 8) **password_hash** (USR + APP)
- 9) **emergency_msg** (USR) - When an emergency assistance alert is activated this is the message that will be sent to the desired contact(s)
- 10) **emer_person_id** (USR) - If the "emergency_msg" contains actual text, i.e. not empty, this either the "person" or "group" id field needs to be set to a valid id value or they both can contain valid id values.
- 11) **emer_group_id** (USR) - If the "emergency_msg" contains actual text, i.e. not empty, this either the "person" or "group" id field needs to be set to a valid id value or they both can contain valid id values.

- 12) **loc_interval_in_mins** (USR) - If the “emergency_msg” contains actual text, i.e. not empty, this field **MUST** contain a positive value greater than zero that specifies the number of minutes to wait between capture and sending the location information for the user of this application.
- 13) **removed** (APP) - Boolean data type, for reporting reasons with respect to transactional activity, this field is used to indicate that the user has been effectively removed from being used in the system, but prior to the user being removed, reports will still use this information to properly report on prior transactions.

Person Data (“person”)

The Person Data array will contain information that is the heart of the “Staying Alert” application, as both the persons that are being “protected” and those persons providing protection via this application must be entered into this data array before they can be protected or provide protection. The fields for “person_data” array are as follow:

1. **id** (DB/APP) -required, the cross-reference field for this “id” will be “**person_id**” or something closely similar.
2. **user_id** (APP) - required, this is the “id” for the user to whom this “person” item was created by and should be associated with as contained in the “user_account” data array.
3. **first_name**(USR) - required
4. **last_name** (USR) - required
5. **pref_name**(USR) - required, i.e. preferred name.
6. **phone** (USR) - required if email is not provided. If provided, assistance requests and check-in notifications will be sent via SMS messaging to the phone number provided.
7. **email** (USR) - required if phone is not provided. If provided, assistance requests and check-in notifications will be sent to the e-mail address provided.
8. **check_in_id_arr** (APP) - This is an array of valid ids for “check-in” alerts for a given person.
9. **check_in_group_id** (APP) - When creating a check-in alert, this is the default “check-in” group id assigned for a new check-in alert, this can be left blank.
10. **check_in_person_id** (APP) - When creating a check-in alert, this is the default “check-in” person id assigned for a new check-in alert, this can be left blank.
11. **assist_group_id** (USR via APP) - When this person needs assistance this is the “group_id” to send out requests for additional assistance to.
12. **assist_person_id** (USR via APP) - When this person needs assistance this is the “group_id” to send out requests for additional assistance to.
13. **assist_id** (USR via APP) - The assistance id record that is to be associated with this person.
14. **person_is_self** (USR) - Boolean data type, indicates that this “person” record is the record for the actual user of this application, and for a given “user_id”, this field can only be set for just one “person”.
15. **createdAt** (DB) - Date type, which includes date and time information.
16. **updatedAt** (DB) - Date type, which includes date and time information.
17. **check_in_confirm_arr** (APP) - This is an array of “id” fields of persons that this specific person has confirmed that they will accept “check-in” alerts on behalf of.
18. **assist_confirm_arr** (APP) - This is an array of “id” fields of persons that this specific person has confirmed that they will accept “check-in” alerts on behalf of.
19. **removed** (APP) - Boolean data type, for reporting reasons with respect to transactional activity, this field is used to indicate that the person has been effectively removed from being used in the system, but prior to the person being removed, reports will still use this information to properly report on prior transactions.

Group Data (“groups”)

The Group Data array will contain information that creates a named collection of persons. The fields for the “groups” array are as follow:

1. **id** (DB/APP) - Required, the cross-reference field for this ID will be “**groups_id**”
2. **group_name** (USR) - required, this is the human readable name to be associated with this record.
3. **person_id_arr** (USR via APP) - required, the collection of “person_id”s that constitute the group.
4. **user_id** (APP) - required, the id of the user to be associated with and who created and can maintain this group.
5. **createdAt** (DB) - Date type, which includes date and time information.
6. **updatedAt** (DB) - Date type, which includes date and time information.
7. **removed** (APP) - Boolean data type, for reporting reasons with respect to transactional activity, this field is used to indicated that the group has been effectively been removed from being used in the system, but prior to the group being removed, reports will still use this information to properly report on prior transactions.

Assistance Data (“**assistance**”)

The Assistance Data array will contain information for the basic information that needs to be shared when a person associated with this assistance record needs immediate and pressing help. The fields for the “user_vitals” array are as follow:

1. **id** (DB/APP) -required, the cross-reference field for this “id” will be “**assist_id**” or something closely similar.
2. **user_id** (APP) - The id of the user that this assistance record is to be associated with.
3. **person_2_assist_id** (USR via APP) - Required, the “person_id” for the person that will be in need of extra assistance.
4. **critical_info_msg** (USR) - Message to be presented immediately upon request for additional assistance for the person specified via “person_2_assit_id”.
5. **removed** (APP) - Boolean data type, for reporting reasons with respect to transactional activity, this field is used to indicated that the group has been effectively been removed from being used in the system, but prior to the group being removed, reports will still use this information to properly report on prior transactions.
6. **createdAt** (DB) - Date type, which includes date and time information.
7. **updatedAt** (DB) - Date type, which includes date and time information.
8. **removed** (APP) - Boolean data type, for reporting reasons with respect to transactional activity, this field is used to indicated that the assistance has been effectively been removed from being used in the system, but prior to the assistance being removed, reports will still use this information to properly report on prior transactions.

Check-In Data (“**check_in**”)

The Check-In Data array will contain information that check-in related information that is related to a specific person, i.e. independent of the user so that a user can set up check-in alerts for other people.

1. **id** (DB/APP) -required, the cross-reference field for this ID will be “**check_in_id**” or something closely similar.
2. **user_id** (APP) - The id of the user that this check-in record is to be associated with.
3. **check_in_name** (USR) - required, this is the human readable name to be associated with this record.
4. **chk_in_assoc_person_id** (USR via APP) - required, this is the person in need of a check-in to be associated with this record.
5. **start_date** (USR) - Date type, independent of the time aspect, only the date information is relevant as to when this check-in is to be carried out, i.e. the time element is ignored.
6. **end_date** (USR) - Date type, independent of the time aspect, only the date information is relevant as to when this check-in is to be carried out, i.e. the time element is ignored.
7. **start_time** (USR) - Date type, independent of the date aspect, only the time information is relevant as to when this check-in is to be carried out, i.e. the date element is ignored.

8. **end_time** (USR) - Date type, independent of the date aspect, only the time information is relevant as to when this check-in is to be carried out, i.e. the time element is ignored.
9. **frequency_qty** (USR) - Number type, the number of time units specified to wait before performing a check-in action.
10. **frequency_units** (USR) - The time units to wait before performing a check-in action, i.e. hour, minutes, seconds, days, weeks, months, etc.
11. **active** (USR) - Boolean type, beyond the date and time windows, this is a separate control to determine if a check-in action is to be performed.
12. **contact_person_id** (USR via APP) - The person to contact when performing a check-in action.
13. **contact_group_id** (USR via APP) - The group to contact when performing a check-in action.
14. **action_plan** (USR) - The plan of action strategy that is sent out to the person and/or group when performing a check-in action.
15. **createdAt** (DB) - Date type, which includes date and time information.
16. **updatedAt** (DB) - Date type, which includes date and time information.
17. **removed** (APP) - Boolean data type, for reporting reasons with respect to transactional activity, this field is used to indicate that the check-in has been effectively removed from being used in the system, but prior to the check-in being removed, reports will still use this information to properly report on prior transactions.

Transaction Data (“transactions”)

The Transaction Data array will contain information that logs each action taken with respect to this entire application in as compact a form as possible. The time and date information is just a “stamp”, a capture of the time when the user requests a given action to be taken.

NOTE: For data integrity purposes, there is no logical reason at present to flag a transaction item has been removed. In the event that the transaction item does need to be removed, such as application testing, those entries should be manually deleted/removed by an individual with database admin access.

The fields for the “transactions” array are as follow:

1. **id** (DB/APP) - required, the cross-reference field for this ID will be “trans_id” or something closely similar.
2. **trans_type** (APP) - required, the sheer number of types of transaction can vary and may not be known apriori, but the “trans_type” needs to be an easily readable 2 to 3 word description describing the nature of the transaction. For example, “Person Add”, “Assist Update”, “Group Delete”, “Check-In Acknowledge” or “Emergency Alert Sent”.
3. **message_sent** (APP) - If the transaction involved sending out a message of some sort, a raw copy of the actual message sent, even if generated, will be copied here.
4. **person_supported_id** (APP) - The id of the person being supported via this transaction.
5. **trans_date** (APP) - Date type, which will include the date and time. Since transactions should only get created, and never updated, the application should take care of capturing the current system date and time to “stamp” the transaction records instead of letting the database do it automatically.
6. **person_id_arr** (APP) - An array of “id”s for the person(s) that were involved in this transaction.
7. **user_id** (APP) - The id of the user that this transaction record is to be associated with.
8. **system_gen_act_notes** (APP) - This is a field to track system related transactional activity, such as changing the user name with an automatically generated message such as the following: “USER NAME CHANGE: {old_name} => {new_name}”.