

TEAM 49: REPORT

(CS6400 PROJECT)

OCTOBER 16, 2016

TEAM 049



BY:

Syedmohammad Razavi (srazavi32)

Maziar Motahari (mmotahari3)

Andrew Keith (akeith30)

Zhenghang Gu (zgu64)

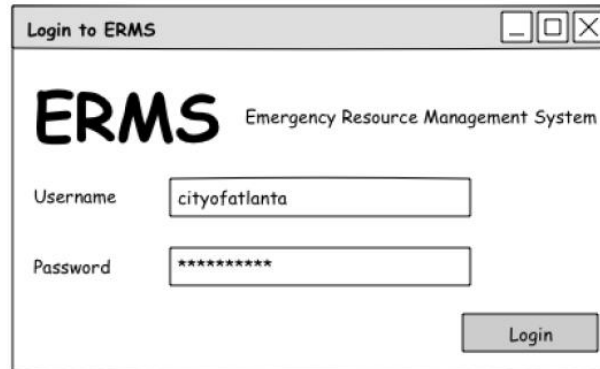
Phase 2 Abstract Code w/SQL
CS6400 – FALL 2016
TEAM 049

Contents

1- Login Task.....	2
2- MAIN MENU SELECTION TASK	3
3- Adding Resources to ERMS.....	4
4- Adding Emergency Incidents to ERMS.....	6
5- Searching for Resources	7
6- Resource Status	10
7- Actions	12
8- Report Generator.....	13

Phase 2 Abstract Code w/SQL
CS6400 – FALL 2016
TEAM 049

1- Login Task



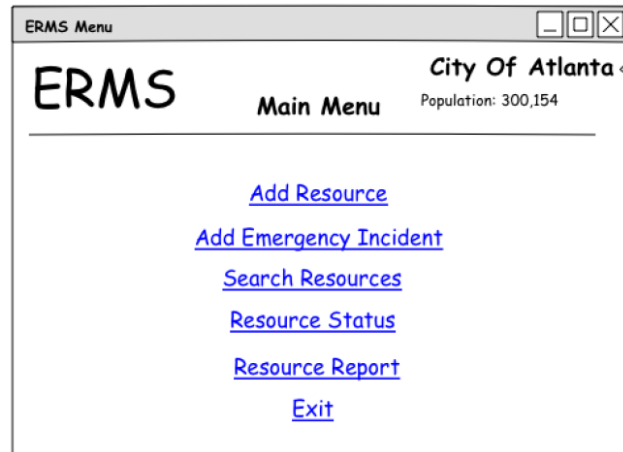
ABSTRACT CODE AND CONSTRAINTS:

- User enters username(@username1) and password(@pass1)
- If data validation is successful for both email and password input fields, then:
 - When Login button is clicked:

```
select * from users where USERNAME=@username1 and PASSWORD=@pass1;
```

- If username and password did not match any data in the database, display the error message form ("Username and Password Combination Does Not Exist!") and return to the login menu.
- If username and password found in the database, store user information as session variable @currentuser and show ERMS menu form

2- MAIN MENU SELECTION TASK



MAIN MENU FORM

ABSTRACT CODE AND CONSTRAINTS:

Find the current user using the username

Find name and display it on the specified location on the form. Assume @currentuser is passed to the form.

```
select NAME from users where USERNAME=@currentuser and PASSWORD=@pass1;
```

Display the following information depend on the type of user.

The population size, if the user is a municipality. If following query result is not empty show population.

```
select population from municipality_users where USERNAME=@currentuser;
```

The jurisdiction, if the user is a government agency.

```
select jurisdiction from govagency_users where USERNAME=@currentuser;
```

The location of the headquarters, if the user is a company.

```
select LOC_OF_HQ from company_users where USERNAME=@currentuser;
```

If "Add Resource" link is clicked, close current form and open "Add Resource" form

If "Add Emergency Incident" link is clicked, close current form and open "Add Incident" form

If "Search Resources" link is clicked, close current form and open "Search Resources" form

If "Resource Status" link is clicked, close current form and open "Resource Status" form

Phase 2 Abstract Code w/SQL
CS6400 – FALL 2016
TEAM 049

If “Resource Report” link is clicked, close current form and open “Resource Report” form

If “Exit” link is clicked, close current form and logout

3- Adding Resources to ERMS

The screenshot shows a web form titled "Add New Resource" with a sub-header "New Resource Info". The form contains the following fields and annotations:

- Resource ID:** 1021156. Annotation: "Auto Assigned unique numeric ID".
- Owner:** City of Atlanta. Annotation: "Owner set by default".
- Resource Name:** 2015 Hummer.
- Primary ESF:** (#13) Public Safety and Security. Annotation: "Each resource must have a single primary Emergency Support Function (ESF)".
- Additional ESFs:** A list box containing (#1) Transportation, (#2) Communications, (#3) Public Works and Engineering, and (#4) Fire Fighting. Annotation: "Zero or more additional ESFs".
- Model:** 2015 Hummer.
- Capabilities:** A list box containing Patrolling and OnBoard Computer.
- Home Location:** GPS. An "add" button is next to it. Annotation: "Any no. of capabilities can be added to the resource".
- Cost:** \$ text per day. Annotation: "Cost options are: Hour, Day and Week but the list should be extensible without changing code."

At the bottom are "Cancel" and "Save" buttons.

New resource form

ABSTRACT CODE AND CONSTRAINTS:

- Fill the primary ESF dropdown. UI should allow for unique selection.
- Fill the additional ESF dropdown. UI should allow for multiple selection.
 - For both above steps the following SQL query is used:

```
select ESF_ID, DESCRIPTION from ESF;
```

- Selected Primary ESF number is stored an integer in @P_ESF1
- Selected Additional ESF numbers are stored in an integer list @A_ESFs1
- The primary ESF of a resource cannot appear as an additional ESF

Phase 2 Abstract Code w/SQL
CS6400 – FALL 2016
TEAM 049

- Fill the Cost_Per dropdown using below Query.

```
select COSTPER from COST_PER;
```

- Find name and display it on the specified location on the form. Assume @currentuser is passed to the form.

```
select NAME from users where USERNAME=@currentuser and PASSWORD=@pass1;
```

- If data validation is successful for all input fields, then:
- When Save button is clicked.
- Read and store all input information in variables below:
(@Name1, @P_ESF1, @M_NAME1, @AMOUNT1, @COST_PER1, @LATITUDE1, @LONGITUDE1, @CAPABILITIES1, @A_ESF1)
- Resource ID will be automatically generated.
- Default Date_AV value is the current date.
- Default status is 'Available'.

```
insert into resource  
(USERNAME, NAME, P_ESF, M_NAME, AMOUNT, COST_PER, STATUS, DATE_AV, LATITUDE, LONGITUDE)  
values (@currentuser, @Name1, @P_ESF1, @M_NAME1, @AMOUNT1, @COST_PER1,  
'Available', CURDATE(), @LATITUDE1, @LONGITUDE1);
```

- After numerical ID is assigned show it on the form and stored as @Res_ID1.
- For each capability in @CAPABILITY1 list:

```
insert into resource_capability (RES_ID, CAPABILITY)  
values (@Res_ID1, @Capability1);
```

- For each capability in @CAPABILITY1 list:

```
insert into resource_capability (RES_ID, CAPABILITY)  
values (@Res_ID1, @Capability1);
```

- Owner is set automatically to the logged in user
- Primary ESF should be selected from one of the 15 predefined ESFs
- Each resource has only one Primary ESF
- Resources can have zero, one, or multiple additional ESFs
- The primary ESF of a resource cannot appear as an additional ESF
- Model field is optional
- Capabilities: zero or more
- Home Location: Latitude and Longitude

Phase 2 Abstract Code w/SQL
CS6400 – FALL 2016
TEAM 049

- The cost is stored in dollars per hour/day/week/each. This list should be extensible without requiring code changes
- Latitude and Longitude: Sign decimal degrees
- On clicking add capacities, add the new capability to the capabilities list and refresh the capability list on the display.
- On clicking cancel button
 - o Close the add resources form
 - o Show the main menu
- On save button
 - o Check if all information is in correct data format.
 - o Save record to the resource table
 - o Close the add resource form
 - o Show the main menu

4- Adding Emergency Incidents to ERMS

New Incident

New Incident Info

Incident ID (assigned automatically) Auto Assigned Unique Numeric ID upon save

Date 04/24/2016

Description Flash Floods in Fulton County

Location Lat 33.684 Lng -86.224 Choose center if the incident is widespread

Cancel Save

ADD NEW INCIDENT FORM

ABSTRACT CODE AND CONSTRAINTS:

- Each incident is automatically assigned a **unique numerical ID** on save
- The **owner** of the incident is automatically set to the current user
- All fields are required and should be validated before saving the incident to the database

Phase 2 Abstract Code w/SQL
CS6400 – FALL 2016
TEAM 049

- All incidents are private to the current user
- On clicking cancel button
 - o Close the add incident form
 - o Show the main menu
- On save button
 - o Check if all information is in correct data format.
 - o Read and store all input information in variables below:
(@DESCRIPTION1,@LATITUDE1,@LONGITUDE1)

```
insert into incidents (USERNAME,DESCRIPTION,LATITUDE, LONGITUDE)
values (@currentuser, @DESCRIPTION1, @LATITUDE1, @LONGITUDE1);
```

- o Close the add incident form
- o Show the main menu

5- Searching for Resources

The screenshot shows a 'Search Resources' dialog box with the following fields and callouts:

- Keyword:** A text input field containing 'Engine'. A callout bubble points to it with the text: 'Searches name, model and capabilities fields of resources'.
- ESF:** A dropdown menu showing '(#4) Fire Fighting'. A callout bubble points to it with the text: 'Search both primary and secondary ESF'.
- Location:** A text input field containing 'Within 15 Kilometers of incident'.
- Incident:** A dropdown menu showing '(100) Flash Floods in Fulton County'.
- Buttons:** 'Cancel' and 'Search' buttons. A callout bubble points to the 'Search' button with the text: 'Return everything when all fields are left blank'.

Search for Resources

Phase 2 Abstract Code w/SQL
CS6400 – FALL 2016
TEAM 049

Search Results

Search Results for Incident:
Flash Floods in Fulton County (102)

ID	Name	Owner	Cost	Status	Next Available	Distance	Action
12	Rescue Boat	Decatur Fire Dept.	\$200/hour	IN REPAIR	09/01/2016	0.9 km	
16	Helicopter	City Of Atlanta	\$150/hour	AVAILABLE	NOW	1.5 km	Deploy Repair
90	500 Ton Crane	John Doe	\$1200/day	NOT AVAILABLE	09/05/2016	3 km	Request
50	Diving Gear	Jane Doe	\$60/day	AVAILABLE	NOW	5 km	Request

Close

Callouts:

- Display incident description and ID if incident was selected
- Distance, Action show only if incident was selected. Sort by distance, then availability and then name
- Resource owned and not currently used can be immediately deployed
- Resources in repair can not be requested

Search Results

ABSTRACT CODE AND CONSTRAINTS:

- User may search by keyword (which includes the resource name, model, and capabilities)
- User may search by Emergency Support Function (which includes both primary and additional ESFs),
- User may search by proximity to an emergency incident.
- It is possible that the user will leave all search fields blank
- Read incidents from incident table and show them in a dropdown selection list

```
select INC_ID, DESCRIPTION from INCIDENTS where username=@currentuser;
```

- If cancel button is pushed, close window and display main menu
- If the search button is pushed do the following steps
- Show ESF list in the dropdown selection list

```
select ESF_ID, DESCRIPTION from ESF;
```

- When Search button is clicked.
- Read and store all input information in variables below:
(@Keyword, @ESF1, @Distance1, @INC_ID1)
- If an incident is selected read location of the incident to calculate distance of the incident to resources.

```
select LATITUDE, LONGITUDE from INCIDENTS where INC_ID=@INC_ID1;
```

This values are stored as @IncLat, @IncLong.

- If no field is selected

```
select a.ID, a.NAME, b.NAME, a.AMOUNT, a.COST_PER, a.STATUS, a.DATE_AV from resource a
inner join users b on a.USERNAME=b.USERNAME
```

Phase 2 Abstract Code w/SQL
CS6400 – FALL 2016
TEAM 049

- Search tables by KEYWORD

```
select a.ID, a.NAME, b.NAME, a.AMOUNT, a.COST_PER, a.STATUS, a.DATE_AV from resource a
inner join users b on a.USERNAME=b.USERNAME
left outer join resource_capability c on a.ID=c.RES_ID
where a.name like @keyword
or a.M_NAME like @keyword
or c.CAPABILITY like @keyword
```

- Search resource table by ESF assume the input ESF_ID is stored as ESFid1

```
select a.ID, a.NAME, b.NAME, a.AMOUNT, a.COST_PER, a.STATUS, a.DATE_AV from resource a
inner join users b on a.USERNAME=b.USERNAME
left outer join additional_esf c on a.ID=c.RES_ID
where a.P ESF = @ESFid1 or c.ESF ID=@ESFid1;
```

- IF incident is selected search resource table by LOCATION will be activated
- Assume function Distance (lat1, long1, lat2, long2) can calculate the distance between two locations 1 and 2 based on their latitude and longitude using *haversine* formul

```
select a.ID, a.NAME, b.NAME, a.AMOUNT, a.COST_PER, a.STATUS, a.DATE_AV,
Distance(a.LATITUDE, a.LONGITUDE, @IncLat, @IncLong) from resource a
inner join users b on a.USERNAME=b.USERNAME
where Distance< @Distance1
order by Distance asc
```

- Depend on how many information fields are provided by the user the intersection of above queries will be displayed as the search results
- Display results in show results from using queries provided above
- Display incident description if an incident was selected

```
select INC_ID, DESCRIPTION from INCIDENTS where INC_ID=@INC_ID1;
```

- Display search results for resource ID, Name, Owner, Cost, Status, Next Available Date
- Display distances only if an incident was selected
- Depends on the status of resource the action options will change

```
select status from RESOURCES where RES_ID=@RES_ID1;
```

- Resources in repair have no action
- Resources in use or available can be requested
- Resources owned and available can be deployed or be scheduled for repair

Phase 2 Abstract Code w/SQL
CS6400 – FALL 2016
TEAM 049

6- Resource Status

Resource Status

Resource Status

Resources in use

Id	Resource Name	Incident	Owner	Start Date	Return by	Action
6	All Terrain Vehicle	North GA Landslide	City Of Atlanta	08/05/2016	09/05/2016	<button>Return</button>
18	Ambulance	North GA Landslide	Grady's	09/01/2016	09/04/2016	<button>Return</button>
14	Gasoline Generator	Midtown Power Outage	John Doe	09/01/2016	09/01/2016	<button>Return</button>

Resources Requested by me

Id	Resource Name	Incident	Owner	Return by	Action
8	Life Jackets	Flash Floods in Fulton County	John Doe	09/05/2016	<button>Cancel</button>

Resource Requests received by me

Id	Resource Name	Incident	Requested By	Return by	Action
29	Snow Ploughs	Heavy snow in North GA	John Doe	09/05/2016	<button>Deploy</button> <button>Reject</button>
6	All Terrain Vehicle	Midtown Building Collapse	City Of Atlanta	09/10/2016	<button>Reject</button>

Repairs Scheduled/In-progress

Id	Resource Name	Start on	Ready by	Action
6	All Terrain Vehicle	09/06/2016	09/10/2016	<button>Cancel</button>

Can not deploy since it is in use

Scheduled for a repair of 5 days on return
Can not cancel if repair has already begun

ABSTRACT CODE AND CONSTRAINTS:

- Search and list all resources that are currently in use responding to any incidents owned by the current user. For each resource display the **ID**, **name**, **incident responding to**, and **owner**.

Phase 2 Abstract Code w/SQL
CS6400 – FALL 2016
TEAM 049

```
select a.RES_ID as 'ID',b.NAME as 'Resource Name',c.DESCRPTION as  
'Incident',d.NAME as 'Owner', a.APP_DATE as 'Start Date', a.RET_DATE as 'Return  
By'  
from requests a  
join resource b  
on a.RES_ID=b.ID  
join incidents c  
on a.INC_ID=c.INC_ID  
join users d  
on b.USERNAME=d.USERNAME  
where c.USERNAME=@currentuser and a.status='Approved';
```

- Search and list resource requests that have been sent by the current user (to another user) but have not yet been responded to

```
select a.RES_ID as 'ID',b.NAME as 'Resource Name',c.DESCRPTION as  
'Incident',d.NAME as 'Owner', a.RET_DATE as 'Return By'  
from requests a  
join resource b  
on a.RES_ID=b.ID  
join incidents c  
on a.INC_ID=c.INC_ID  
join users d  
on b.USERNAME=d.USERNAME  
where c.USERNAME=@currentuser and a.status='Pending';
```

- Search and list resource requests received by the current user that are awaiting the user's response

```
select a.RES_ID as 'ID',b.NAME as 'Resource Name',c.DESCRPTION as  
'Incident',d.NAME as 'Requested By', a.RET_DATE as 'Return By'  
from requests a  
join resource b  
on a.RES_ID=b.ID  
join incidents c  
on a.INC_ID=c.INC_ID  
join users d  
on c.USERNAME=d.USERNAME  
where b.USERNAME=@currentuser and a.status='Pending';
```

- Search and list all resources that have a repair schedule request or are under repair

```
Select ID as 'ID',NULL as 'START_DATE',DATE_AV as'Ready by',STATUS from resource  
where status="In-Repair" and USERNAME=@currentuser  
union  
select RES_ID as 'ID',START_DATE,READY_DATE as'Ready by',"REPAIR_SCHEDULED" as  
'STATUS' from rep requests;
```

7- Actions

- A given resource cannot be used to respond to multiple incidences at the same time
- Repairs can be scheduled by the owner.
- A resource can be requested directly from the search results screen only when the user selected an incident on the search criteria form
- Resource requests are always made within the context of an incident
- Both available and in use resources may be requested
- Multiple requests may queue up for the same resource
- A resource may only be deployed to respond to one incident at a time
- Resources currently being repaired cannot be requested
- When Resources are requested, an expected return date must be added
- Repair requests should be of the form “for N days after return” for *In Use* resources and “for next N days” for *available* resources
- The *Next Available* date should take into account the requested repair
- If the repair duration has already begun, the repair cannot be cancelled and hence the resource cannot be deployed/requested.
- If the duration is yet to begin, the owner can accept requests for it but will have to explicitly cancel the repair request
- Once a resource has been returned back to available status, the system should prevent the same resource from being requested again for the same incident.
- The system should allow the user to *cancel* any pending resource requests.
- If a resource is currently in use, the deploy button should be disabled
- **Deploy** the resource: When Button Deploy is pressed. Resource **status** will be set to in use. The request will be removed from the requests Table.

```
DELETE from requests where RES_ID=@RES_ID1 and INC_ID=@INC_ID1;

update resource
set STATUS='IN-USE' and DATE_AV=$DATE_AV where ID=@RES_ID1;
```

- **Return** Resource: If there is a Repair request the status will change to In-Repair and Next available time will be calculated accordingly; otherwise, it changes the status of the resource to available change next available date to current date.

```
select START_DATE, RETURN_DATE from rep_request where RES_ID=@RES_ID1;
```

If above query returns results, then

```
update resource
set STATUS='IN-REPAIR' and DATE_AV=@RETURN_DATE1 where ID=@RES_ID1;
```

Else

```
update resource
set STATUS='AVAILABLE' and DATE_AV=$curdate() where ID=@RES_ID1;
```

Phase 2 Abstract Code w/SQL
CS6400 – FALL 2016
TEAM 049

- *Cancel* any pending resource requests: the request will disappear from the request table.

```
DELETE from requests where RES_ID=@RES_ID1 and INC_ID=@INC_ID1;
```

- *Reject* a request: the pending request will be removed from requests table.

```
DELETE from requests where RES_ID=@RES_ID1 and INC_ID=@INC_ID1;
```

- *Repair*: Is requested by owner. A Repair request form is displayed and user enters the expected number of days for repair. If the status is "AVAILABLE" it will change to "IN-REPAIR" and the request will not be recorded. Otherwise, it will be recorded in the rep_request table.

- If the status is "IN-USE"

```
INSERT INTO rep_request values (@RES_ID1,@start_date1,@return_date1)
```

- If the status is "AVAILABLE"

```
update resource
set STATUS='IN-REPAIR' and DATE_AV=@RETURN_DATE1 where ID=@RES_ID1;
```

- *Request*:

```
INSERT INTO requests values (@RES_ID1,@ INC_ID1, @REQ_DATE1, @RET_DATE1)
```

8- Report Generator

Resource Report			
Resource Report by Primary Emergency Support Function			
#	Primary Emergency Support Function	Total Resources	Resources in Use
1	Transportation	5	2
2	Communications	8	7
3	Public Works and Engineering	2	0
4	Firefighting	4	0
5	Emergency Management	12	0
6	Mass Care, Emergency Assistance, Housing, and Human Services	1	1
7	Logistics Management and Resource Support	0	0
8	Public Health and Medical Services	0	0
9	Search and Rescue	10	1
10	Oil and Hazardous Materials Response	0	0
11	Agriculture and Natural Resources	0	0
12	Energy	2	2
13	Public Safety and Security	14	3
14	Long-Term Community Recovery	0	0
15	External Affairs	0	0
	TOTALS	58	16

This report should only consider the primary ESF for each resource and ignore the additional ESFs field

Phase 2 Abstract Code w/SQL
CS6400 – FALL 2016
TEAM 049

Only resources owned by the current user should be counted for the Total Resources column.

The Resources in Use column counts the total number of resources with the given ESF that are currently in use

All ESFs should be shown, even if the user owns no resources for that ESF.

```
Select a.ESF_ID, a.DESCRPTION, count(b.ID) as 'Total Resources', count(c.ID) as  
'Resources in use'  
from ESF a  
left outer join resource b  
on a.ESF_ID=b.P_ESF and b.USERNAME=@currentuser  
left outer join resource c  
on a.ESF_ID=c.P_ESF and c.status='In-USE' and c.USERNAME=@currentuser  
group by a.ESF ID, a.DESCRPTION;
```