

[Table of Contents](#)

[Log In](#)

[Main Menu](#)

[Adding a Resource](#)

[Adding an Incident](#)

[Searching for Resources](#)

[Resource Status](#)

[Resource Report](#)

[Exit](#)

Log In

Abstract Code

- Display at the top left hand box 'Login to ERMS'
- Display Box Title 'ERMS Emergency Resource Management System'
- Display *User Name* and *Password* fields
- User enters *username* (\$Username) and *password* (\$Password) input fields on the **LOG IN** form to database;
- User clicks Enter button:

```
SELECT password FROM 'ERMSUser' WHERE username = $Username;
```

- If User record is found but `ERMSUser.password != $Password`:
 - Go back to **LOG IN** form and return error message displaying "invalid username or password"
- Else:
 - Store login information as session variable \$Username
 - advance user to **MAIN MENU** form;
- Set \$Username = username.
- If user 'x' is pressed then window closes

Main Menu

Abstract Code

- Display "**Add a Resource**", "**Add an Incident**", "**Search for Resources**", "**Resource Status**", "**Resource Report**" tabs and *Exit* button on the **MAIN MENU** form;
- A user can:
 - Click **Add a Resource** button – Advance User to **ADD A RESOURCE** Screen.
 - Click **Add an Incident** button – Advance User to **ADD AN INCIDENT** Screen.
 - Click **Search for Resources** button – Advance User to **RESOURCE SEARCH** Screen.
 - Click **Resource Status** button– Advance User to **RESOURCE STATUS** Screen.
 - Click **Resource Report** button– Advance User to **RESOURCE REPORT** Screen

Phase 2 AC with SQL | CS 6400 - Summer 2018 | Team 02

- Click **Exit** button- Logs user out of the system, redirects to the **MAIN MENU**.
- Display username in the top right corner:

```
SELECT username FROM 'ERMSUser' WHERE username = $Username;
```

- Check if User is a *Municipality*:

```
Select 'ERMSUser'.username, 'Municipality'.category from 'ERMSUser' LEFT JOIN  
'Municipality' on 'ERMSUser'.username = Municipality.username WHERE  
'Municipality'.category is not NULL;
```

- Display category below username in the top right corner
- Else check if User is a government agency:

```
Select 'ERMSUser'.username, 'GovernmentAgency'.agency_name from 'ERMSUser' LEFT  
JOIN 'GovernmentAgency' on 'ERMSUser'.Username = 'GovernmentAgency'.Username  
WHERE 'GovernmentAgency'.agency_name is not NULL;
```

- Display agency_name below username in the top right corner
- Else check if User is a company:

```
Select 'ERMSUser'.username, 'Company'.hq_location, 'Company'.number_of_employees from  
'ERMSUser' LEFT JOIN 'Company' on 'ERMSUser'.username = 'Company'.username WHERE  
'Company'.hq_location AND 'Company'.number_of_employees is not NULL
```

- Display hq_location and number of employees below username in top right corner
- Display *Exit* button. If user clicks on it, current \$Username **logs out** of the system and returns to the **LOG IN** form screen.
- If 'x' is **pressed**, then window **closes**

- Display Box Title on the left 'ERMS'
- Display at the top left hand box 'Login to ERMS'

Adding a Resource

Abstract Code

- User clicks on the **Add a Resource** button from the MAIN MENU;
- User advances to the ADD RESOURCE screen;
- Display `$Username` on the *Owner* field:

```
SELECT username FROM 'User' WHERE username = $Username;
```

- Display ESF_number and ESF_description in *Primary ESF* field:

```
SELECT ESF_number, ESF_description FROM 'dimESF';
```

- Display ESF_number and ESF_description in *Additional ESF* field:

```
SELECT ESF_number, ESF_description FROM 'dimESF';
```

- Display dropdown menu with integer values in in *Max distance* and *Cost* fields
- Display time_unit in *Cost time unit* field:

```
SELECT time_unit FROM 'CostTimeUnits';
```

- User manually fills out *Resource Name* (`$resource_name`), *Model* (`$model`) fields by inputting string characters
- User manually fills *Latitude* (`$latitude`) and *Longitude* (`$longitude`) fields by inputting float values
- User fills *Capabilities* (`$capabilities`) field by manually inputting string characters in the field next to **Add** button and then clicking **Add** button
- User fills *Primary ESF* (`$primary_esf_number` + `$primary_esf_descriptiob`), *Additional ESF* (`$additional_esf_number` + `$additional_esf_number`) and *Cost option* (`$time_unit`) fields by clicking one of the available options from the dropdown menu
- User fills *Max Distance* and *Cost* fields by clicking on an integer in the arrow menu

Phase 2 AC with SQL | CS 6400 - Summer 2018 | Team 02

- Validate that all required fields are entered
 - *Resource Name, Primary ESF, Capabilities, Latitude, Longitude, Cost, Cost time unit* fields should all contain values.
 - If at least one required field has a missing value then error box appears stating 'Missing value, please fill all required fields'.
- Ensure that a non-negative integer in *Cost* has been entered
 - If it has not been entered correctly then an error box appears stating 'error, user must enter a non-negative number'
- Ensure that valid Longitude and Latitude coordinates have been entered
 - If it has not been entered, then an error box appears stating 'error'
- Generate next sequential resource id (\$resource_id)
- Insert new Resource and its accompanying attributes into the [ERMSResource](#) Table

```
INSERT INTO 'ERMSResource' (resource_id, resource_owner, resource_name,
primary_ESF_number, model, latitude, longitude, time_unit, dollars, max_distance, status)
VALUES ($resource_id, $Username,$resource_name, $primary_ESF_number, $model,
$latitude, $longitude, $time_unit, $cost, NULL, $max_distance, 'Available');
```

- Insert \$resource_id and \$capabilities into [Capabilities](#) table:

```
INSERT INTO 'Capabilities' (resource_id, capability) VALUES ($resource_id, $capabilities);
```

- Insert \$resource_id and \$additional_esf_number INTO [AdditionalESF](#) table:

```
INSERT INTO 'AdditionalESF' (resource_id, additional_esf_number) VALUES ($resource_id,
$additional_esf_number);
```

- Display at the top left hand box 'Add a New Resource' Display Box Title 'New Resource Info'
- If 'x' is pressed then window closes
- If **Cancel** button is pressed then user is returned to 'Main Menu' Page
- If '+' button is clicked a new 'Add a Resource' form appears
- When hovering over additional ESF selection a blue highlight will appear
- When 'Add' button is clicked string entered in the box adjacent to the left is added to the capabilities box
 - If there is no string in the adjacent box and 'Add' is clicked then nothing happens.

- Scroll Arrows appear in additional ESF's and Capabilities boxes
- Dropdown menu arrows are displayed in Capabilities and Hour Box

Adding an Incident

Abstract Code

- User clicks on the Add an Incident button from the MAIN MENU;
- User advances to the New Incident Screen
- Retrieve available set of selection options for incident_type from [IncidentType](#) and display in *Declaration* field with dropdown menu:

```
SELECT incident_type FROM 'IncidentType';
```

- User clicks on available option in *Declaration* (\$incident_type) field.
- System generates automatic unique Incident number (\$incident_number)
- User fills out *Date* (\$date), *Description* (\$description), *Latitude* (\$latitude) and *Longitude* (\$longitude) fields.
- User submits a new Incident by clicking the **Save** button.
- Upon Saving validate that all required fields contain valid values.
- Ensure that valid Longitude and Latitude coordinates have been entered.
- Ensure that correct date format (MMDDYYYY format), is entered
- If required fields are entered incorrectly display error message 'Must enter a correct value'
- Insert new 'Incident' and its accompanying attributes into [Incident](#)

```
INSERT INTO 'Incident' (incident_number, incident_type, incident_owner, incident_date, incident_description, latitude, longitude) VALUES ($incident_number, $incident_type, $Username, $date, $description, $latitude, $longitude);
```

- Display at the top left hand box 'New Incident'
- Display Box Title 'New Incident Info'
- If 'x' is pressed then window closes
- If **Cancel** button is pressed then user is returned to 'Main Menu' Page

Searching for Resources

Abstract Code

[Table of Contents](#)

Phase 2 AC with SQL | CS 6400 - Summer 2018 | Team 02

- User clicks on the **Search for Resources** button from the MAIN MENU.
 - User advances to the Search Resources screen.
 - Display at the top left hand box 'Search Resources' string.
 - Display Box Title 'Search Resource' string.
 - If 'x' is pressed, then window closes
 - If **Cancel** button is pressed, then user is returned to 'Main Menu' Page
 - Display dropdown menu arrows for ESF and Incident Boxes
-
- Display all ESF_number and accompanying ESF_description instances in the *ESF* field:

```
SELECT ESF_number, ESF_description FROM 'dimESF';
```

- Display integer arrow menu in *Location* field
- Display all incident_number, incident_type and incident_description instances in *Incident* field:

```
SELECT incident_number, incident_type, incident_description FROM 'Incident';
```

- User manually fills *Keyword* field (\$keyword)
- User selects option from *ESF*, *Location* and *Incident* fields
- User clicks on **Search** button and advances to **Search Results** table
- Display resources from the *ERMSResource* table matching only the entered values on the Search Form:

```
SELECT
    ERMSResource.resource_id as ID
  , ERMSResource.resource_name as Name
  , ERMSUser.name as Owner
  , (CONCAT("$",trim(cast(ERMSResource.dollars as CHAR(50))), "/"
,ERMSResource.time_unit)) as Cost
  , ERMSResource.status as Status
  , case when ERMSResource.status = 'AVAILABLE' then 'NOW' else
cast(Request.next_available as CHAR(10)) end as Next_Available
```

```
,
DISTANCE_FORMULA(Incident.latitude,ERMSResource.latitude,Incident.longitude,ERMSRe
source.longitude) as Distance
FROM
    ERMSResource
    INNER JOIN ERMSUser ON ERMSResource.resource_owner =
ERMSUser.username
    INNER JOIN Incident ON Incident.incident_number =$incident_number and
Incident.incident_type=$incident_type
    LEFT JOIN Request ON ERMSResource.resource_id = Request.resource_id
    LEFT JOIN Capabilities ON Capabilities.resource_id = ERMSResource.resource_id
WHERE
    (case when $KEYWORD is NULL then TRUE else UPPER(Capabilities.capability) like
'%UPPER($KEYWORD)%'
    OR UPPER(ERMSResource.model) like '%UPPER($KEYWORD)%' OR
UPPER(ERMSResource.resource_name) like '%UPPER($KEYWORD)%')
AND
( case when $ESF is NULL then TRUE else $ESF = ERMSResource.primary_ESF_number
OR $ESF in (select additional_ESF_number FROM AdditionalESF WHERE
AdditionalESF.resource_id = ERMSResource.resource_id) AND
    ( case when $RADIUS is NULL the TRUE else
DISTANCE_FORMULA(Incident.latitude,ERMSResource.latitude,Incident.longitude,ERMSRe
source.longitude) < $RADIUS);
```

- If user does not input any fields for the search, display all available resources.

```
SELECT
    ERMSResource.resource_id as ID
    , ERMSResource.resource_name as Name
    , ERMSUser.name as Owner
    , (CONCAT("$",trim(cast(ERMSResource.dollars as CHAR(50))), "/"
,ERMSResource.time_unit)) as Cost
    , ERMSResource.status as Status
    , case when ERMSResource.status = 'AVAILABLE' then 'NOW' else
cast(Request.next_available as CHAR(10)) end as Next_Available
```



```
FROM
    ERMSResource
    INNER JOIN ERMSUser ON ERMSResource.resource_owner =
ERMSUser.username
    LEFT JOIN Request ON ERMSResource.resource_id = Request.resource_id
    LEFT JOIN Capabilities ON Capabilities.resource_id = ERMSResource.resource_id
WHERE
    (case when $KEYWORD is NULL then TRUE else UPPER(Capabilities.capability) like
'%UPPER($KEYWORD)%'
    OR UPPER(ERMSResource.model) like '%UPPER($KEYWORD)%' OR
UPPER(ERMSResource.resource_name) like '%UPPER($KEYWORD)%')
AND
( case when $ESF is NULL then TRUE else $ESF = ERMSResource.primary_ESF_number
OR $ESF in (select additional_ESF_number FROM AdditionalESF WHERE
AdditionalESF.resource_id = ERMSResource.resource_id);
```

- ;
- Resources are displayed on Search Results Table
 - If user selects an incident from the SEARCH RESULTS Screen, retrieve Resource Location and available actions;
 - (Display “Request” for the available action if the resource is currently in use and “Deploy” if the resource is not currently in use)
 - If user selects “Request,” require the user to enter a Date for the estimated time when the resource will be returned;
 - If Location Float number is greater than Max Distance Attribute of Resource, do not retrieve this Resource in Search Results Table

Resource Status

Abstract Code

- User clicks on the **Resource Status** button from the MAIN MENU;
- User advances to Resource Status screen.
- Retrieve and display resources currently set to “in-use” status, below ‘Resource in use’ title :

```
SELECT
    ERMSResource.resource_id as ID
, ERMSResource.resource_name as Name
```

```
, Incident.incident_description as Incident
, ERMSUser.name as Owner
, Request.request_start_date as Start_date
, Request.exp_return_date as Return_By
FROM
    ERMSResource
INNER JOIN Request    ON ERMSResource.resource_id = Request.resource_id
INNER JOIN Incident   ON Incident.incident_number = Request.incident_number and
Incident.incident_type = Incident.incident_type
INNER JOIN ERMSUser   ON ERMSUser.username = ERMSResource.resource_owner
WHERE
    Request.incident_owner=$CurrentUsername AND
    Request.date_request_accepted is NOT NULL AND
    Request.actual_return_date IS NULL AND
    ERMSResource.status = 'IN USE' ;
```

- Display Action column to the right of return by with **Return** button displayed. If user clicks on return button then `ERMSResource.status = 'AVAILABLE'`,

- Retrieve and display resources below 'Resources Requested by Me' title:

```
SELECT
    ERMSResource.resource_id as ID
, ERMSResource.resource_name as Name
, Incident.incident_description as Incident
, ERMSUser.name as Owner
, Request.exp_return_date as Return_By
FROM
    ERMSResource
INNER JOIN Request    ON ERMSResource.resource_id = Request.resource_id
INNER JOIN Incident   ON Incident.incident_number = Request.incident_number and
Incident.incident_type = Incident.incident_type
INNER JOIN ERMSUser   ON ERMSUser.username = ERMSResource.resource_owner
```

```
WHERE
Request.incident_owner=$Username AND
Request.date_request_accepted is NULL;
```

- Retrieve and display resources below 'Resource Requests Received by Me'

```
SELECT
    ERMSResource.resource_id as ID
    , ERMSResource.resource_name as Name
    , Incident.incident_description as Incident
    , ERMSUser.name as Owner
    , Request.exp_return_date as Return_By
FROM
    ERMSResource
INNER JOIN Request    ON ERMSResource.resource_id = Request.resource_id
INNER JOIN Incident  ON Incident.incident_number = Request.incident_number and
Incident.incident_type = Incident.incident_type
INNER JOIN ERMSUser  ON ERMSUser.username = Incident.incident_owner
WHERE
    Request.resource_owner=$CurrentUsername AND
    Request.date_request_accepted is NULL;
```

Resource Report

Abstract Code

- User clicks on the Resource Report button from the MAIN MENU and advances to Resource Report Screen;
- Display Resource Report by Primary Support Function:

```
SELECT
    dimESF.ESF_number
    , dimESF.ESF_description
    , sum(case when ERMSResource.resource_owner= $Username then 1 else 0 end ) as
Total_Resources
    , sum(case when ERMSResource.resource_owner= $CurrentUsername and
ERMSResource.Status = 'IN USE' then 1 else 0 end ) as Resources_in_Use
```

```
from    dimESF
LEFT JOIN ERMSResource on ERMSResource.primary_ESF_number =
dimESF.ESF_number
group by 1,2
order by 1
;
```

Exit

Abstract Code

- User clicks on the Exit button from the MAIN MENU;
- Logs the current user out of the system;
- Redirects the system back to the LOGIN screen;