Phase 1 Report

Table of Contents

Data types

Users

Display Name	Varchar(20)	Not Null
Username	Varchar(12)	Not Null
Password	Varchar(12)	Not Null
Role	Varchar(20)	Not Null

CIMT

Username	Varchar(12)	Not Null
Phone Number	Varchar(10)	

[Back to Top](#)

Resource Provider

Username	Varchar(12)	Not Null
US Postal Address	Varchar(30)	
Street Address	Varchar(20)	
Address Number	Varchar(20)	
Street Name	Varchar(20)	
Apt Number	Varchar(5)	
City	Varchar(20)	
Zip Code	Varchar(5)	

Admin

Username	Varchar(12)	
Email Address	Varchar(20)	

Incidents

ID	Varchar(20)	Not Null
Description	Varchar(30)	Not Null
Date	Date	Not Null
Category	Varchar(30)	Not Null
Cat_ID	Varchar(2)	Not Null
Cat_Type	Varchar(30)	Not Null
User	Varchar(20)	Not Null

Resource

ID1	Varchar(20)	Not Null
Name	Varchar(20)	Not Null
Primary Function	Varchar(20)	Not Null
Secondary Function	Varchar(20)	
Descriptions	Varchar(30)	
Capabilities	Varchar(20)	
Distance	Double(10,1)	
Price	Varchar(20)	Not Null
Cost	Double(10,2)	Not Null
Unit	Varchar(20)	Not Null
User	Varchar(20)	Not Null

Business constraints

- A resource is available if it is not currently being used to respond to an incident.
- New resources entered into the system are available by default.
- In no circumstances should the system allow a resource that is currently in use be deployed to respond to another incident.
- The system should not allow the modification of resource information while it is in use.

- A resource should not be added if it is outside state lines or in a foreign country.
- All available resources should and will be dispatched and/or be in use within 24 hours.
- Incidents reported on CIMT should be either on campus or within a 1 mile radius of campus.
- All money values entered into the system will be considered in US Dollars such as the cost of resources.
- All possible emergencies should have at least one resource, regardless if it's available or not.
- Users residing outside the country cannot offer their resources unless approved by admin.

Markup Annotations

<u>Bold Underline</u>: <u>Form</u>.

Bold Italics: Buttons / Input Fields.

Bold*: Task.*

Italics: Form Input fields / Column names in tabulated form.

$XYZ: Database field/column named 'XYZ'.

Task decomposition with abstract code:

Login

<u>Task Decomposition:</u>

<u>Abstract Code:</u>

User enters *Username* ('$Username') and *Password* ('$Password') input fields. If

data validation is successful for both *Username* and *Password* input fields, then:
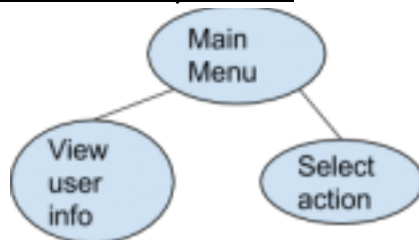
When **Login** button is clicked:

If *Username* exists but *Password* != '$Password', or *Username* is not

found: Go back to **Login** form, with error message

Else, log user in and go to **Main Menu** form.

Else, *Username* and *Password* input fields are invalid, display **Login** form, with error message.

Main Menu

<u>Task Decomposition:</u>

<u>Abstract Code:</u>

User successfully logged in from the **Login**

Run the **view user info** task by querying the user to display the user's '$Name' and user role ($name will stored in state):

If role = "cimt", show $name and then , select cimt table, show $phoneNumber.

If role = "resource_provider", show $name and then , select resource_provider table, $address

If type = "admin", show '$name and then , select admin table,$email

Run the **select action** task to display "Add Resource", "Add Emergency

Incident", "Search Resources", "Resource Status", and "Resource Report" links.

Upon clicking:

> ***Add Resource*** button - Jump to **Add Resource** task.
> ***Add Emergency Incident*** button - Jump to **Add Emergency Incident** task
> ***Search Resources*** button - Jump to **Search Resources** task.
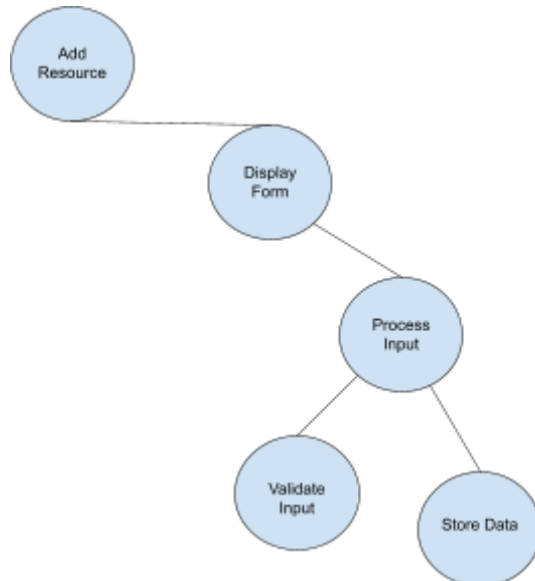> ~~***Resource Status*** button - Jump to **Resource Status** task.~~
> ***Resource Report*** button - Jump to **Generate Resource Report** task.
> ***Exit*** button - Invalidate login session and go back to **Login** form.

● Upon clicking "x" cross sign on **Main Menu** form:
> Exit the current form, and take user back to the **Login** form.

Add Resource Task

Task Decomposition:

Abstract Code:

User clicked Add Resource from the **Main Menu**

Run **Add Resource task** via HTTP GET request to run **Display Form task**

Add Resource Form will include the following input fields, buttons, and drop down menu

- *Plus Button* - discards current form and runs **Add Resource Task** again
- Resource ID($id) - field is blank, empty on load
- Owner - display user's $name
- *Resource Name*($name) **-** required field
- *Primary Function*($primary_function) **,** required field, a pull down menu with the following menu options (that are queried from the database table):
    - #1: transportation (this is the default selected item in the pull down menu)
    - #2: communications
    - #3: engineering
    - #4: search and rescue
    - #5: education
    - #6: energy
    - #7: firefighting
    - #8: human services
- *Secondary Function* ($secondary_function)- optional, displays same drop down menu from previous "Primary Function" minus the function already chosen (stored in $function state)
- *Description*($description) **-** optional, user-entered text field
- *Capabilities*($capabilities) **-** optional, user-entered text field
    - **Plus** Button - when pressed, saves the Capabilities input in a state, will display the previous input dynamically
- *Distance*($distance)**:** optional field, user input field (numbers only)
- *Cost / Unit:* required
    - Cost($cost), User inputted field that is numbers only
    - Units($units), dropdown menu containing "day", "hour", "use", will be queried from database table
- *Cancel button***:** this allows the user to exit the current view, returning to the Main Menu.
- *Save button***:** when the user clicks on this button, application will begin process input task

Process Input

- Starts with Validate Input task
    - Checks that required fields are not blank, display error message if they are
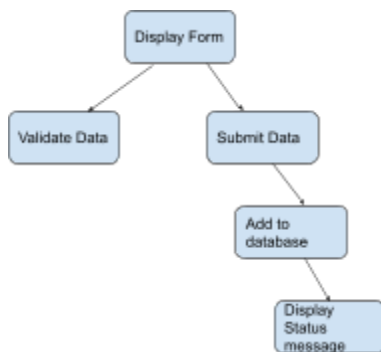   - Checks that input fields are the appropriate data (cost and distance

are numbers)
- After validation, begin store data task

Store Data

- Run query to store new resource into Resources table
  - Resource ID is auto-generated and stored in $id
  - Resource name is stored in $name
  - Primary Function - $primary_function (also increment the primary functions table)
  - Secondary Function - $secondary_function
  - Description - $description
  - Capabilities - $capabilities (store in capabilities table with primary key being the resource ID)
  - Distance - $distance
  - Cost - $cost
  - Unit - $unit
- Returns database generated Resource ID
- Submits request via HTTP POST
- If save data task is successful, display "Resource Saved" message

Add Emergency Incident Task

- Display Page
- New Incident Information as header
- *Plus Button* clears form/ jumps to a new add incident form
- Display following **Add Emergency Task Form**
  - **Category** (required) $category
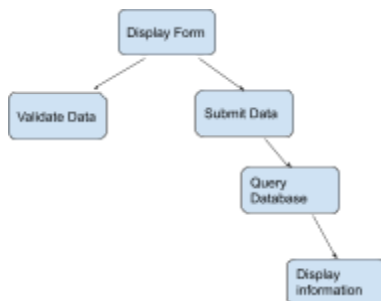   - Drop down menu displays messages
      - "must evac, secure lockdown"

- - - "may evac, secure lockdown"
      - "no evac, limited lockdown"
      - "no evac, no lockdown"
    - **Date** (required mm/dd/yyyy) $date
      - User provided date
      - Valid Format: accepts
      - Invalid Format: alerts user to try again
    - **Description** (required) $description
      - User provided text describing incident
- On *Save button* click
  - Checks that date and description have valid inputs
  - Generates number for ID
    - +1 one of last added into database
  - Adds to the database
    - Category
    - Incident ID
    - Date
    - Description
    - Owner
  - Submit request via HTTP POST
  - If successfully added, displays "Incident Saved" message
- On *Cancel button* click, brings back to webpage

Search Resources

Task Decomposition

Abstract Code

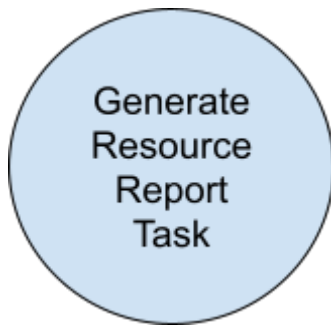- Display page
- "Search Resource" as heading
- ***Plus button*** clears form (basically the same as opening a new form)
- Form displayed next
  - **Keyword:** (optional) $keyword
    - User inputs text
  - **Primary Function** (required) $primary_function
    - comes with a pull down menu with the following menu options (that are queried from the database table):
      - #1: transportation (this is the default selected item in the pull down menu)
      - #2: communications
      - #3: engineering
      - #4: search and rescue
      - #5: education
      - #6: energy
      - #7: firefighting
      - #8: human services
  - **Distance** (optional) $distance
    - User inputs numbers
    - Trimmed to .1 of a mile
- On click of cancel button return to main menu
- On click of search button, queries database using given criteria
  - If no fields given, displays all resources
- Displays returned table under form

Generate Resource Report Task

Task Decomposition:

Abstract Code:

Generate Resource Report Task

Access resource report table within database

Query select all from table and store in an object

Display table on page with information provided by resource report table

While there are still rows in the object, populate page table with info.