Phase 2 Abstract Code w/SQL
CS 6400 – Fall 2016
Team 028

Login

Abstract Code

- User enters *username* ($Username) and *password* ($Password_s).
- If data validation is successful for both *username* and *password*, then:
- While TRUE:
  - If ***Login*** button is clicked:

```
SELECT * FROM USERS
WHERE Username=$Username AND Password_s=$Password_s;
```

    - If no record is found:
      - Go back to **Login** form with error message.
    - Else:
      - Go to **Main Menu** form.
  - Else:
    - Do nothing.
- Else either *username* or *password* are invalid, display **Login** form, with error message

Main Menu

Abstract Code

- Display menu options.
- *Username* ($Username) is already provided when login.
- Read user name and information given user types

```
SELECT USERS.Name_s, COMPANY.headquater
From USERS INNER JOIN COMPANY ON USERS.Username=COMPANY.Username
Where Username=$Username;
```

  - If no record is found

```
SELECT USERS.Name_s, GOV_AGENCY.Jurisdiction
From USERs INNER JOIN GOV_AGENCY ON
USERS.username=GOV_AGENCY.Username
WHERE Username = $Username;
```

    - If no record is found

```
SELECT USERS.Name_s, MUNICIPALITY.Population_size
From USERS INNER JOIN MUNICIPALITY ON
USERS.username=MUNICIPALITY.username
WHERE Username=$Username;
```

    - Display what is found.
- If ***Add Resource*** button is clicked:
    - Go to **Add Resource** form.
- Else if ***Add Emergency*** Incident button is clicked:
    - Go to **Add Emergency** form.
- Else if ***Search Resources*** button is clicked:
    - Go to **Search Resources** form.
- Else if ***Resource Status*** button is clicked:
    - Go to **Resource Status** form.
- Else if ***Resource Report*** button is clicked:
    - Go to **Resource Report** form
- Else if ***Exit*** button is clicked:
    - Exit program

Add Resource

Abstract Code

- Automatically generate *resource ID* ($Res_ID).
- Load ESF list and Cost-Per list.

```
SELECT ESF_ID_Desc FROM ESF

SELECT Unit FROM Cost_unit
```

- Display *resource ID* ($Res_ID), *username* ($Username), and other blank fields.
- While TRUE:
    - Else if Save button is clicked:
        - If input fields, including *resource name* ($Res_name), *primary ESF* ($Prim_ESF), *additional ESFs* ($ADDI_ESFs), *model* ($Model), *capabilities* ($Capability), *home lat* ($Home_lat), *home long* ($Home_long), *pirce* ($Price), and *cost per unit* ($Unit), are all valid:

```
for each $Res_name, $ID_desc, $model, $Home_lat, $Home_long, $Status_s, $Price, $Unit

INSERT INTO RESOURCE (Username, Res_ID, Res_name, Prim_ESF, Model, Home_lat, Home_long, Status_s, Price, Cost_unit)

        VALUES ($username, $Res_ID, $Res_name, $Prim_ESF, $Model, $Home_lat, $Home_long, $Status_s, $Price, $Unit)

end for

for each $Res_ID, $Capability

INSERT INTO RES_CAP (Res_ID, Res_capability)

        VALUES($Res_ID, $Capability)

end for

for each $Res_ID, $ADDI_ESFs

INSERT INTO RES_ADDI_ESF (Res_ID, ID_Desc)

        VALUES($Res_ID, $ADDI_ESFs)

end for
```

            - Break While.
        - Else show invalid fields and ask for new valid inputs.
    - Else if Cancel button is clicked:
        - Void the auto-generated *resource ID* ($Res_ID).
        - Break While.
    - Else:
        - Do nothing.
- Go back to **Main Menu** form.

Add Incident

Abstract Code

- Automatically generate *Incident ID* ($Inc_ID).
- Display *Incident ID* ($Inc_ID) and other blank fields.
- While TRUE:
    - If *Save* button is clicked
        - If input fields, including *date* ($Date_s), *description* ($Description), *longitude* ($Longitude), and *latitude* ($Latitute), are all valid:

```
for each $Date, $Description, $Longitude, $Latitude

INSERT INTO INCIDENT (Username, Inc_ID, Date_s, Description, Longitude, Latitude)

VALUES ($Username, $Inc_ID, $Date_s, $Description, $Longitude, $Latitude)

end for
```

            - Break While.
        - Else show invalid fields.
    - Else if *Cancel* button is clicked:
        - Void the auto-generated *resource ID* ($Res_ID).
        - Break While.
    - Else:
        - Do nothing.
- Go back to **Main Menu** form.

Search Rescources

Abstract Code

- Load ESF list and incident list.

```
SELECT ESF_ID_Desc FROM ESF

SELECT Inc_ID, Description FROM INCIDENT
```

- While TRUE:
    - If *Search* button is clicked:
        - If input fields, *keyword* ($Keyword), *ESF* ($ESF_ID_Desc), *distance* ($Distance), and *incident* ($Inc_ID, $Description), are all valid:

```
SELECT R.Res_ID, R.Res_name, R.Username, R.Price, R.Cost_unit,
R.Status_s, Distance

FROM RESOURCES AS R INNER JOIN RES_CAP ON

R.Res_ID=RES_CAP.Res_ID

WHERE (Res_name LIKE $Keyword OR Model LIKE $Keyword OR
Res_capability LIKE $Keyword) AND (ESF_ID_desc LIKE $ESF_ID_Desc) AND
(Distance<=$Distance)

ORDER BY Distance ASC;
```

        - Break While
        - Go to **Search Results** form.
    - Else show invalid fields.
- Else if Cancel button clicked:
    - Break.
- Else:
    - Do nothing.

- Go back to **Main Menu** form.

Distance is calculated by application through *haversine()* formula

Search Results (Deploy/Repair/Request)

Abstract Code

- Display the *name* ($Description) and *ID* ($Inc_ID) of the incident searched.
- Display the *ID* ($Res_ID), name ($Res_name), owner ($Username), cost ($Price, $Cost_unit), and status ($Status_s), distance (Distance) of the returned resources which satisfy the search conditions, sorted by the distance (Distance).
- While TRUE:
    - If ***Deploy*** button is clicked: Update resource; Input expected return date;

```
for each $Status_s

UPDATE RESOURCES

SET Status_s='NOT AVAILABLE'

WHERE Res_name=$Res_name AND Res_ID=$Res_ID

end for
```

    - Else if ***Repair*** button is clicked: Update resource; Display **Repair** form; Input repair *start date* ($Start_date) and *available date* ($Return_by)

```
for each $Status_s, $Start_Date, $Return_by

UPDATE RESOURCES

SET Status_s='IN REPAIR', Start_date=$Rep_start, Return_by=$Return_by

WHERE Res_name=$Res_name AND Res_ID=$Res_ID

end for
```

    - Else if ***Request*** button is clicked: Update resource and request; Display **Request** form; Input request *start date* ($Start_date) and *return date* ($Return_by);

```
for each $Start_date, $Return_by
UPDATE REQUEST
SET Start_date=$Start_date, Return_by=$Return_by
WHERE Res_ID=$Res_ID
end for
```

    - Else if ***Close*** button is clicked:
        - Break While.
- Go back to **Main Menu** form.

Search Results (Deploy/Repair/Request)

Abstract Code

- Given that *username* ($Username) is provided.
- Display Resources in Use

```
SELECT R.Res_ID, R.Res_name, I.Description, R.Username, RE.Start_date, RE.Return_by
FROM RESOURCES AS R INNER JOIN REQUEST AS RE ON R.Res_ID=RE.Res_ID
INNER JOIN INCIDENT AS I ON I.Inc_ID=RE.Inc_ID
Where I.Username=$Username AND R.Status_s =“in use”
SORT BY R.Res_ID
```

- Display Resources Requested by me

```
SELECT R.Res_ID, R.Res_name, I.Description, R.Username, RE.Return_by
FROM RESOURCES AS R INNER JOIN REQUEST AS RE ON R.Res_ID=RE.Res_ID
INNER JOIN INCIDENT AS I ON I.Inc_ID=RE.Inc_ID
WHERE I.Username=$Username AND R.Status_s!=“in use”
SORT BY R.Res_ID
```

- Display Resources Requests received by me

```
SELECT R.Res_ID, R.Res_name, I.Description, R.Username, RE.Return_by
FROM RESOURCES AS R INNER JOIN REQUEST AS RE ON R.Res_ID=RE.Res_ID
INNER JOIN INCIDENT AS I ON I.Inc_ID=RE.Inc_ID
WHERE R.Username=$Username
SORT BY RE.Return_by
```

- Display Repairs Schedule/In-Progress

```
SELECT R.Res_ID, R.Res_name, R.Rep_start, R.Rep_ready
FROM RESOURCES AS R
WHERE R.Username=$Username AND R.Status_s=“in repair”
```

- While TRUE:
    - If ***Return*** button is clicked

```
for each $Status_s
UPDATE RESOURCES AS R
SET R.Status_s="available"
WHERE R.Res_ID=$Res_ID
```

- Else if ***Cancel*** button is clicked

```
for each $Status_s
DELETE FROM REQUEST
WHERE REQUEST.Res_ID=$Res_ID
```

- Else if ***Deploy*** button is clicked

```
for each $Status_s
UPDATE RESOURCES AS R
SET R.Status_s="in use"
WHERE R.Res_ID=$Res_ID
```

- Else if ***Reject*** button is clicked

```
for each $Res_ID
DELETE FROM REQUEST
WHERE REQUEST.Res_ID=$Res_ID
```

- Else if Cancel button is clicked

```
for each $Status_s
UPDATE RESOURCES AS R
SET R.Status_s="available"
WHERE R.Res_ID=$Res_ID
```

- Else if Shut-down button is clicked:
  - Break While.
- Else:
  - Do nothing.

- Go back to **Main Menu** form.

Resource Report

Abstract Code

- Given that the *username* ($Username) is provided
- Summary the resource information

```
SELECT Prim_ESF, count * as Num_tot
FROM RESOURCES
WHERE username=$username
GROUP BY Prim_ESF
```

```
ORDER BY Prim_ESF;


SELECT Prim_ESF, count * Num_use
FROM RESOURCES
WHERE Status_s='in use' AND Username=$Username
GROUP BY Prim_ESF
ORDER BY Prim_ESF;
```

- Display *ESF* ($ESF_ID_Desc), *total number of resources* ($Num_tot), *and number of resources in use* ($Num_use), sorted by ESF.