# **Abstract Code - Search Resource**

While no buttons are pushed, do nothing;

//populate dropdowns

SELECT ID\_description FROM ESF;

SELECT Inc\_ID, Description FROM INCIDENT;

If data validation is successful for all Keyword($Keyword), ESF($ID\_description), Location\_distance($Distance) and Incident($Inc\_ID, $Description):

If ***Search*** button is pushed, then do the following:

Display Search Result table according to search conditions;

SELECT R.Res\_ID, R.Res\_name, R.Username, R.Price, R.Costunit, R.Status, R.Available\_date, Distance

FROM RESOURCES AS R INNER JOIN RES\_CAP ON RESOURCES. Res\_ID= RES\_CAP. Res\_ID

WHERE (Res\_name LIKE $Keyword OR Model LIKE $Keyword OR Res\_capability LIKE $Keyword) AND (ID\_desc LIKE $ID\_description) AND (Distance<=$Distance)

ORDER BY Distance ASC;

/\* Distance is calculated by application through *haversine* formula \*/

else if ***Cancel*** button is pushed: go to **ERMS main menu** of user;

else if data input fields are invalid, go to **Search Resource** form, with error message;

**Search Result**

Find the Incident according to search conditions;

Display the name($Description) and ID($Inc\_ID) of incident;

Find the resources according to search conditions;

Display the resources($Res\_ID) and their attributes, sorted by distance, then availability($Status) and then name($Res\_name);

While no buttons are pushed, do nothing;

While a button is pushed, then do the following:

If ***Deploy*** button is pushed, then do the following: Update resource; Input expected return time;

//update status

for each $Status

UPDATE RESOURCES

SET Status=’NOT AVAILABLE’

WHERE Res\_name=$Res\_name AND Res\_ID=$ Res\_ID

end for

else if ***Repair*** button is pushed: Update resource; Display **Repair** form;

input repair start date($Rep\_start) and available date ($Available\_date)

for each $Status, $Rep\_start, $Available\_date

UPDATE RESOURCES

SET Status=’IN REPAIR’, Rep\_start=$Rep\_start, Available\_date=$ Available\_date

WHERE Res\_name=$Res\_name AND Res\_ID=$Res\_ID

end for

else if ***Request*** button is pushed: Update resource; update request; Display **Rquest** form;

input request start date($Start\_date) and return date ($Return\_by)

for each $Start\_date, $Return\_by

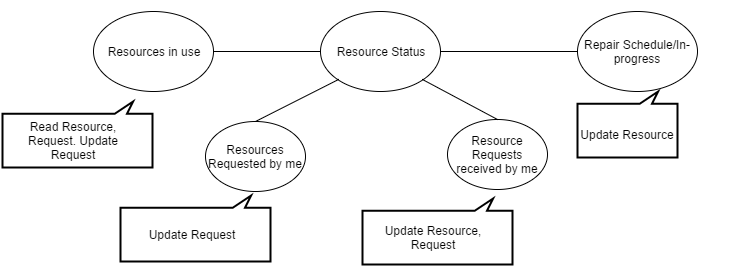
UPDATE REQUEST

SET Start\_date=$Start\_date, Return\_by=$Return\_by

WHERE Res\_ID=$Res\_ID

end for

else if ***Close*** button is pushed: Go to **ERMS main menu** of user;



# **Abstract Code w/ SQL- Resource Status**

# **Resources in use**

//application provides $Username of current user

//showing 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 = “in use”

SORT BY R.Res\_ID

If ***Return*** button is pressed, then do the following: Update resource;

for each $Status

UPDATE RESOURCES AS R

SET R.Status = “available”

WHERE R.Res\_ID = $Res\_ID

# **Resources requested by me**

//application provides $Username of current user

//showing 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 != “in use”

SORT BY R.Res\_ID

If ***Cancel*** button is pressed, then do the following: cancel request;

for each $Status

DELETE FROM REQUEST

WHERE REQUEST.Res\_ID = $Res\_ID

# **Resources Requests received by me**

//application provides $Username of current user

//showing resources 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

If ***Deploy*** button is pressed, then do the following: Update resource;

for each $Status

UPDATE RESOURCES AS R

SET R.Status = “in use”

WHERE R.Res\_ID = $Res\_ID

If ***Reject*** button is pressed, then do the following:Cancel request;

for each $Res\_ID

DELETE FROM REQUEST

WHERE REQUEST.Res\_ID = $Res\_ID

# **Repairs Schedule/ In-Progress**

//application provides $Username of current user

//showing repairs schedule/In-process

SELECT R.Res\_ID, R.Res\_name, R.Rep\_start, R.Rep\_ready

FROM RESOURCES AS R

WHERE R.Username = $Username AND R.Status = “in repair”

If ***Cancel*** button is pressed, then do the following: Cancel request;

for each $Status

UPDATE RESOURCES AS R

SET R.Status = “available”

WHERE R.Res\_ID = $Res\_ID