Summary Phase 1 Report
Table of Contents

Data Types
A. Users:

Attribute	Data Type	Constraint
<u>Username</u>	String / Char	PK, NOT NULL
Password	String / VarChar	NOT NULL
DisplayName	String / Char	NOT NULL

➔ Cert_Member

Attribute	Data Type	Constraint
PhoneNumber	String / VarChar	NOT NULL

➔ Resource_Provider

Attribute	Data Type	Constraint
Number	Char	NOT NULL
Street	String / VarChar	NOT NULL
ApartmentNumber	Char	NULL

City	String / Char	NOT NULL
State	String / Char	NOT NULL

➔ Admin

Attribute	Data Type	Constraint
Email	String / VarChar	NOT NULL

B. Resources:

Attribute	Data Type	Constraint
Resource_id	String / VarChar	PK, NOT NULL
Resource_name	String / VarChar	NOT NULL
Description	String / Char	NOT NULL
Capabilities	String / Char	NULL
Distance	Decimal (3, 1)	NULL
Cost	Decimal (5, 2)	NOT NULL
Cost_unit	String / Char	NOT NULL
Primary_function	Integer	FK (Function_id), NOT NULL
Owner	String / Char	FK (Username), NOT NULL

➔ Function

Attribute	Data Type	Constraint
Function_id	Integer	PK, NOT NULL
Description	String / Char	NULL

C. Incidents:

Attribute	Data Type	Constraint
Incident_id	Int	PK, NOT NULL
Date	Date	NOT NULL
Description	String / VarChar	NOT NULL
Category	Char	FK (Category_id), NOT NULL

➔ Category

Attribute	Data Type	Constraint
Category_id	Int	PK, NOT NULL
Type	String / VarChar	NOT NULL

Business Logic Constraints (These are our assumptions based off of the given business requirements)

- 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.
- In no circumstances should the system allow a resource be deployed to respond to an incident if the resource cannot be delivered to the site of the incident in a reasonable time to assist.
- In no circumstances should the system allow a resource be deployed to respond to an incident if the resource cannot be realistically accessed due to scenarios; such as, the resource being damaged because of the incident, the owner not being present to give access to the resource, financial incapability, etc.
- A resource cannot be deployed to an incident that has already passed.
- A resource provider can only submit resources that they are realistically owners of, and have available to deploy at any time.
- A submitted resource must be in compliance with the law.
- The system does not account for SQL injections.
- User can only login from one machine at a time.

Markup Annotations

- **Bold Underline: Form / View.**
- **Bold Italics: Buttons**
- **Bold: Task.**
- Italics: Form Input fields / Column names in tabulated form.
- $XYZ: Database field/column named 'XYZ'.

# Task Decomposition with Abstract Code:

Login

Task Decomposition:

Abstract Code:

- 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:
        - *Username* and *Password* input fields are invalid. Go back to **Login** form and display error message.
    - Else, log user in and go to **Main Menu** view.

Main Menu

Task decomposition:
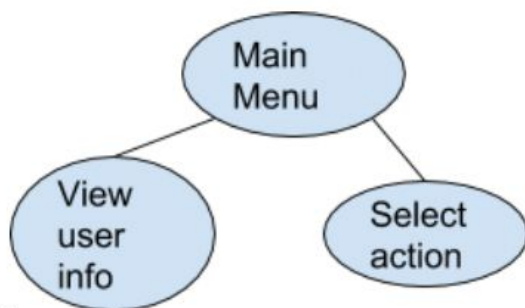
<u>Abstract Code:</u>
- User successfully logged in from the **<u>Login</u>** form
- Run the **view user info** task by querying the user to display the user's '$Name' and type:
  - If type = "municipality", show municipality's '$Category'.
  - If type = "government agent", show '$AgencyName'.
  - If type = "company", show '$Headquarter' (location of the headquarters) and '$NumberOfEmployees'.
- 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 **<u>Add Resource</u>** form.
  - ***Add Emergency Incident*** button - Jump to **<u>Add Emergency Incident</u>** form.
  - ***Search Resources*** button - Jump to **<u>Search Resources</u>** form.
  - ***Resource Status*** button - Jump to **<u>Resource Status</u>** view.
  - ***Resource Report*** button - Jump to **<u>Resource Report</u>** view.
  - ***Exit*** or "***X***" button - Invalidate login session and go back to **<u>Login</u>** form.
- When ***Exit*** button is clicked on **<u>Main Menu</u>** view:
  - Exit the current form, invalidate login session, and take user back to the **<u>Login</u>** form.

New Resource Information

<u>Task Decomposition:</u>

<u>Abstract Code:</u>
- User selects **add resource** task from **select action** task.
- Display **<u>Add Resource</u>** form, and run the **add resource** task.
- Display *Owner* ('$Owner') with the logged in user display name.

- User enters *Resource Name* ('$Resource_name'), *Primary Function* ('$Primary_function'), *Secondary Function* ('$Secondary_function'), *Description* ('$Description'), *Capabilities* ('$Capability'), *Distance* ('$Distance'), *Cost* ('$Cost'), and *Unit* ('$Unit') input fields, drop down menu, and list view:
    - *Primary Function* drop down menu:
        - Query ('**Function**') table -> ('$Function_id**'**) and ('$Description'), return and display *Primary Function* ('$Function_id') ('$Description').
    - If *Primary Function* drop down menu selected value is "value", then:
        - *Secondary Function* list view does not include "value"
- If data validation is not successful for all fields, then:
    - Display red error message instructing user to input correct format next to input field or drop down menu.
- Else if data validation is successful for all fields, then:
- When *Save* button is clicked:
    - Uniquely generate ('$Resource_id').
    - If *Resource Name*, *Primary Function*, *Cost*, or *Unit* is empty or not filled out:
        - Go back to **Add Resource** form, and display red error message next to required empty fields (error message: " * ").
    - Else, insert user-entered *values*.
- User adds new resource.
    - Run **display added resource** task: display user-entered fields as read-only.
- When "**+**" button is clicked:
    - Clear user entered values from input fields and drop down menus.
- When *Cancel* button is clicked:
    - Exit the current form, and take user back to the **Main Menu** form.

New Incident Information

Task Decomposition:

Abstract Code:

- User selects **add emergency incident** task from **select action** task.

- Display **Add Emergency Incident** form, and run the **add incident** task.

- User enters *Category* ('$Category'), *Date* ('$date'), *Description* ('$Description') input fields and drop down menu.

    - *Category* drop down menu:

        - Query ('**Category**') table -> ('Category_id') and ('$type'), return and display *Category* ('$type').

- If *Date* format is not "mm/dd/yyyy", then:

    - Display red error message instructing user to input correct format: mm/dd/yyyy

- Else if data validation is successful for all fields, then:

- When *Save* button is clicked:

    - Uniquely generate ('$Incident_id').

    - If *Category*, *Date*, and *Description* is empty or not filled out:

        - Go back to **Add Emergency Incident** form, and display error message next to required empty fields (error message: " * ").

    - Else, insert user-entered *values*.

- User adds new incident.

    - Run **display added incident** task: display user-entered fields as read-only.

- When "**+**" button is clicked:

    - Clear user entered values from input fields and drop down menus.

- When *Cancel* button is clicked:

    - Exit the current form, and take user back to the **Main Menu** form.

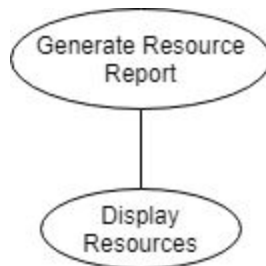# Search Resources

Task Decomposition:

Abstract Code:

- User selects **search resources** task from **select action** task.
- Display **Search Resources** form, and run the **search results** task.
- User enters *Category* ('$Category'), *Date* ('$Date'), *Incident* ('$Incident'), and *Distance* ('$Distance') input fields and drop down menu.
    - *Primary Function* drop down menu:
        - Query ('**Function**') table -> ('$Function_id**'**) and ('$Description').
- If data validation is not successful for all fields, then:
    - Display red error message instructing user to input correct format next to input field or drop down menu.
- Else if data validation is successful for all fields, then:
- When *Save* button is clicked, query the ('**Resource**') table to run a comparison:
    - If *Keyword* == ('$Resource_name') or ('$Description') or ('$Capability'), then:
        - Return ('$Resource_id'), ('$Resource_name'), ('$Owner'), ('$Cost'), ('$Unit'), ('$Distance'). Sort by resource's distance and display.
    - Else if "value" selected in *Primary Function*, then:
        - Return ('$Resource_id'), ('$Resource_name'), ('$Owner'), ('$Cost'), ('$Unit'), ('$Distance') where ('$Primary_function') == "value"
    - Else if "value" input in *Distance*, then:
        - Return ('$Resource_id'), ('$Resource_name'), ('$Owner'), ('$Cost'), ('$Unit'), ('$Distance') where ('$Distance') to 0.1 precision <= "value".
- Search returns results matching the user's search criteria.
    - Run **display search resources** task: display *Resource ID* ('$Resource_id'), *Resource Name* ('$Resource_name'), *Owner* ('$Owner'), *Cost/Unit* ('$Cost') ('$Unit'), *Distance* ('$Distance') in table list fashion.
- When "**+**" button is clicked:
    - Clear user entered values from input fields, drop down menus, and search results.
- When *Cancel* button is clicked:
    - Exit the current form, and take user back to the **Main Menu** form.
- User searches for resources using Keyword (optional).

Generate Resource Report

Task Decomposition:

Abstract Code:

- User selects **generate resource report** task from **select action** task.
- Displays the user's <u>**Resource Report**</u> form according to the each resource's primary function sorted by their Primary Function #.
  - Displays title "Resource Report" on top of the form.
- The table format of the report consists of:
  - *Primary Function # (PF#)*: pulled from *PF* in the database. *PF#* starts from 1 and goes up in ascending order.
  - *Primary Function (PF)*: a brief description of the *PF* in the database. Description of the *PF* is originally entered on <u>**Add Resource**</u> form which serves as a mandatory field.
  - *Total Resources*: row-wide aggregation by grouping all the resources belonging to each PF in the system for the user.
  - *Total*: column-wide aggregation by grouping all the resources across all PFs in the system.