Emergency Resource Management System Documentation

Table of Contents

Data Types

User

Attribute	**Data Type**	**Nullable**
name	String	NOT NULL
username	String	NOT NULL
password	String	NOT NULL

Individual

Attribute	**Data Type**	**Nullable**
JobTitle	String	NOT NULL
DateHired	Date	NOT NULL

Municipality

Attribute	**Data Type**	**Nullable**
City	String	NOT NULL
County	String	NOT NULL
State	String	NOT NULL
Country	String	NOT NULL

GovernmentAgency

Attribute	**Data Type**	**Nullable**
AgencyNameAndLocalOffice	String	NOT NULL

Company

Attribute	**Data Type**	**Nullable**
Location	String	NOT NULL
NumOfEmployees	Integer	NOT NULL

Resource

Attribute	**Data Type**	**Nullable**
Name	String	NOT NULL
Model	String	NULL
Capabilities	List<String>	NULL
HomeLatitude	Float	NOT NULL
HomeLongitude	Float	NOT NULL
MaxDistance	Float	NULL
Cost	Float	NOT NULL

TimeUnit

Attribute	**Data Type**	**Nullable**
UnitName	String	NOT NULL
LengthOfUnit	Float	NOT NULL

ESF

Attribute	**Data Type**	**Nullable**
Number	Integer	NOT NULL
Description	String	NOT NULL

Incident

Attribute	**Data Type**	**Nullable**
Number (Partial Key)	Integer	NOT NULL
Date	DateTime	NOT NULL
Description	String	NOT NULL
Latitude	Float	NOT NULL
Longitude	Float	NOT NULL

Request

Attribute	**Data Type**	**Nullable**
ReturnDate	Date	NOT NULL
RequestDate	Date	NOT NULL

In Use

Attribute	**Data Type**	**Nullable**
ReturnDate	Date	NOT NULL
StartDate	Date	NOT NULL

Declaration

Attribute	**Data Type**	**Nullable**
Abbreviation	String	NOT NULL
Name	String	NOT NULL

Business Logic Constraints

ERMS User

- User information will be loaded by the database administrator behind the scenes.
- Number of employees must be ≥ 0 for companies.

Add New Resource

- The value of Latitude must be in [-90, 90]. (North is positive, South is negative)
- The value of Longitude must be in [-180, 180]. (West is negative, east is positive)
- The value of Max Distance must be NULL or in [0, 20038] km. (20038 km is slightly larger than half of the circumference 40075.017 km of the earth)
- The numeric value of cost must be ≥ 0.

Add New Incident

- The Date must be before or equal to current computer system date.
- The value of Latitude must be in [-90, 90]. (N is positive, S is negative)
- The value of Longitude must be in [-180, 180]. (W is negative, E is positive)

Search Resource

- The value of “Location within ___ km” field must be NULL or in [0, 20038].

Search Results

- Users cannot request resource owned by themselves.

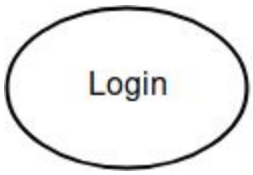
Other Constraints for Attributes

- For Time Unit entity, its Length of Unit (in seconds) must be ≥ 0.
- For Request and In Use relations, their return date must be equal or later than current date.

Task Decomposition with Abstract Code

Login to ERMS

Task Decomp



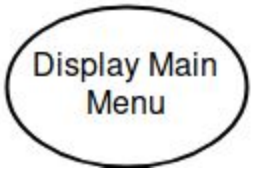
Lock Types: Read-only on User Table
Number of Locks: Single
Enabling Conditions: None
Frequency: Around 200 logins per day
Consistency (ACID): Not critical, order is not critical.
Subtasks: Mother Task is not needed. No decomposition needed.

Abstract Code

- Display username and password fields
- User enters *username* ('$Username'), *password* ('$Password') input fields.
- When ***Login*** button is clicked:
    - If data validation is successful for both *username* and *password* input fields, then:
        - If User record is found but User.password != '($Password)':
            - Go back to **Login to ERMS** form, with error message.
        - Else:
            - Store login information as session variable '($UserID)'.
            - Go to **Main Menu**.
    - Else *email* and *password* input fields are invalid, display **Login to ERMS** form, with error message.

Main Menu

Task Decomp



Lock Types: Lookup User Tables (for Name and their type), all are Read-only.
Number of Locks: Single
Enabling Conditions: Trigger by successful login.
Frequency: User Detail and Menu Options have the same frequency.
Consistency (ACID): not critical, order is not critical.
Subtasks: Mother Task is not needed. No decomposition needed.

Abstract Code

- Lookup and display different User Tables for User.username and type
- Show "***Add Resource***", "***New Incident***", "***Search Resources***", "***Resources Status***" and "***Resource Report***" tabs.
- Upon:
    - Click ***Add Resource*** button- Jump to the **Add Resource** task.
    - Click ***New Incident*** button- Jump to the **New Incident** task.
    - Click ***Search Resources*** button- Jump to the **Search Resources** task.
    - Click ***Resources Status*** button- Jump to the **Resources Status** task.
    - Click ***Resource Report*** button- Jump to the **Resource Report** task.
    - Click ***Exit*** button- Invalidate login session and go back to the **Login to ERMS** form.

Add New Resource

Task Decomp

Lock Types: Write-only on Resource Table AdditionalESF Table, read only on ESF Table and TimeUnit Table
Number of Locks: Four. Several different schema constructs are needed

Enabling Conditions: Enabled when a user clicks *Add a Resource* from the main menu
Frequency: Depending on the rate of increase of resources
Consistency (ACID): Not critical, order is not critical.
Subtasks: Mother Task is not needed. No decomposition is needed as the search in ESF and TimeUnit table is trivial. AdditionalESF Table and Resource Table are always changed together as AdditionalESF is actually an "attribute" of Resource so it is not necessary to split them.

Abstract Code

- Generate and display a unique numeric ID
- Read the ESF Table and display dropdown menu for *Primary ESF* and multichoice list for *Additional ESF*
- Read the TimeUnit Table and display the dropdown menu for *Time Unit*
- Find the logged in user; display the logged in user's own User.username as the resource's owner
- The user fills out the **New Resource** Form for the new resource, including recording *Resource Name*, *Primary ESF*, *Additional ESFs*, *Model*, *Capabilities*, *Home Location*, *Cost/Cost Per* and *Maximum Distance*.
- When the user clicks ***Save*** button:
    - *Resource Name*, *Primary ESF*, *Additional ESFs*, *Model*, *Capabilities*, *Home Location*, *Cost/Cost Per* and *Maximum Distance* are saved
    - The resource is assigned the unique numerical ID automatically
    - The resource's owner is also set automatically to the logged in user
- When the user clicks ***Cancel*** button:
    - Go back to the **Main Menu**

New Incident

Task Decomp

Lock Types: Write-only on Incident Table and read only on Declaration Table
Number of Locks: Two. Several different schema constructs are needed
Enabling Conditions: Trigger by click ***New Incident.***
Frequency: Depending on the rate of incident
Consistency (ACID): not critical, order is not critical.

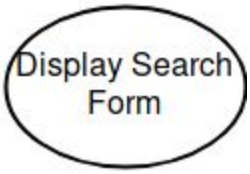
Subtasks: Mother Task is not needed. No decomposition is needed as the search in Declaration table is trivial.

Abstract Code

- Read the Declaration Table and display dropdown menu for *Declaration*
- The user fills out the **New Incident** Form for the new emergency incident, including choosing a incident *declaration*, recording the *date* of the incident, a brief *description*, and the *latitude/longitude* coordinates of the incident.
- When the user clicks ***Save*** button:
    - The incident *declaration*, the *date* of the incident, a brief *description*, and the *latitude/longitude* coordinates of the incident are saved into Incident table
    - The incident is automatically assigned a unique ID that combines the abbreviation of the incident declaration abbreviation with an automatically generated number unique to that incident declaration
    - the owner of the incident is automatically set to the current user
- When the user clicks ***Cancel*** button:
    - Go back to the **Main Menu**

Search Resources

Task Decomp



Lock Types: Read-only on Incident Table and ESF Table
Number of Locks: Two
Enabling Conditions: Trigger by clicking ***Search Resource*** in **Main Menu** form
Frequency: 2000 search requests per day.
Consistency (ACID): not critical, order is not critical.
Subtasks: Mother Task is not needed. No decomposition is needed as the search in ESF table is trivial.

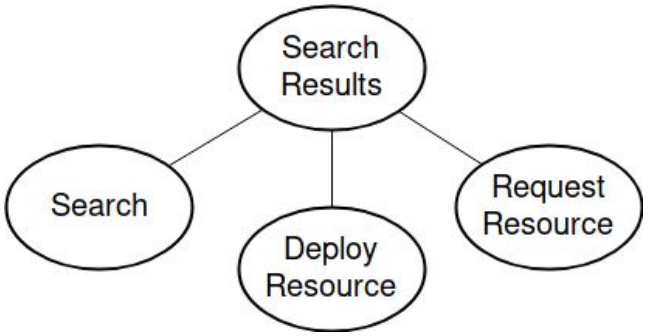
Abstract Code

- Get the ESF Table and display them in *ESF* dropdown menu

- Find a list of incidents that is owned by current user. That is to say Incident.owner match current user's User.username. Display them in *Incident* dropdown menu
- User enters *keyword* ('$Keyword'), chooses *ESF* ('$ESF') from ESF dropdown menu, enters *radius* ('$Radius') and chooses emergency *Incident*('$Incident') from Incident dropdown menu.
- Upon:
    - Click ***Search*** button-
        - If all the dropdown selection and Location meet the requirement (can be NULL):
            - Jump to the **Search Results** task.
    - Click ***Cancel*** button- jump back to the **Main Menu**.

Search Results

Task Decomp



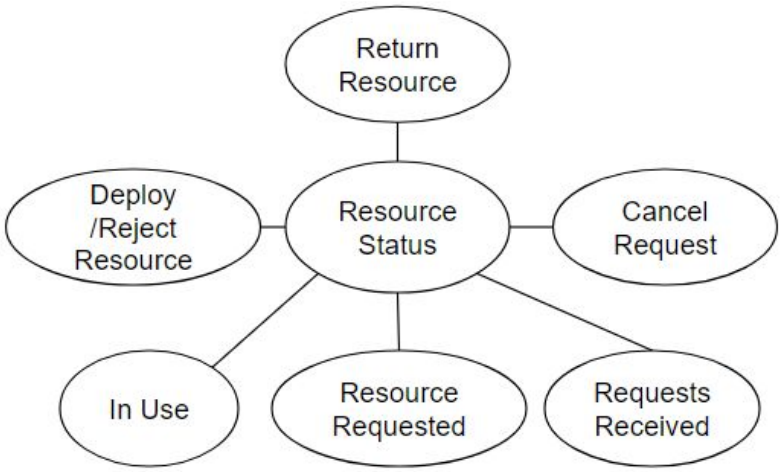
Lock Types: Read-only on Resource, and AdditionalESF tables. Read and write for Requests and InUse tables.
Number of Locks: Three. Several different schema constructs are needed
Enabling Conditions: Trigger by clicking ***search*** button in **Search Resources** form.
Frequency: 2000 search requests per day, 5000 resource requests per day and 2000 deployment of (own) resource per day
Consistency (ACID): Critical for deployments and requests, order is not critical.
Subtasks: Mother Task is needed to coordinate subtasks. Deploy/Request Resource should be done after Search task.

Abstract Code

- Display incident description and ID if Incident was selected
- Find a list of resources that *keyword*('$Keyword') is matched in Resource.name Resource.model or Resource.Capabilities, proximity to incident is within *Location*('$Location'), and *ESF*('$ESF') is matched in Resource.PrimaryESF or AdditionalESF. Note that Proximity is calculated using haversine formula. If any of the filters are NULL, do not validate it.
- Find availability (resource status) from the InUse Table
- Display resource ID, resource name, resource owner, resource cost and cost per unit, resource status, next available date, and distance
- If *Incident* was selected
    - For all resources row:
        - If Resource.owner is current user:
            - If resource is available, display ***Deploy*** button.
            - Else do not display button
        - Else display ***Request*** button
- Upon:
    - Click ***Request*** Button (if there exists such a button)-
        - Popup a dialog to get an Return Date
        - Update Request Table with Incident ID, Resource ID, Return Date, Request Date. Note underlined part is the key for table.
        - Gray shade the clicked button and make it unclickable
    - Click ***Deploy*** buttons (if there exists such a button)-
        - Popup a dialog to get an Return Date
        - Update InUse Table with Resource ID, Incident ID, Return date. Note underlined part is the key for table. For every resource, it is guaranteed there is no more than one Incident taking it.
        - Gray shade the clicked button and make it unclickable
    - User can click ***Close*** to go back to **Main Menu**

Resource Status

Task Decomp



Lock Types: Read-only on Resource and Incident tables. Read and Write on Requests, InUse, and LastUsed tables.
Number of Locks: Five. Several different schema constructs are needed
Enabling Conditions: Triggered by clicking ***Resource Status*** from **Main Menu**
Frequency: Low
Consistency (ACID): Not Critical
Subtasks: In Use, Resource Requested, Requests Received must be done but can be done in parallel for mother task Resource Status to be done. Deploy/Reject Resource, Return Resource, Cancel Request must be done after mother task.

Abstract Code

- Find all resources that are in InUse Table and display Resource.Name, Incident.Description, resource owner's User.Name, Start Date, Return Date from InUse table and ***Return*** button.
- Find all resources that are requested by me and display Resource.Name, Incident.Description, resource owner's User.Name, Return Date from Requests table and ***Cancel*** button.
- Find all resource requests that are received by me and display Resource.Name, Incident.Description, resource owner's User.Name, and Return Date from Requests table.
    - If resource is in use, display ***Reject*** button.
    - Else, display ***Deploy*** and ***Reject*** buttons
- If the user clicks ***Return*** in the Resource in Use list:
    - **Return Resource**: Remove the resource from the Resource InUse Table

    - Update the LastUsed table with this returned resource
    - Grayshade ***Return*** and make it unclickable
- If the user clicks ***Cancel*** in the Resources Requested by Me list:
    - **Cancel Request**: Remove the request from the Requests Table
    - Grayshade ***Cancel*** and make it unclickable
- If the user clicks ***Deploy*** in the Resource Requests Received by Me list:
    - **Deploy Resource**: Move the resource from Resource Requests Table to Resource in Use Table and record the current date as Start Date
    - Grayshade ***Deploy*** and make it unclickable
- If the user clicks ***Reject*** in the Resource Requests Received by Me list:
    - **Reject Resource**: Remove the resource from Resource Requests Table
    - Grayshade ***Reject*** and make it unclickable

Resource Report

Task Decomp

Lock Types: Read-only on ESF and InUse table, Resource Table
Number of Locks: Three
Enabling Conditions: Triggered by clicking ***Resource Report*** from **Main Menu**
Frequency: Low frequency
Consistency (ACID): not critical
Subtasks: Mother task are not needed. No decomposition needed. Although there are three reading from two tables, the ESF one is trivial. For the remaining two, inconsistent result is fine.

Abstract Code

- Display *ESF#* and *Primary Emergency Support Function* columns according to ESF Table
- Generate a Resource table grouped by ESF# ('$ESF_number') with Primary Emergency Support ('$ESF') Function, Total Resources
- Query the number of resources in each ESF group from the table obtained in step 2 and display in Total Resources column
- Select resources that are in InUse table from the table obtained in step 2
- Query the number of resources in each ESF group, and display *Resources in Use* column accordingly.
- Click [X](the close window button) to jump back to **Main Menu**.