Georgia Tech

Master of Science in Analytics

Emergency Resource Management System

Phase 1 Report | CS 6400 | Team 05

Table of Contents

Data Types

Assumptions

- Attribute data types are listed by entity. In cases where a data attribute was not explicitly labeled as “optional”, it was assumed to be required and hence not nullable.
- All surrogate and foreign keys were excluded from the data type list as per instructors comments in Piazza @136
- In cases where two attributes combine to give one piece of information about an entity e.g. Latitude and Longitude combined to show Location. Only the unique component attributes (Latitude and Longitude) were shown to avoid unnecessary redundancy.

User

Attribute	Data Type	Nullable
Name	String	Not Nullable
UserName	String	Not Nullable
Password	String	Not Nullable

Municipality

Attribute	Data Type	Nullable
Category	String	Not Nullable

Company

Attribute	Data Type	Nullable
HQLocation	String	Not Nullable
NumberOfEmployees	Integer	Not Nullable

Government Agency

Attribute	Data Type	Nullable
AgencyName	String	Not Nullable
LocalOffice	String	Not Nullable

Individual

Attribute	Data Type	Nullable
JobTitle	String	Not Nullable
DateHired	Date	Not Nullable

Resource

Attribute	Data Type	Nullable
Name	String	Not Nullable
PrimaryESF	String	Not Nullable
AdditionalESFs	List<String>	Nullable
Model	String	Nullable
Capabilities	List<String>	Nullable
Latitude	Float	Not Nullable
Longitude	Float	Not Nullable
Cost	Float	Not Nullable
CostPer	String	Not Nullable
MaximumDistance	Float	Nullable
AvailabilityStatus	String	Not Nullable

Incident

Attribute	Data Type	Nullable
Type	String	Not Nullable
Date	Date	Not Nullable
Description	String	Not Nullable
Latitude	Float	Not Nullable
Longitude	Float	Not Nullable

Business Logic Constraints

Users

- Users are added by an admin behind the scenes
- No registration capabilities exist
- Municipality User Categories come from a predefined list
    - City
    - County
    - State
    - Country

Adding Resources

- Resources are automatically given unique IDs when added to the system
- Resource owner is automatically set to the logged-in user that created the resource
- Resource ESF's come from a predefined list
    - Transportation
    - Communications
    - Public Works and Engineering
    - Firefighting
    - Emergency Management
    - Mass Care, Emergency Assistance, Housing, and Human Services
    - Logistics Management and Resource Support
    - Public Health and Medical Services
    - Search and Rescue
    - Oil and Hazardous Materials Response
    - Agriculture and Natural Resources
    - Energy
    - Public Safety and Security
    - Long-Term Community Recovery
    - External Affairs
- Resource ESF's list should be changeable by admin behind the scenes
- Resource coordinates should be valid latitude and longitude coordinates in decimal degree format
- Resource cost types initially include:
    - Hourly
    - Daily
    - Weekly
    - Each
- Resource Cost types should be changeable by admin behind the scenes

- Resource Cost cannot be negative
- New resources added to the system are set to “Available” by default
- Resources should be viewable to all users

Adding Incidents

- Incidents are automatically given unique IDs when added to the system
    - Consists of a combination of the abbreviation of the incident type with an automatically generated number unique to that incident type
        - Each incident type starts increments by 1 starting from a value of 1
- Incident types come from a predefined list
    - Major Disaster Declaration (abbreviated MD)
    - Emergency Declaration (abbreviated ED)
    - Fire Management Assistance (abbreviated FM)
    - Fire Suppression Authorization (abbreviated FS)
- Incident types should be changeable by admin behind the scenes
- Incident owner is automatically set to the logged-in user that created the incident
- Incident coordinates should be valid latitude and longitude coordinates in decimal degree format
- Incident names do not need to be unique
- Incidents should only be viewable to the user that created the incident and cannot be shared

Searching Resources

- Search results return resources that meet the user’s criteria regardless of resource owner (including own resources)

Resources Search Results

- Search parameter behavior includes:
    - Matching substrings with keyword from any data element listed below:
        - Resource name
        - Model
        - Capabilities
    - Exact match with emergency support function
        - Primary ESF
        - Additional ESF
    - Less than or equal to emergency incident proximity
        - Haversine formula used to calculate distance between two points defined by latitude and longitude coordinates as follows:
            - $\Delta lat = lat2 - lat1$ $\Delta lon = lon2 - lon1$
            - $a = \sin2\ (\Delta lat\ /\ 2) + \cos(lat1) * \cos(lat2) * \sin2\ (\ \Delta lon\ /\ 2)$
            - $c = 2 * \operatorname{atan2}\ (\sqrt{a}, \sqrt{(1 - a)})$
            - $d = R * c$
    - Populating multiple search criteria should treat each search criteria as a required parameter (AND)
    - Leaving all search criteria blank will returns all potential results
    - Search results should be ordered in the following manner:
        - Distance (ascending)
        - Resource Name (alphabetically descending A to Z)

Requesting Resources

- A resource can't be used to respond to multiple incidents at the same time
- A resource can't be requested without associating an incident
- A resource in an available and in use status can be requested
- An expected return date must be included with the request

Resource Status

- A resource must be returned to the available status before being deployed
- A resource that has been deployed should have the status set to in use
- A resource can't be requested again for the same incident once returned back to an available status

Resource Report

- Only resources owned by the current user should be included

Task Decomposition (TD) & Abstract Code (AC)

Login

Task Decomposition

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

Abstract Code

- User populates *username* and *password* input fields
- If data validation is successful for both *username* and *password* input fields when ***Login*** button is clicked:
    - If User record is found but user password doesn't equal password:
        - Display an error message and return the User to the **Login** form
    - Else:
        - Store login information as session variable
        - Go to **Main Menu** form
- Else:
    - Display an error message and return the User to the **Login** form

Main Menu/Navigation Bar

Task Decomposition

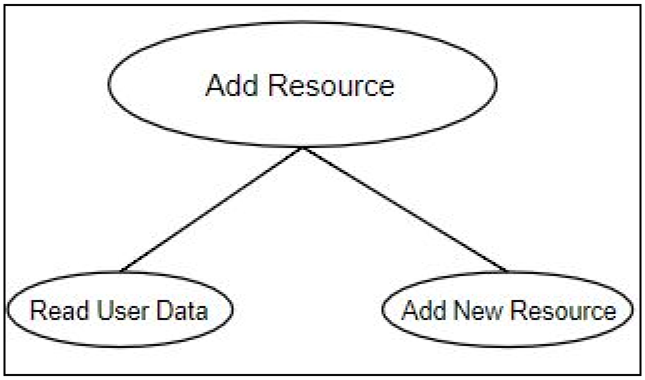
- Lock Types: Lookup User name and other details depending on user type, all are Read-Only
- Number of Locks: Single
- Enabling Conditions: Trigger by successful login.
- Frequency: Menu has more frequencies than successful logins
- Consistency (ACID): Not critical, order is not critical.
- Subtasks: Mother Task is not needed. No decomposition needed.

Abstract Code

- Show **"*Add Resource*", "*Add Emergency Incident*", "*Search Resources*", "*Resource Status*", "*Resource Report*", "*Exit*"** buttons
- Query information about the user name and other details depending on user type from the HTTP Session/Cookie
- Upon:
    - Click ***Add Resource*** button - Jump to the **Add Resource** task
    - Click ***Add Emergency Incident*** button **-** Jump to the **Add Incident** task
    - Click ***Search Resources*** button - Jump to the **Search for Resources** task
    - Click ***Resource Status*** button - Jump to the **View Resource Status** task
    - Click ***Resource Report*** button **-** Jump to the **Generate Resource Report** task
    - Click ***Exit*** button - close session and return the User to the **<u>Login</u>** form

Add Resource

Task Decomposition



- Lock Types: Lookup and Insert name for current User. Lookup list of ESF Categories. Insert Resource Name, Primary ESF, Additional ESFs, Model, Capabilities, Home Location, Cost/Cost Per, Maximum Distance, Status for a Resource.
- Number of Locks: Several different schema constructs are needed
- Enabling Conditions: Enabled by user's login and choosing ***Add Resource*** in main menu
- Frequency: Lookups have higher frequency than insertions
- Consistency (ACID): Consistency is not critical. The creation of the Resource ID must be handled consistently.
- Subtasks: Task is decomposed into **Read User Data** and **Add New Resource**

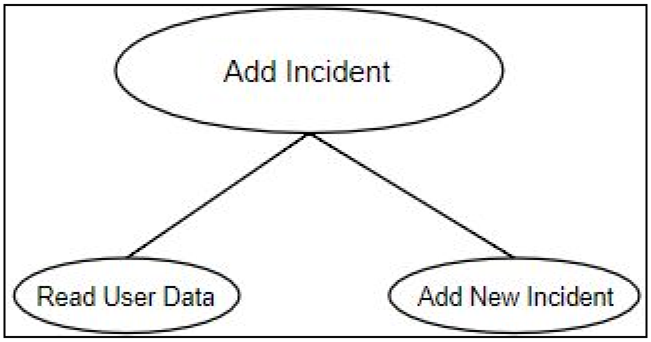
Abstract Code

- User clicks on ***Add Resource*** button from **Main Menu**
- Show *Resource ID, Owner, Resource Name, Primary ESF, Additional ESFs, Model, Capabilities, Home Location, Max Distance, and Cost* input fields
- Show ***Add, Cancel,*** and ***Save*** buttons
- Run the **Read User Data** subtask:
    - Query for information to populate the Owner field about the current user
    - Query for information to populate the *Primary ESF* field, a dropdown box that includes a list of preloaded ESF selection allowing for selection of 1 value
    - If field in *Primary ESF* has been selected:
        - Query for information to populate the *Additional ESFs* field, a dropdown box that includes a list of preloaded ESF selection allowing for selection of 0 or many values
            - When populating *Additional ESFs,* remove the *Primary ESF* value as an option from the *Additional ESFs* field
    - Query for information to populate the *Cost* field, a dropdown box includes list of preloaded selection for units of time
- Auto assign a unique numeric ID for for *Resource ID*
- Upon:
    - Click ***Add*** button
        - Add input from the *Capabilities* field into the *Capabilities* list
            - Allows for multiple text inputs
    - Click ***Cancel*** button
        - Exit out of **Resource Add** form and go back to **Main Menu**
    - Click ***Save*** button

- Validate all fields before storing resource to the database. Verify that all required fields are filled in, the dollar amount in *Cost* is not negative, *Max Distance* is not negative, and Latitude and Longitude for *Home Location* contain valid coordinates.
- If all fields are valid:
    - Run **Add New Resource** subtask to save resource
        - Update Resource
- Else:
    - Display warning message to User

Add Incident

Task Decomposition



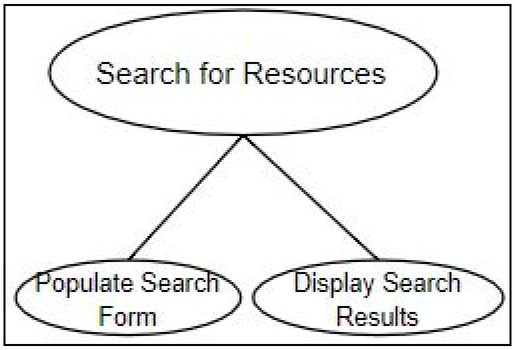
- Lock Types: Lookup and Insert name for a User. Lookup emergency types and Insert emergency type, date, description, and latitude/longitude for an Incident.
- Number of Locks: Several different schema constructs are needed
- Enabling Conditions: Enabled by user's login and choosing ***Add Incident*** in main menu
- Frequency: **Read User Data** has a higher frequency than **Add New Incident**
- Consistency (ACID): Consistency is not critical. It is only required for the creation of the unique Incident ID.
- Subtasks: Task is decomposed into **Read User Data** and **Add New Incident**

Abstract Code

- User clicked on ***Add Emergency Incident*** button from **<u>Main Menu</u>:**
- Show *Declaration, Incident ID, Date, Description, Location* input fields.
- Show ***Cancel,*** and ***Save*** buttons.
- Run the **Read User Data** subtask
  - Query for information to populate the Owner field about the current user (field does not appear on screen)
  - Query for the four types of incident declarations for the *Declaration* field dropdown.
- Auto assign an ID for the *Incident ID* based on declaration upon save.
- Upon:
  - Click ***Cancel*** button
    - Exit out of **<u>Add Incident</u>** form and go back to **<u>Main Menu</u>**
- Click ***Save*** button
  - Validate all fields before storing resource to the database. Verify that all required fields are filled in, *Date* is valid date, and Latitude and Longitude for *Home Location* contain valid coordinates.
  - If all fields are valid:
    - Run **Add New Incident** subtask to save incident
      - Update Incident
  - Else:
    - Display warning message to User

Search for Resources

Task Decomposition



- Lock Types: Lookup ESF categories, Incident Descriptions. Lookup ID, Name, Owner, Cost, Status, Next Available, and Distance for Resource.
- Number of Locks: Several different schema constructs are needed
- Enabling Conditions: Enabled by user's login and choosing ***Search Resources*** in main menu
- Frequency: Same frequency
- Consistency (ACID): Consistency is not critical
- Subtasks: Task is decomposed to **Populate Search Form** and **Display Search Results**

Abstract Code

- User clicked on ***Search Resources*** button from **<u>Main Menu</u>:**
- Show *Keyword, ESF, Location, Incident* input fields.
- Show ***Cancel,*** and ***Search*** buttons.
- For the *Location field* show **Up** and **Down** toggle buttons that increase or decrease the field value by 1
- Run the **Populate Search Form** subtask
  - Query for information to populate the *Primary ESF* field
  - Query for information to populate the *Incident* field with incidents created by the current user
- Upon:
  - Click ***Search*** button
    - For fields that have input, validate and verify fields before querying the database. *Location* is an integer.
    - Show **Close** button
    - If all search criteria is blank:
      - Run **Display Search Results** subtask and return all potential results
      - Display search results data elements *ID*, *Name*, Owner, Cost, and Status
    - Else:
      - Run **Display Search Results** subtask and return subset of dataset by returning results that meet all criteria populated:
        - Populating multiple search criteria should treat each search criteria as a required parameter (AND)
        - Matching substrings with *Keyword* (resource name, model or capabilities
        - Exact match with *ESF* (primary and additional ESF)
        - Less than or equal to *Location*

        - Haversine formula used to calculate distance between two points defined by latitude and longitude coordinates as follows:
          - $\Delta lat = lat2 - lat1$ $\Delta lon = lon2 - lon1$
          - $a = \text{sin2}\ (\Delta lat\ /\ 2) + \cos(lat1) * \cos(lat2) * \text{sin2}\ (\ \Delta lon\ /\ 2)$
          - $c = 2 * \text{atan2}\ (\surd a, \surd(1 - a))$
          - $d = R * c$
      - Equal to *Incident*
    - If the *Incident* field is not populated:
      - Display search results data elements *ID*, *Name*, Owner, *Cost*, and *Status*
    - Else:
      - Display additional columns *Distance*, and *Action*
      - If resource is owned by current User and not in use:
        - Display **Deploy** button
      - Else:
        - Display **Request** button
- Click ***Cancel*** button
  - Exits out of **Search for Resources** form and goes back to **Main Menu**

Request or Deploy Resource

Task Decomposition

- Lock Types: Update Availability and Request Status for Resource.
- Number of Locks: Few different schema constructs are needed
- Enabling Conditions: Enabled by user's login, choosing ***Search Resources*** in main menu and returning results from the **Search for Resources** task
- Frequency: Same Frequency
- Consistency (ACID): Consistency is not critical
- Subtasks: Mother Task is not needed. No decomposition needed.

Abstract Code

- If Deploy button exists and selected:
    - Run **Request or Deploy Resource** task to update the status of resource
    - Remove button after selection
- Else if Request button exists and selected:
    - Run **Request or Deploy Resource** task to create a request for resource
        - If the resource had previously been used for the same incident then don't allow the resource to be assigned to this incident in the future
    - Remove button after selection
- Upon:
    - Click ***Close*** button
        - Exits out of **<u>Search Results</u>** form and goes back to **<u>Main Menu</u>**

View Resource Status

Task Decomposition

- Lock Types: Lookup current User. Lookup ID, Name, Owner, Availability Status, Request Status, Start Date, and Expected Return Date for Resource. Lookup associated Incident for Resource.
- Number of Locks: Few different schema constructs are needed
- Enabling Conditions: Enabled by user's login and choosing ***Resource Status*** in main menu
- Frequency: Same frequency
- Consistency (ACID): Consistency is not critical
- Subtasks: Mother Task is not needed. No decomposition needed.

Abstract Code

- User clicks on ***Resource Status*** button from **<u>Main Menu</u>**
- Show ***Cancel*** button
- Run **View Resource Status** task
    - If resources are being used by current user:
        - Display a grid labeled Resources In Use associated to resources owned by the current User with the following data elements:
            - *ID*
            - *Resource Name*
            - *Incident*
            - *Owner*
            - *Start Date*
            - *Return By*
            - *Action*
        - Show ***Return*** button in *Action* column. If selected, run **Manage Resources** task (below)
    - If resources are requested by current user:
        - Display a grid labeled Resources Requested By Me associated to resources that have been requested by the current User with the following data elements:
            - *ID*
            - *Resource Name*
            - *Incident*
            - *Owner*
            - *Return By*
            - *Action*
        - Show ***Cancel*** button in *Action* column. If selected, run **Manage Resource** task (below)
    - If resource requests are received by current user:
        - Display a grid labeled Resource Requests Received By Me associated to resources that have been requested by other Users owned by the current User with the following data elements:
            - *ID*
            - *Resource Name*

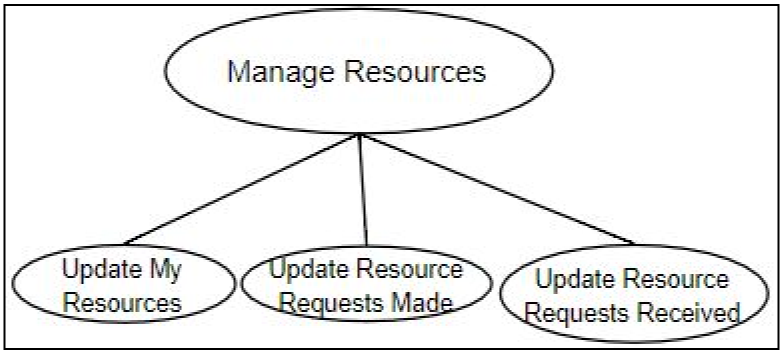
- *Incident*
- *Owner*
- *Return By*
- *Action*

- Show ***Reject*** button in *Action* column. If selected, run **Manage Resource** task (below)
- If resource requested is not currently in use:
    - Show ***Deploy*** button in addition to the ***Reject*** button in the *Action* column. If selected, run **Manage Resource** task (below)

- Upon:
    - Click ***Cancel*** button
        - Exit out of **Resource Status** form and go back to **Main Menu**

Manage Resources

Task Decomposition


Manage Resources
Update My Resources
Update Resource Requests Made
Update Resource Requests Received

- Lock Types: Update resources in use by current user, update resource requests received by current user, and update resources requested by current user
- Number of Locks: Several different schema constructs are needed
- Enabling Conditions: Enabled by user's login, choosing ***Resource Status*** in main menu and returning results from the **View Resource Status** task
- Frequency: Different frequencies for each action
- Consistency (ACID): Consistency is very important to make sure resources are updated correctly between multiple users
- Subtasks: Task should be decomposed into **Update My Resources, Update Resource Requests Made** and **Update Resource Requests Received**

Abstract Code

- If ***Return*** button selected in Resources in Use grid:
  - Run **Update My Resources** subtask to delete the request of a resource
    - Remove button after selection
- If ***Cancel*** button selected in Resources Requested by Me grid:
  - Run **Update Resource Requests Made** subtask to update the resource and request status
    - Remove button after selection
- If ***Action*** button selected in Resource Requests Received by Me grid:
  - Run **Update Resource Requests Received** subtask to update the request status
    - Remove button after selection
- Upon:
  - Click ***Cancel*** button
    - Exit out of <u>**Resources Status**</u> form and go back to <u>**Main Menu**</u>

Resource Report

Task Decomposition

- Lock Types: Lookup Primary ESF, and Status of Resources for all Resources owned by current User
- Number of Locks: Few different schema needed
- Enabling Conditions: Enabled by user's login and choosing ***Resource Report*** in main menu
- Frequency: Same frequency
- Consistency (ACID): Consistency is not important
- Subtasks: Mother Task is not needed. No decomposition needed.

Abstract Code

- User clicks on ***Resource Report*** button from **<u>Main Menu</u>**
- Show ***Cancel*** button
- Run **Generate Resource Report** task
    - Return row of data for each ESF with the following data elements (including elements with 0 items):
        - *ESF #*
        - *Primary Emergency Support Function*
        - *Total Resources*
            - Return count of resources owned by current User
        - *Resources In Use*
            - Return count of resources owned by current User in use
    - Return total aggregation row that sums the column values
- Upon:
    - Click ***Cancel*** button
        - Exit out of **<u>Resource Report</u>** form and go back to **<u>Main Menu</u>**