**Table of Contents:**

Login

Main Menu

My Items

Searching for Items

View Item

Propose Swap

Accept or Reject Swaps

Rate Swaps

Swap History

Update My Information

# Login

Abstract Code

Graphical user interface, text

Description automatically generated

* User Enters Email , Password
* If the data validation is successful, then click the Login button

|  |
| --- |
| Select password from user where Email = '$Email '; |

* If user record not found then show message “User not found” go back to **Login** form
* If user record found but User.password!= '$Password' then show message “Incorrect password” go back to **Login** form
* Else Go to **Main Menu** Page
* Email will be stored in '$Email' variable
* Show Register button
* On click will go to **Register User** module

# 

# User Registration

Abstract Code

Graphical user interface

Description automatically generated

* Email Textbox: User will be allowed to enter email address.All the necessary Data Validation will be in place
* Nickname Textbox: User will be allowed to enter Nickname.All the necessary Data Validation will be in place
* Password Textbox: User will be allowed to enter Password.All the necessary Data Validation will be in place
* First Name Textbox: User will be allowed to enter First Name.All the necessary Data Validation will be in place
* Last Name Textbox: User will be allowed to enter Last Name.All the necessary Data Validation will be in place
* Postal Code Textbox: User will be allowed to enter Postal code.All the necessary Data Validation will be in place
* City and State will be populated based on the verified postal code

|  |
| --- |
| Select City,State from address where postalcode='$Postalcode'; |

* Phone Number Textbox: User will be allowed to enter Phone Number.All the necessary Data Validation will be in place
* Show phone number in swaps checkBox: User will allowed to check the checkbox.All the necessary Data Validation will be in place
* Type drop down box: User will be allowed to select from the set of values for the phone type.This dropbox is enabled if there is a legit phone number filled in the phone number text box. Below query is executed for fetching the values.
* Postal data validation for each field the values will be stored in variables.
* The register button is enabled only if all the above fields are validated. On click below insert query will be executed

|  |
| --- |
| Insert into user (Email,Password,FirstName,LastName,Nickname,Phonenumber,Postalcode) values ('$Email', '$Password', '$FirstName', '$LastName', '$NickName', '$Phonenumber', '$Postalcode'); |

|  |
| --- |
| Insert into phone(Number,Type,ShareFlag) values ('$Phonenumber, '$Type', '$ShareFlg'); |

* If Insert Statement fails then show message “Registration Failed”
* Else Go to “Login Page”

# Main Menu

Abstract Code

Graphical user interface, diagram

Description automatically generated

* Show ListItem,My Items,Search Items,Swap History,Update my info,My Rating,Unaccepted Swaps,Unrated Swaps,Logout
* Upon

1. Click ListItem to go ListItem module
2. Click MyItems to go MyItems module
3. Click Search Items to go Search Items module
4. Click Swap History to go Swap History module
5. Click Update my Info to go Edit/View Profile
6. My rating will display your aggregated rating. Below query is executed

|  |
| --- |
| Select MyRating from user where Email='$Email'; |

1. Unaccepted Swaps will show count of unaccepted swaps as a hyperlink. On click will go Swap History module

* For getting the Unaccepted swaps below select query is executed

|  |
| --- |
| Select UnacceptedSwaps from user where Email='$Email'; |

1. Unrated Swaps will show count of unrated swaps as a hyperlink. On click will go Swap History module.

* For getting the Unaccepted swaps below select query is executed

|  |
| --- |
| Select UnratedSwaps from user where Email='$Email'; |

1. Click Logout to go the Login module

List Item

Graphical user interface, text, application, chat or text message

Description automatically generated

Abstract Code

* Show Game Type Drop box. The values will be prepopulated by the code
* Based on the selection of the GameType further fields will be shown
* **Jigsaw Puzzle**

1.Show the Title Text Box

2.Show Condition Drop Box. The values will be prepopulated by the code

3.Show the Piece Count

4.Show the Description Text Box ( With fixed number of characters)

5.Enable the List Item Button and Once clicked below Insert query will be executed

|  |
| --- |
| Insert into item (Email,Title,Condition,Description,Itemtype) values ('$Email', '$Title', '$Condition', '$Description', '$ItemType'); |

The Itemnumber is incremental number which will be populated for each insert into Item table.

|  |
| --- |
| Select max(ItemNumber) ItemNumber from item where Title='$Title' and ItemType='$ItemType'and Email='$Email'; |

Above select query output is used to get the Itemnumber for the current transaction and stored into '$ItemNumber' .

|  |
| --- |
| Insert into itemtype (ItemNumber,Name,PieceCount) values ('$ItemNumber' , '$ItemType', '$PieceCount') ; |

* **BoardGame**

1.Show the Title Text Box

2.Show Condition Drop Box. The values will be prepopulated by the code

3.Show the Description Text Box ( With fixed number of characters)

4.Enable the List Item Button and Once clicked below Insert query will be executed

|  |
| --- |
| Insert into item (Email,Title,Condition,Description,Itemtype) values ('$Email', '$Title', '$Condition', '$Description', '$ItemType'); |

The Itemnumber is incremental number which will be populated for each insert into Item table.

|  |
| --- |
| Select max(ItemNumber) ItemNumber from item where Title='$Title' and ItemType='$ItemType'and Email='$Email'; |

Above select query output is used to get the Itemnumber for the current transaction and stored into '$ItemNumber' .

|  |
| --- |
| Insert into itemtype (ItemNumber,Name) values ('$ItemNumber' , '$ItemType') ; |

* **CardGame**

1.Show the Title Text Box

2.Show Condition Drop Box. The values will be prepopulated by the code

3.Show the Description Text Box ( With fixed number of characters)

4.Enable the List Item Button and Once clicked below Insert query will be executed

|  |
| --- |
| Insert into item (Email,Title,Condition,Description,Itemtype) values ('$Email', '$Title', '$Condition', '$Description', '$ItemType'); |

The Itemnumber is incremental number which will be populated for each insert into Item table.

|  |
| --- |
| Select max(ItemNumber) ItemNumber from item where Title='$Title' and ItemType='$ItemType'and Email='$Email'; |

Above select query output is used to get the Itemnumber for the current transaction and stored into '$ItemNumber' .

|  |
| --- |
| Insert into itemtype (ItemNumber,Name) values ('$ItemNumber' , '$ItemType') ; |

* **VideoGame**

1.Show the Title Text Box

2.Show Condition Drop Box. The values will be prepopulated by the code

3.Show the Description Text Box ( With fixed number of characters)

4. Show Platform DropBox. The values will be prepopulated by the code

5. Show Media DropBox. The values will be prepopulated by the code

6.Enable the List Item Button and Once clicked below Insert query will be executed

|  |
| --- |
| Insert into item (Email,Title,Condition,Description,Itemtype) values ('$Email', '$Title', '$Condition', '$Description', '$ItemType'); |

The Itemnumber is incremental number which will be populated for each insert into Item table.

|  |
| --- |
| Select max(ItemNumber) ItemNumber from item where Title='$Title' and ItemType='$ItemType'and Email='$Email'; |

Above select query output is used to get the Itemnumber for the current transaction and stored into '$ItemNumber' .

|  |
| --- |
| Insert into itemtype (ItemNumber,Name,Platform,Media) values ('$ItemNumber' , '$ItemType', '$Platform', '$Media') ; |

* **ComputerGame**

1.Show the Title Text Box

2.Show Condition Drop Box. The values will be prepopulated by the code

3.Show the Description Text Box ( With fixed number of characters)

4. Show Platform DropBox. The values will be prepopulated by the code

5.Enable the List Item Button and Once clicked below Insert query will be executed

|  |
| --- |
| Insert into item (Email,Title,Condition,Description,Itemtype) values ('$Email', '$Title', '$Condition', '$Description', '$ItemType'); |

The Itemnumber is incremental number which will be populated for each insert into Item table.

|  |
| --- |
| Select max(ItemNumber) ItemNumber from item where Title='$Title' and ItemType='$ItemType'and Email='$Email'; |

Above select query output is used to get the Itemnumber for the current transaction and stored into '$ItemNumber' .

|  |
| --- |
| Insert into itemtype (ItemNumber,Name,Platform) values ('$ItemNumber' , '$ItemType', '$Platform') ; |

On Completion of Transaction below message might be displayed

* On Success – Below message box will be shown with Item Number. For fetching the item number below query is executed. The user will be redirected to the Main Menu.

|  |
| --- |
| Select max(ItemNumber) ItemNumber from item where Title='$Title' and ItemType='$ItemType'and Email='$Email'; |

Graphical user interface, application

Description automatically generated

* On Failure – Message box will be shown “Item Listing Failed”

# My Items

**Table

Description automatically generated**

Abstract Code

* Show Item Counts for the User.Below select query is executed for it

|  |
| --- |
| Select sum(BoardGames) BoardGames, sum (CardsGames) CardsGames,sum(ComputerGames) ComputerGames,sum(JigSawPuzzles) JigSawPuzzles,sum(VideoGames)+sum(BoardGames) +sum (CardsGames)+sum(ComputerGames)+sum(JigSawPuzzles) +sum(VideoGames) Total from  (Select case when ItemType=’Board Game’ then 1 else 0 end BoardGames,  case when ItemType=’Card Game’ then 1 else 0 end CardsGames,  case when ItemType=’Computer Game’ then 1 else 0 end ComputerGames ,  case when ItemType=’JigSaw Puzzle’ then 1 else 0 end JigSawPuzzles,  case when ItemType=’Video Game ’ then 1 else 0 end VideoGames  from item I join itemtype ty on I.ItemType=ty.Itemtype and I.ItemNumber=ty. ItemNumber where I.Email='$Email') ; |

* Show My Item Details by executing below select query

**Table

Description automatically generated**

|  |
| --- |
| Select I.ItemNUmber, I.Itemtype GameType, I.Title, I.Condition, I.Description,ty.Plaform,ty.Media,ty.PieceCount,ty.Description  from item I join itemtype ty on I.ItemType=ty.Itemtype and I.ItemNumber=ty. ItemNumber where Email ='$Email'; |

* **The Details button for every Item will consume the data returned from the below query**

|  |
| --- |
| Select I.ItemNUmber, I.Itemtype GameType, I.Title, I.Condition, I.Description,ty.Platform,ty.Media,ty.PieceCount,ty.Description  from item I join itemtype ty on I.ItemType=ty.Itemtype and I.ItemNumber=ty. ItemNumber where Email ='$Email' and I.ItemNumber=$ItemNumber; |

* **The Description restricting to 100 chars will be done in the code**

# Searching for Items

Graphical user interface, text, application

Description automatically generated

Abstract Code

* **By Keyword**:

1. Show the “By keyword” radiobutton
2. Once the radiobutton is selected then enable the textbox
3. On click of the “Search” button, below query will be executed

|  |
| --- |
| Select U.Email, I.ItemNUmber, I.Itemtype GameType, I.Title, I.Condition, I.Description,ty.Platform,ty.Media,ty.PieceCount,ty.Description,addr.PostalCode,addr.longitude,addr.latitude  from user U join item I on U.Email=I.Email join itemtype ty on I.ItemType=ty.Itemtype and I.ItemNumber=ty. ItemNumber;  join address addr on I.PostalCode= U.PostalCode  where I.Email=$Email where U.Email <>'$Email' and (lower(Title) like ‘%lower($Keyword)%’ or lower(Description) like ‘%lower($Keyword)%’) ; |

**Below query will provide the logged user address details**

|  |
| --- |
| Select U.User, addr.PostalCode,addr.longitude,addr.latitude  from user U join address addr on I.PostalCode= U.PostalCode  where I.Email='$Email' where U.Email ='$Email'; |

**/\* Loggined User details\*/**

**Using the longitude and latitude of the logged user and the searched item Owner distance between is calculated**

1. On execution of the query
2. For return of records below screen will be displayed

Table

Description automatically generated

1. For no rows returned Message will be shown “No results found”

* **In My postal code**:

1. Show the “In My postal code” radiobutton
2. On click of the “Search” button, below query will be executed

|  |
| --- |
| Select U.Email, I.ItemNUmber, I.Itemtype GameType, I.Title, I.Condition, I.Description,ty.Platform,ty.Media,ty.PieceCount,ty.Description,addr.PostalCode,addr.longitude,addr.latitude  from user U join item I on U.Email=I.Email join itemType ty on I.ItemType=ty.Itemtype and I.ItemNumber=ty. ItemNumber  join address addr on I.PostalCode= U.PostalCode  where I.Email='$Email' where U.Email <>'$Email' and U.postalCode='$PostalCode' /\*Logged in User Postal Code\*/ ; |

Below query will provide the logged user address details

|  |
| --- |
| Select U.User, addr.PostalCode,addr.longitude,addr.latitude  from user U join address addr on I.PostalCode= U.PostalCode  where I.Email='$Email' where U.Email ='$Email'; |

**/\* Logged User details\*/**

**Using the longitude and latitude of the logged user and the searched item Owner distance between is calculated**

1. On execution of the query

a.For return of records below screen will be displayed

Table

Description automatically generated

b.For no rows returned Message will be shown “No results found”

* **Within X miles of me**:

1. Show the “Within the X miles” radiobutton
2. Once the radiobutton is selected then enable the mile selector
3. On click of the “Search” button, below query will be executed

|  |
| --- |
| Select U.Email, I.ItemNUmber, I.Itemtype GameType, I.Title, I.Condition,  I.Description,ty.Platform,ty.Media,ty.PieceCount,ty.Description,addr.PostalCode,addr.longitude,addr.latitude from user U join item I on U.Email=I.Email join itemtype ty on I.ItemType=ty.Itemtype and I.ItemNumber=ty. ItemNumber  join address addr on I.PostalCode= U.PostalCode;  where I.Email='$Email' where U.Email <>'$Email'; |

**Below query will provide the loginned user address details**

|  |
| --- |
| Select U.User, addr.PostalCode,addr.longitude,addr.latitude  from user U join address addr on I.PostalCode= U.PostalCode  where I.Email='$Email' where U.Email ='$Email' ; |

**/\* Logged In User details\*/**

**Using the longitude and latitude of the logged in user and the searched item Owner distance between is calculated**

1. The distance of each user with the logged user is calculated and then compared with the selected miles and only items whose distance with the logged in user is within the selected miles
2. On execution of the query
3. For return of records below screen will be displayed

Table

Description automatically generated

1. For no rows returned Message will be shown “No results found”

* In postal code:

1. Show the “In postal code” radiobutton
2. Once the radiobutton is selected then enable the textbox. Do validate the postal code value.( It should match with the postal code reference table)

Below query for getting the postal code from Address table

**Select distinct postal code from Address**

1. On click of the “Search” button, below query will be executed

|  |
| --- |
| Select U.Email, I.ItemNUmber, I.Itemtype GameType, I.Title, I.Condition, I.Description,ty.Platform,ty.Media,ty.PieceCount,ty.Description,addr.PostalCode,addr.longitude,addr.latitude  from user U join item I on U.Email=I.Email join itemtype ty on I.ItemType=ty.Itemtype and I.ItemNumber=ty. ItemNumber  join address addr on I.PostalCode= U.PostalCode  where I.Email='$Email' where U.Email <>'$Email' and U.postalCode='$Postalcode' /\*Input from textbox\*/; |

**Below query will provide the loginned user address details**

|  |
| --- |
| Select U.User, addr.PostalCode,addr.longitude,addr.latitude  from user U join address addr on addr.PostalCode= U.PostalCode  where I.Email='$Email'; |

**/\* Logged In User details\*/**

**Using the longitude and latitude of the logged in user and the searched item Owner distance between is calculated**

1. The distance of each user with the logged in user is calculated and then compared with the selected miles and only items whose distance with the logged in user is within the selected miles
2. On execution of the query
3. For return of records below screen will be displayed

Table

Description automatically generated

1. For no rows returned Message will be shown “No results found”

# View Item

Abstract Code

* Once the Details button is clicked from the above list then below screen is shown

Graphical user interface, text, application

Description automatically generated

* Below select query is executed for getting the above details

|  |
| --- |
| Select I.ItemNUmber,U.NickName, I.Itemtype GameType, I.Title, I.Condition, I.Description,ty.Platform,ty.Media,ty.PieceCount,ty.Description,addr.longtitude,addr.latitude,u.myrating,u.Email/\*not to be shown on screen\*/ from user U join item I on U.Email=I.Email join itemtype ty on I.ItemType=ty.Itemtype and I.ItemNumber=ty. ItemNumber join address addr on addr.PostalCode= U.PostalCode where ItemNumber='$iTemNumber' /\*itemselected\*/; |

**Below query will provide the logged user address details**

**/\* Logged In User details\*/**

|  |
| --- |
| Select U.User, addr.PostalCode,addr.longitude,addr.latitude  from user U join address addr on I.PostalCode= U.PostalCode  where I.Email=$Email where U.Email ='$Email'; |

* Using the longitude and latitude of the logged in user and the searched item Owner distance between is calculated
* Show “Propose Swap” button and once clicked show the “Propose Swap Page”

# Propose Swap

Abstract Code

* Once the “Propose Swap” button is clicked then below screen is shown

Table

Description automatically generated

* Below select query is executed for getting the above details

|  |
| --- |
| Select I.ItemNUmber,U.NickName, I.Itemtype GameType, I.Title, I.Condition, I.Description,ty.Platform,ty.Media,ty.PieceCount,ty.Description  from user U join item I on U.Email=I.Email join itemtype ty on I.ItemType=ty.Itemtype and I.ItemNumber=ty. ItemNumber  where U.Email ='$Email'/\* logged in \*/ and I.Item not in (select distinct Item from Swap) ; |

* The Selected Item Details from the previous screen are persisted for this page
* All the Items for the logged in user will displayed with select radiobutton for every Item
* Once logged in User selects one item then “Confirm” Button will be enabled

|  |
| --- |
| Insert into swap  (ProposedEmail,DateProposed,Status,Other User,CounterPartyEmail,ProposedItem,DesiredItem)  Values ('$Email'/\*logged in\*/, systemdate,”Proposed”,'$Nickname'/\*Select Previous screen\*/, '$CounterPartyEmail'/\*Select Previous screen\*/, '$Item','$DesiredItem'/\*Select Previous screen\*/); |

* Once the “Confirm” Button is clicked then below insert statement will be executed

**SwapId will be an incremental Number which will be populated for every insert**

# Accept or Reject Swaps

Abstract Code

* You can come on this screen by clicking “Unaccepted Swaps” link then below screen is shown

Graphical user interface, application, table

Description automatically generated

* Below select query is executed for getting the above details

|  |
| --- |
| Select ProposedDate,Desired ,Swapid,Item,Proposer,ProposedItem  ,addr.longtitude,addr.latitude from user U join swap S on U.Email=S.CounterPartyEmail join address addr on u.postalcode=addr.postalcode where S.Status='Proposed'; |

**/\* Getting the Proposer rating,nickname, and longtitude and latitiude,phonenumber,email and first name\*/**

|  |
| --- |
| Select ed U.myrating,u.nickname,addr.longtitude,addr.latitude  from user U join swap S on U.Email=S.ProposedEmail  join address addr on u.postalcode=addr.postalcode  where S.Status='Proposed'; |

* If Accept button is clicked then below Update Statement is executed

|  |
| --- |
| Update swap set status='Accepted' ,DateResponded=system date where CounterpartyEmail='$CounterpartyEmail' and Swapid='$Swapid'; |

**And below message will be displayed**

**Graphical user interface, text, application

Description automatically generated**

* If Reject button is clicked then below Update Statement is executed

|  |
| --- |
| Update swap set status='Rejected' ,DateResponded=system date where CounterpartyEmail='$CounterpartyEmail' and Swapid='$Swapid'; |

# Rate Swaps

Abstract Code

* Below table will be shown. Select query will be executed for this table

Table

Description automatically generated

|  |
| --- |
| Select dateresponsed as acceptancedatee, 'Proposer',ProposedItem, DesiredItem from swap where ProposedEmail='$Email' and ProposerRating is null  Union  Select dateresponsed as acceptancedatee, 'CounterParty',ProposedItem, DesiredItem from swap where CounterPartyEmail='$Email' and CounterRating is null; |

* **Every row in the table will have a Rating dropdown with prepopulated values**
* **Once the rating is select for that swap then below update Stmt is executed**

|  |
| --- |
| Update swap set counterpartyrating='$Rating' where CounterpartyEmail='$Email' and Swapid='$SwapId'; |

|  |
| --- |
| Update swap set proposerrating ='$Rating' where ProposedEmail='$Email' and Swapid='$SwapId'; |

# Swap History

Abstract Code

* Below table will be shown. Select query will be executed for this table

Table

Description automatically generated

|  |
| --- |
| Select 1 Total, Case when Status='Accepted' then 1 else 0 end Accepted, Case when Status='Rejected' then 1 else 0 end Rejected, 'Proposer' from swap where ProposedEmail='$Email' Union  Select 1 Total, Case when Status=’Accepted’ then 1 else 0 end Accepted  , Case when Status='Rejected' then 1 else 0 end Rejected, 'CounterParty' from swap where CounterpartyEmail='$Email' ; |

* Below table will be shown. Select query will be executed for this table

Table

Description automatically generated

|  |
| --- |
| Select ProposedDate,DateResponded Accepted\_Rejected\_Date,Status, 'Proposer' MyRole, ProposedItem,DesireedItem,OtherUSer,ProsperRating  from swap where ProposedEmail ='$Email'  Union  Select ProposedDate,DateResponded Accepted\_Rejected\_Date,Status, 'CounterParty' MyRole, ProposedItem,DesireedItem,OtherUSer,CounterpartyRating  from swap where CounterpartyEmail ='$Email'; |

* **If the any Rating is null then Drop down rating will provided to select the rating**

**Table

Description automatically generated**

**Once rating is selected then below Update stmt will be executed**

|  |
| --- |
| Update swap set counterpartyrating=$rating where CounterpartyEmail='$Email'and Swapid='$SwapId'; |

|  |
| --- |
| Update swap set Proposerrating =$rating where ProposedEmail='$Email' and Swapid='$SwapId'; |

* **Once the Detail link/Button is clicked below screen is shown**

**Graphical user interface, text, application, email

Description automatically generated**

**Below select queries need to be executed**

|  |
| --- |
| Select ProposedDate,DateResponded Accepted\_Rejected\_Date,Status, ‘Proposer’ MyRole, ProposedItem,DesireedItem,OtherUSer,CounterpartyRating from swap where CounterpartyEmail='$Email'; |

|  |
| --- |
| Select I.ItemNUmber, I.Itemtype GameType, I.Title, I.Condition, I.Description,ty.Platform,ty.Media,ty.PieceCount,ty.Description  from item I join itemtype ty on I.ItemType=ty.Itemtype where ItemNumber='$ProposedItem'; |

|  |
| --- |
| Select I.ItemNUmber, I.Itemtype GameType, I.Title, I.Condition, I.Description,ty.Platform,ty.Media,ty.PieceCount,ty.Description  from item I join itemtype ty on I.ItemType=ty.Itemtype where ItemNumber='$DesiredItem'; |

|  |
| --- |
| Select U.User, addr.PostalCode,addr.longitude,addr.latitude from user U join address addr on I.PostalCode= U.PostalCode where I.Email='$Email' where U.Email='$ProposedEmail';  Select U.User, addr.PostalCode,addr.longitude,addr.latitude from user U join address addr on I.PostalCode= U.PostalCode where I.Email=$Email where U.Email ='$CounterpartyEmail'; |

# 

# Update My Information

Abstract Code

Below table will be shown. Select query will be executed

Graphical user interface, application

Description automatically generated

|  |
| --- |
| Select Nickname,password,FirstName,LastName,City,State,Postalcodr,PhoneNumber,PhoneType from user U join address addr on U.postalCode=addr. postalCode  where Email='$Email'; |

* Nickname Textbox: User will be allowed to change Nickname.All the necessary Data Validation will be in place
* Password Textbox: User will be allowed to change Password.All the necessary Data Validation will be in place
* First Name Textbox: User will be allowed to enter First Name.All the necessary Data Validation will be in place
* Last Name Textbox: User will be allowed to enter Last Name.All the necessary Data Validation will be in place
* Postal Code Textbox: User will be allowed to change Postal code.All the necessary Data Validation will be in place
* City and State will be populated based on the verified postal code

|  |
| --- |
| Select City,State from address where postalcode=$postalcode; |

* Phone Number Textbox: User will be allowed to change Phone Number.All the necessary Data Validation will be in place
* Show phone number in swaps checkBox: User will allowed to uncheck the checkbox.All the necessary Data Validation will be in place
* Type drop down box: User will be allowed to change from the set of values for the phone type.This dropbox is enabled if there is a legit phone number filled in the phone number text box.
* The Update button is enabled only if all the above fields are validated. On click below update query will be executed

|  |
| --- |
| Update user set password='$Password',FirstName='$FirstName',LastName='$LastName',Nickname'$NickName',PostalCode='$PostalCode' where Email='$Email';  Update phone set Type='$Type',ShareFlag='$ShareFlg' where Number ='$PhoneNumber'; |