Table of Contents:

Customers

Customer Main Menu

Abstract Code

- Show "***View Profile***", "***Check Tool Availability***", "***Make Reservation***", "***Purchase Tool***", "***Exit/Logout***" options/tabs.
- Upon:
    - Click ***View Profile*** button - jump to **View Profile** task.
    - Click ***Check Tool Availability*** button - jump to **Check Tool Availability** task.
    - Click ***Make Reservation*** button - jump to **Reservation form**.
    - Click ***Purchase Tool*** button - jump to **Purchase Tool form**.
    - Click ***Exit/Logout*** button - Invalidate login session and go back to the **Login form**.

Login

Abstract Code

- User need to select one role from Customer or Clerk and the default one is Customer
- When User enters *username(*$username*)* and *password* ($password)into input fields, entered data is validated
- **If** data validation is successful for both *username* and *password* input field, then:
    - When ***Sign in*** button is clicked:
        - **If** Customer is selected:

```
SELECT Password FROM Customer  WHERE username = '$username';
```

            - **If** Customer username and password record are not found:
                - ➢ Go to **Customer Registration** form.
            - **Else if** Customer username record is found but Customer.Password != '$Password'::
                - ➢ Go back to **Login** form, with error message "Password is incorrect for this user".
            - **Else**:
                - ➢ Go to **Customer Main Menu** page.
        - **If** Clerk is selected:

```
SELECT Password FROM Clerk WHERE username = '$username';
```

- **If** Clerk record is not found
  - ➢ Go back to **Login form**, with error message "There is no clerk with this username"
- **Else if** Clerk.Password != '$Password':
  - ➢ Go back to **Login form**, with error message "Password is incorrect for this user"
- **Else**:
  - ➢ Go to **Clerk Main Menu** page.

- Else *username* and *password* input fields are invalid, ***Sign in*** button is disabled and display **Login form**, with error message

Registration

Abstract Code

- Display enter field: *$Username*, *$First Name*, *$Middle Name, $Last Name, $Email Address, $PassWord, $Re-type Password, $Street Address, $Zip Code, $Name on Credit card, $Credit Number, $CVC, $validate data*, user select one *$State* from all states; use select one *$Expiration Month* from 12 months; user select one $*Expiration Year*.

  $assume application requires user to enter all field above and requires user to choose one kind of phone as primary phone and user must enter phone number for that kind of phone. Application can obtain the *$area_code, $phone_number, $extension*, $phone_type for each kind of phone user entered, and $*primary* to record which one is primary phone

- **If** data validation is successful for all input field and all business logic is satisfied, then:
  - When ***Register*** button is clicked:

  For each phone user entered: {

```
INSERT INTO Phone(area_code, phone_number, extension, phone_type)
VALUES ('$area_code', '$phone_number', '$extension', '$phone_type')
```

  If err, reload the page

If ($phone_type = $ primary): {
$assume application can use $primary_phone_id to record surrogate key of PhoneID of newly inserted phone number
}

$assume application can obtain the surrogate key $PhoneID after each new phone is inserted into Phone table, and insert it into a list $list_of_PhoneID

}

```
INSERT INTO Customer(username, email, password, first_name, middle_name,
last_name, card_number, name_on_card, month, year, cvc_3_digit Number, zip_code,
zip_extension, state, city, street, PhoneID) VALUES ('$username', '$email', '$password',
'$first_name', '$middle_name', '$LastName', '$card_number', '$name_on_card',
'$Month', '$Year', '$cvc_3_digit_number', '$zip_code', '$zip_extension', '$state', '$city',
'$street', '$primary_phone_id');
```

If err, reload the page

$assume application can obtain the $CustomerID after a new customer is inserted into Customer table, plus

For each $PhoneID in $list_of_PhoneID: {

```
INSERT INTO Secondary(CustomerID, PhoneID) VALUES ('$CustomerID', '$PhoneID')
```

If err, reload the page
}

- **If** there is error: go back to **Reservation form,** with error message.

View Profile

Abstract Code

- Customer clicked on ***View Profile*** button from **Customer Main Menu**:
- Run the **View Profile** task: query for information about the Customer and their profile, assume $CustomerID of current customer is managed by application

  - **View Customer** to Find the current Customer using the Customer.username, and display:
    - Customer E-mail
    - Full Name (first name + middle name + last name)
    - Home Phone (if not null)

- Work Phone (if not null)
- Cell Phone (if not null)
- Address

```
SELECT email, phone_type, CONCAT (area_code, phone_number, '-', extension) AS phone, CONCAT (first_name, ' ', middle_name, ' ' , last_name) AS full_name, CONCAT (street, ' ', city, ' ',  state, ' ', zip_code, ' ' , zip_extension) AS address  FROM Customer NATURAL JOIN Secondary  NATURAL JOIN Phone WHERE CustomerID = '$CustomerID';
```

- **View Reservation** Find the following items and display them from most recent to oldest, and for each Reservation display:
    - Reservation id
    - Start Date
    - End Date
    - Number of Days
    - Total Deposit Price
    - Total Rental Price
    - pick-up clerk name
    - drop-off clerk name

```
SELECT reservationID, start_date, end_date, DATEDIFF(end_date, start_date) AS number_of_days, power_source, sub_option, sub_type, CONCAT(Clerk1.last_name, ' ', Clerk1.middle_name, ' ', Clerk1.last_name ) AS drop_clerk,  CONCAT(Clerk2.last_name, ' ', Clerk2.middle_name, ' ', Clerk2.last_name ) AS pick_clerk, SUM(original_price * 0.4) AS total_deposit,  SUM(DATEDIFF(end_date, start_date) * original_price * 0.15) AS total_rental FROM Reservation NATURAL JOIN AddRes NATURAL JOIN Tool LEFT OUTER JOIN Clerk AS Clerk1 ON Clerk1.ClerkID = Reservation.DropClerkID LEFT OUTER JOIN Clerk AS Clerk2 ON Clerk2.ClerkID = Reservation.PickClerkID WHERE Reservation.CustomerID = '$CustomerID' GROUP BY Reservation.reservationID ORDER BY start_date DESC;
```

Check Tool Availability

Abstract Code

When customer clicked "***Check Tool Availability***" button, a page is loaded,

Include **Basic Keyword Search Function**, wait user for entering *Start Date, End Date, Keyword, Type, Power Source, SubType*

While the ***Search*** button is not clicked, do nothing.
While the ***Search*** button is clicked, then: Search function return $type, $power_source, $sub_type, $keyword, and $start_date, $end_date from user entered

Assume $type, $power_source, $sub_type, $keyword will give us the name string of the selected item by radios or dropdown. In particular, if user select All type, we will get an empty string. $start_date, $end_date will have date format.

- View Tool
- Find tools matching the entered $type, $power_source, $sub_type, $keyword, $start_date, $end_date. Excluded those tools have been rented, in repair or for-sale. Display ToolID, Description, Rental Price and Deposit Price

```
SELECT ToolID, type, power_source, sub_option, sub_type, (0.15 *
original_price) AS rental_price FROM Tool  WHERE ToolID IN (
SELECT ToolID FROM Tool  WHERE type LIKE '%$type%' AND power_source=
'$power_source' AND sub_type = '$sub_type' AND sub_option LIKE
'%$Keyword%') AND ToolID IN (SELECT ToolID FROM AddRes NATURAL JOIN
Reservation WHERE Reservation.start_date > '$end_date' OR
Reservation.end_date < '$start_date') AND ToolID NOT IN ( SELECT ToolID
FROM ServiceOrder WHERE CURDATE() < service_end_date) AND ToolID NOT
IN (SELECT ToolID FROM ToolForSale);
```

- **If** the number of available tools is greater than 10:
  Display the error message: “Specify a more unique search”

- Assume each short description with a link containing the $ToolID, $power_source, $sub_option, when the short description of each available tool is clicked, call helper method **display tool detail**.

Make Reservation

Abstract Code

When customer clicked “***Make Reservation***” button, a page is loaded,

Include **Basic Keyword Search Function**, wait user for entering *Start Date, End Date, Keyword, Type, Power Source, SubType*

While the ***Search*** button is not clicked, do nothing.
While the ***Search*** button is clicked, then: Search function return $type, $power_source, $sub_type, $keyword, and $start_date, $end_date from user entered

- Call **Check Tool Availability** task to display each available tool's $Tool ID, $Description(a link containing the $ToolID, $power_source, $sub_option, call helper method **display tool detail** when called), $Rental price and $Deposit price
- When the ***Add*** checkbox of any available tool is clicked:
  Display that tool to the shopping cart, and hide that tool in the available tools for rent

- When the ***Remove*** checkbox of any tool in the shopping cart is clicked: show that tool in the available tools, and remove that tool in the shopping cart
- When the ***Calculate Total*** button is clicked,
  - **If** there is no tool in shopping cart
    display the **Make Reservation form** with error message "please add tools"
  - **Else if** the number of tool in shopping cart is greater than 10:
    display the **Make Reservation form** with error message "reduce the number of tool in the current Reservation to 10"
  - **Else**:

$Assume application keep $CustomerID and $start_date, $end_date from user input, and $Tool ID, $Description, $Rental price and $Deposit price of each tool in shopping cart and pass them to **Reservation Summary Form** and display all those information. By the way, $Number_of_Days_Rented, calculate $Total_Deposit_Price and $Total_Rental_Price are all calculated by application

When ***Submit*** button is clicked**:**

```
INSERT INTO Reservation(CustomerID, start_date, end_date) VALUES
('$CustomerID', '$start_date', '$end_date')
```

If err, reload the page

$Assume application can have a $list_of_ToolID to keep all the $ToolID in shopping cart, and get the surrogate key $reservationID of Reservation from above Query

For each $ToolID in $list_of_ToolID: {

```
INSERT INTO AddRes(ToolID, reservationID) VALUES ('$ToolID',
'$reservationID')
```

If err, reload the page

}

Display the **Reservation Confirmation Page** with all information in **Reservation Summary Form** plus $reservationID

When ***Reset*** button is clicked:
Remove the Reservation in the memory, Display **Make Reservation form**.

Purchase Tool

Abstract Code

Include **Basic Keyword Search Function**, wait user for entering *Keyword, Type, Power Source, SubType*

When the ***Search*** button is not clicked, do nothing.
When the ***Search*** button is clicked, then: Search function return $type, $power_source, $sub_type, $keyword

Assume $type, $power_source, $sub_type, $keyword will give us the name string of the selected item by radios or dropdown. In particular, if user select All type, we will get an empty string.

- **View tool** find sale- order matching the inputted $Keyword, tool type, power source, subtype and Display ToolID, Short Description(a link containing the $ToolID, $power_source, $sub_option, call helper method **display tool detail** when called), purchase price of each found tool.

```
SELECT SaleID, ToolID, power_source, sub_option, sub_type, (0.5 * original_price) AS purchase_price FROM ToolForSale NATURAL JOIN Tool WHERE sold_date IS NULL AND type LIKE '%$type%' AND power_source= '$power_source' AND sub_type = '$sub_type' AND sub_option LIKE '%$Keyword%';
```

When the ***Purchase Tool*** button of any available tool is clicked:
- add the tool to the shopping cart for purchasing, hide that tool in the available tools to purchase

When the ***Remove*** checkbox of any tool in the shopping cart for purchasing is clicked:
- remove this tool from the shopping cart for purchasing, add that tool back to the available list for purchasing

When the ***Calculate Total*** button is clicked,
- **If** there is no tool in the temporary list for Reservation:
  display the **Purchase Tool form** with error message “please purchase tools”
- **Else if** the number of tool in the temporary list is greater than 10:
  display the **Purchase Tool form** with error message “reduce the number of tool in the current Reservation to 10”.
- **Else**:
  pass all information in shopping for purchasing to **Purchase Summary Form** and display it with *current dates*, *total purchase price* and *information* about tools in the temporary list.

$Assume application keep $ToolID, $Description, $Rental price and $Deposit price of each tool in shopping cart. Then pop up **Purchase Summary Form** and display all those information. By the way, calculate $Total_Price are all calculated by application

When ***Submit*** button is clicked**:**

For each $ToolID in the list: {

```
UPDATE ToolForSale SET sold_date=CURDATE(), CustomerID = '$CustomerID'
WHERE ToolID = '$ToolID'
```

If err, reload the page

}

Display the **Purchase Confirmation Page** with all information in **Purchase Summary Form**

When ***Reset*** button is clicked:

Discard the temporary list for purchase in the memory, Display **Purchase Tool form**

Clerks

Clerk Main Menu

Abstract Code

- Show "***Pick-Up Reservation***", "***Drop-off Reservation***", "***Add New Tool***", "***Service Order***", "***Service Status***", "***Sell Tool***", "***Sale Status***", "***Generate Reports***" options/tabs.
- Upon:
  - Click ***Pick-Up Reservation*** button - jump to **View Pick-up Reservation** task.
  - Click ***Drop-off Reservation*** button - jump to **View Drop-off Reservation** task.
  - Click ***Add New Tool*** button - jump to **Add Tool form**.
  - Click ***Service Order*** button - jump to **Repair** task.
  - Click ***Service Status*** button - jump to **Check Service Status** task.
  - Click ***Sell Tool*** button - jump to **Mark for Sell** task
  - Click ***Sale Status*** button - jump to **View Sale Status** task
  - Click ***Generate Reports*** button - jump to **Select a Report** form.

Clerk Login

(see Customer login)

Pickup Reservation

Abstract Code

- Clerk clicked on ***Pick-Up Reservation*** button from **Clerk Main Menu**:
  - **View Reservation** to find Reservation not been picked up, and display Reservation ID, Customer username, Customer ID, start date and end date for for each tuple from below query.

```
SELECT reservation.reservationID, start_date, end_date, Customer.CustomerID,
username, CONCAT(first_name, last_name) AS customer_name, DATEDIFF(end_date,
start_date) as number_of_days, SUM(original_price * 0.4) AS total_deposit, SUM(
DATEDIFF(end_date, start_date) * original_price * 0.15) AS total_rental FROM
Reservation NATURAL JOIN Customer NATURAL JOIN AddRes NATURAL JOIN Tool WHERE
PickClerkID IS NULL GROUP BY reservation.reservationID ORDER BY
reservation.reservationID;
```

When reservationID is clicked, display reservationID, customer_name, total_rental, total_deposit from above query

- When the ***Pick Up*** button is clicked:
  If the *Reservation ID* entered is null or does not exist:
  display **Pickup Reservation** form with error message "please enter available Reservation"
  Else:
  Pass $reservationID, $customer_name, $total_rental, $total_deposit of entered reservationID in **Pickup Reservation page** to **Pick-up Reservation Confirmation page** and display this page.

  - **If** Existing Credit Card is selected:
    Do nothing.

    Else:
    Display **Pickup Reservation Confirmation** form with room for user to enter *card_number, name_on_card, month, year, cvc_3_digit_number*. If data type is invalid, ***Confirm Pick-up*** button is disabled.

    When ***Confirm Pick-up*** button is clicked,
    If New Credit Card is selected:{

```
UPDATE Customer SET card_number = '$card_number', name_on_card =
'$name_on_card', month = '$month', year = '$year', cvc_3_digit_number
= '$cvc_3_digit_number', WHERE CustomerID = '$CustomerID';
```

    If err, reload the page
    }

    Find Reservation with Reservation id, **update Reservation**

```
UPDATE Reservation SET PickClerkID = '$PickClerkID' WHERE
reservationID = '$reservationID';
```

    If err, reload the page

    Assume application can pass $reservationID to and display Rental contract after pick-up page

```
SELECT CONCAT(Clerk.first_name, Clerk.last_name) AS pick_up_clerk_name, CONCAT
(Customer.first_name, Customer.last_name) AS customer_name, card_number, start_date,
end_date, ToolID,  power_source, sub_option, sub_type, (0.15*original_price*
DATEDIFF(end_date,start_date)) AS rental_price, (0.4*original_price) AS deposit_price
FROM Reservation NATURAL JOIN AddRes NATURAL JOIN Tool NATURAL JOIN Customer
INNER JOIN Clerk ON reservation.PickClerkID = Clerk.ClerkID  WHERE reservationID =
'$reservationID';
```

$assume application can loop through the result, and calculate $total_rental_price and $total_deposit_price

when ***print contract*** button is clicked, this page is printed out, Short Description of tool(a link containing the $ToolID, $power_source, $sub_option, call helper method **display tool detail** when called)

Drop-off Reservation

Abstract Code

- Clerk clicked on ***Drop-off Reservation*** button from **Clerk Main Menu**:
- Run the **View Reservation** task:
  - Find all Reservations have been picked up and have not been dropped off, and display Reservation ID, Customer, Pick-up clerk, total deposit and total rental.

```
SELECT reservation.reservationID, start_date, end_date, Customer.CustomerID,
username, CONCAT(first_name, last_name) AS customer_name, DATEDIFF(end_date,
start_date) as number_of_days, SUM(original_price * 0.4) AS total_deposit, SUM(
DATEDIFF(end_date, start_date) * original_price * 0.15) AS total_rental FROM
Reservation NATURAL JOIN Customer NATURAL JOIN AddRes NATURAL JOIN Tool WHERE
PickClerkID IS NOT NULL AND DropClerkID IS NULL GROUP BY reservation.reservationID
ORDER BY reservation.reservationID;
```

  - When the ***Drop-off*** button is clicked:
    If the *Reservation ID* entered is null or does not exist:
    display **Drop-off Reservation** form with error message “please enter available Reservation”.
    Else:
    Pass $reservationID to **Drop-off Reservation Confirmation** form and run below query:

```
SELECT reservationID, ToolID, (0.15 * original_price*DATEDIFF(end_date,
start_date)) AS rental_price, (0.4*original_price) AS deposit_price,
```

```
CONCAT(first_name, last_name) AS customer_name, power_source,
sub_option, sub_type FROM Reservation NATURAL JOIN AddRes NATURAL
JOIN Tool NATURAL JOIN Customer WHERE reservationID = '$reservationID';
```

$assume application can loop through the result, and calculate $total_rental_price $total_deposit_price, and $balance_due = $total_rental_price - $total_deposit_price

display reservationID, customer_name, total_rental_price, total_deposit_price, balance_due and a list_of_tool containing $ToolID, $deposit_price, $rental_price, $short_description of tool(a link containing the $ToolID, $power_source, $sub_option, call helper method **display tool detail** when called)

When ***Drop-off*** button is clicked, update reservation information, check if rental time is equal or greater to 50 and add the tool for sale, and finally pop final receipt

```
UPDATE Reservation SET DropClerkID = '$DropClerkID' WHERE reservationID =
'$reservationID';
```

If err, reload the page

For each $ToolID in $list_of_tool: {

```
SELECT COUNT(*) AS rental_time FROM AddRes  WHERE ToolID = '$ToolID';
```

If $rental_time >= 50: {

```
INSERT INTO ToolForSale (for_sale_date, ToolID, ClerkID) VALUES (CURDATE(),
'$ToolID', 1);
```

If err, reload the page
1 represent Jill Watson, system's automated virtual clerk
}

}

Add Tool

Abstract Code

Display List

$Assume application allows user to enter only $type and $power_source at first:

If $type = 'hand_tool' and $power_source = 'manual':

sub-type dropdown is populated with 'Screwdriver', 'Socket', 'Ratchet', 'Wrench', 'Plier', 'Gun' and 'Hammer';

Else if $type = 'Garden' and $power_source = 'manual':

sub-type dropdown is populated with 'Digger', 'Pruner', 'Rakes', 'Wheelbarrows'and 'Striking';

Else if $type = 'Ladder' and $power_source = 'manual':

sub-type dropdown is populated with 'Straight' and 'Step';

Else if $type = 'Power' and $power_source = 'A/C':

sub-type dropdown is populated with 'Drill', 'Saw', 'Sander', 'Air-Compressor' and 'Mixer';

Else if $type = 'Power' and $power_source = 'Cordless':

sub-type dropdown is populated with 'Drill', 'Saw' and 'Sander'

Else if $type = 'Power' and $power_source = 'Gas':

sub-type dropdown is populated with Air-Compressor', 'Mixer' and 'Generator'

Else sub-type dropdown is disabled;

$When user chooses one $sub_type from the sub-type dropdown list:

Hand Tools

If $type = 'hand_tool', $power_source = 'manual' and $sub_type = 'Screwdriver':

sub-option dropdown is populated with 'phillips', 'hex', 'torx', 'slotted', and user chooses one as $sub_option

display enter fields $original_price, $manufacturer, $width, $width_fraction, $width_unit, $length, $length_fraction, $length_unit, $weight and $screw_size

when user entered all fields and click ***confirm*** button,

$Assume application can combine $width, $width_fraction, $width_unit into $Width and $length, $length_fraction, $length_unit into $Length

```
INSERT INTO Tool (original_price, power_source, sub_option, sub_type, width, length, weight, type) VALUES ('$original_price', '$power_source', '$sub_option', '$sub_type', '$Width', '$Length', '$weight', '$type');
```

$Assume application can keep surrogate key of new inserted Tool as $ToolID

```
INSERT INTO Screwdriver(ToolID, screw_size) VALUES ('$ToolID', '$screw_size');
```

ELSE If $type = 'hand_tool', $power_source = 'manual' and $sub_type = 'Rachet':

sub-option dropdown is populated with 'adjustable', 'fixed' and user chooses one as $sub_option

display enter fields $original_price, $manufacturer, $width, $width_fraction, $width_unit, $length, $length_fraction, $length_unit, $weight, $drive_size

when user entered all fields and click ***confirm*** button,

$Assume application can combine $width, $width_fraction, $width_unit into $Width and $length, $length_fraction, $length_unit into $Length

```
INSERT INTO Tool (original_price, power_source, sub_option, sub_type, width, length, weight, type) VALUES ('$original_price', '$power_source', '$sub_option', '$sub_type', $Width, '$Length', '$weight', '$type');
```

$Assume application can keep surrogate key of new inserted Tool as $ToolID

```
INSERT INTO Rachet(ToolID, drive_size) VALUES ('$ToolID', '$drive_size');
```

ELSE If $type = 'hand_tool', $power_source = 'manual' and $sub_type = 'Plier':

sub-option dropdown is populated with 'needle nose', 'cutting', 'crimper' and user chooses one as $sub_option

display enter fields $original_price, $manufacturer, $width, $width_fraction, $width_unit, $length, $length_fraction, $length_unit, $weight, $Adjustable

when user entered all fields and click *confirm* button,

$Assume application can combine $width, $width_fraction, $width_unit into $Width and $length, $length_fraction, $length_unit into $Length

```
INSERT INTO Tool (original_price, power_source, sub_option, sub_type, width, length, weight, type) VALUES ('$original_price', '$power_source', '$sub_option, '$sub_type', '$Width', '$Length', '$weight', '$type');
```

$Assume application can keep surrogate key of new inserted Tool as $ToolID

```
INSERT INTO Plier(ToolID, adjustable) VALUES ('$ToolID', '$Adjustable');
```

ELSE If $type = 'hand_tool', $power_source = 'manual' and $sub_type = 'Hammer':

sub-option dropdown is populated with 'claw', 'sledge', 'framing' and user chooses one as $sub_option

display enter fields $original_price, $manufacturer, $width, $width_fraction, $width_unit, $length, $length_fraction, $length_unit, $weight, $anti_vibration

when user entered all fields and click *confirm* button,

$Assume application can combine $width, $width_fraction, $width_unit into $Width and $length, $length_fraction, $length_unit into $Length

```
INSERT INTO Tool (original_price, power_source, sub_option, sub_type, width, length, weight, type) VALUES ('$original_price', '$power_source', '$sub_option', '$sub_type', $Width, '$Length', '$weight', '$type');
```

$Assume application can keep surrogate key of new inserted Tool as $ToolID

```
INSERT INTO Hammer(ToolID, anti_vibration) VALUES ('$ToolID', '$anti_vibration');
```

ELSE If $type = 'hand_tool', $power_source = 'manual' and $sub_type = 'Gun':

sub-option dropdown is populated with 'nail', 'staple' and user chooses one as $sub_option

display enter fields $original_price, $manufacturer, $width, $width_fraction, $width_unit, $length, $length_fraction, $length_unit, $weight, $capacity, $gauge_rating

when user entered all fields and click *confirm* button,

$Assume application can combine $width, $width_fraction, $width_unit into $Width and $length, $length_fraction, $length_unit into $Length

```
INSERT INTO Tool (original_price, power_source, sub_option, sub_type, width, length, weight, type) VALUES ('$original_price', '$power_source', '$sub_option', '$sub_type', '$Width', '$Length', '$weight', '$type');
```

$Assume application can keep surrogate key of new inserted Tool as $ToolID

```
INSERT INTO Gun(ToolID, capacity, gauge_rating) VALUES ('$ToolID', '$capacity', '$gauge_rating');
```

ELSE If $type = 'hand_tool', $power_source = 'manual' and $sub_type = 'Wrench':

sub-option dropdown is populated with 'crescent', 'torque', 'pipe' and user chooses one as $sub_option

display enter fields $original_price, $manufacturer, $width, $width_fraction, $width_unit, $length, $length_fraction, $length_unit, $weight

when user entered all fields and click *confirm* button,

$Assume application can combine $width, $width_fraction, $width_unit into $Width and $length, $length_fraction, $length_unit into $Length

```
INSERT INTO Tool (original_price, power_source, sub_option, sub_type, width, length, weight, type) VALUES ('$original_price', '$power_source', '$sub_option', '$sub_type', '$Width', '$Length', '$weight', '$type');
```

$Assume application can keep surrogate key of new inserted Tool as $ToolID

```
INSERT INTO Wrench (ToolID) VALUES ('$ToolID');
```

ELSE If $type = 'hand_tool', $power_source = 'manual' and $sub_type = 'Socket':

sub-option dropdown is populated with 'deep', 'standard' and user chooses one as $sub_option

display enter fields $original_price, $manufacturer, $width, $width_fraction, $width_unit, $length, $length_fraction, $length_unit, $weight, $drive_size, $deep_socket, $sae_size

when user entered all fields and click *confirm* button,

$Assume application can combine $width, $width_fraction, $width_unit into $Width and $length, $length_fraction, $length_unit into $Length

```
INSERT INTO Tool (original_price, power_source, sub_option, sub_type, width, length, weight, type) VALUES ('$original_price', '$power_source', '$sub_option', '$sub_type', '$Width', '$Length', '$weight', '$type', '$drive_size', '$deep_socket', '$sae_size');
```

$Assume application can keep surrogate key of new inserted Tool as $ToolID

```
INSERT INTO Socket (ToolID, drive_size, deep_socket, sae_size) VALUES ('$ToolID', '$drive_size', '$deep_socket', '$sae_size');
```

Garden Tools

ELSE If $type = 'garden_tool', $power_source = 'manual' and $sub_type = 'Wheelbarrows':

sub-option dropdown is populated with '1-wheel', '2-wheel' and user chooses one as $sub_option

display enter fields $original_price, $manufacturer, $width, $width_fraction, $width_unit, $length, $length_fraction, $length_unit, $weight, $handle_material, $bin_material, $bin_volume, $wheel_count

when user entered all fields and click *confirm* button,

$Assume application can combine $width, $width_fraction, $width_unit into $Width and $length, $length_fraction, $length_unit into $Length

```
INSERT INTO Tool (original_price, power_source, sub_option, sub_type, width, length, weight, type) VALUES ('$original_price', '$power_source', '$sub_option', '$sub_type', '$Width', '$Length', '$weight', '$type');
```

$Assume application can keep surrogate key of new inserted Tool as $ToolID

```
INSERT INTO Wheelbarrows (ToolID, handle_material, bin_material, bin_volume, wheel_count) VALUES ('$ToolID', '$handle_material', '$bin_material', '$bin_volume', '$wheel_count');
```

ELSE If $type = 'garden_tool', $power_source = 'manual' and $sub_type = 'Digger':

sub-option dropdown is populated with 'pointed shovel', 'flat shovel', 'scoop shovel', 'edger' and user chooses one as $sub_option

display enter fields $original_price, $manufacturer, $width, $width_fraction, $width_unit, $length, $length_fraction, $length_unit, $weight, $handle_material, $blade_material, $blade_length

when user entered all fields and click *confirm* button,

$Assume application can combine $width, $width_fraction, $width_unit into $Width and $length, $length_fraction, $length_unit into $Length

```
INSERT INTO Tool (original_price, power_source, sub_option, sub_type, width, length, weight, type) VALUES ('$original_price', '$power_source', '$sub_option', '$sub_type', '$Width', '$Length', '$weight', '$type');
```

$Assume application can keep surrogate key of new inserted Tool as $ToolID

```
INSERT INTO Digger (ToolID, handle_material, blade_material, blade_length) VALUES ('$ToolID', '$handle_material', '$blade_material', '$blade_length');
```

ELSE If $type = 'garden_tool', $power_source = 'manual' and $sub_type = 'Rakes':

sub-option dropdown is populated with 'leaf', 'landscaping', 'rock' and user chooses one as $sub_option

display enter fields $original_price, $manufacturer, $width, $width_fraction, $width_unit, $length, $length_fraction, $length_unit, $weight, $handle_material, $tine_count

when user entered all fields and click *confirm* button,

$Assume application can combine $width, $width_fraction, $width_unit into $Width and $length, $length_fraction, $length_unit into $Length

```
INSERT INTO Tool (original_price, power_source, sub_option, sub_type, width, length, weight, type) VALUES ('$original_price', '$power_source', '$sub_option', '$sub_type', '$Width', '$Length', '$weight', '$type');
```

$Assume application can keep surrogate key of new inserted Tool as $ToolID

```
INSERT INTO Rakes (ToolID, handle_material, tine_count) VALUES ('$ToolID',
'$handle_material', '$tine_count');
```

ELSE If $type = 'garden_tool', $power_source = 'manual' and $sub_type = 'Pruner':

sub-option dropdown is populated with 'sheer', 'loppers', 'hedge' and user chooses one as $sub_option

display enter fields $original_price, $manufacturer, $width, $width_fraction, $width_unit, $length, $length_fraction, $length_unit, $weight, $handle_material, $blade_material, $blade_length

when user entered all fields and click *confirm* button,

$Assume application can combine $width, $width_fraction, $width_unit into $Width and $length, $length_fraction, $length_unit into $Length

```
INSERT INTO Tool (original_price, power_source, sub_option, sub_type, width, length,
weight, type) VALUES ('$original_price', '$power_source', '$sub_option', '$sub_type',
'$Width', '$Length', '$weight', '$type');
```

$Assume application can keep surrogate key of new inserted Tool as $ToolID

```
INSERT INTO Pruner (ToolID, handle_material, blade_material, blade_length) VALUES
('$ToolID', '$handle_material', '$blade_material', '$blade_length');
```

ELSE If $type = 'garden_tool', $power_source = 'manual' and $sub_type = 'Striking':

sub-option dropdown is populated with 'bar pry', 'rubber mallet', 'tamper', 'pick axe', 'single bit axe' and user chooses one as $sub_option

display enter fields $original_price, $manufacturer, $width, $width_fraction, $width_unit, $length, $length_fraction, $length_unit, $weight, $handle_material, $head_weight

when user entered all fields and click *confirm* button,

$Assume application can combine $width, $width_fraction, $width_unit into $Width and $length, $length_fraction, $length_unit into $Length

```
INSERT INTO Tool (original_price, power_source, sub_option, sub_type, width, length, weight, type) VALUES ('$original_price', '$power_source', '$sub_option', '$sub_type', '$Width', '$Length', '$weight', '$type');
```

$Assume application can keep surrogate key of new inserted Tool as $ToolID

```
INSERT INTO Striking (ToolID, handle_material, head_weight) VALUES ('$ToolID', '$handle_material', '$head_weight');
```

Ladders

ELSE If $type = 'Ladder', $power_source = 'manual' and $sub_type = 'Straight':

sub-option dropdown is populated with 'rigid', 'telescoping' and user chooses one as $sub_option

display enter fields $original_price, $manufacturer, $width, $width_fraction, $width_unit, $length, $length_fraction, $length_unit, $weight, $step_count, $weight_capacity, $rubber_feet

when user entered all fields and click *confirm* button,

$Assume application can combine $width, $width_fraction, $width_unit into $Width and $length, $length_fraction, $length_unit into $Length

```
INSERT INTO Tool (original_price, power_source, sub_option, sub_type, width, length, weight, type) VALUES ('$original_price', '$power_source', '$sub_option', '$sub_type', '$Width', '$Length', '$weight', '$type');
```

$Assume application can keep surrogate key of new inserted Tool as $ToolID

```
INSERT INTO Straight (ToolID, step_count, weight_capacity, rubber_feet) VALUES ('$ToolID', '$step_count', '$weight_capacity', '$rubber_feet');
```

ELSE If $type = 'Ladder', $power_source = 'manual' and $sub_type = 'Step':

sub-option dropdown is populated with 'folding', 'multi-position' and user chooses one as $sub_option

display enter fields $original_price, $manufacturer, $width, $width_fraction, $width_unit, $length, $length_fraction, $length_unit, $weight, $step_count, $weight_capacity, $pail_shelf

when user entered all fields and click *confirm* button,

$Assume application can combine $width, $width_fraction, $width_unit into $Width and $length, $length_fraction, $length_unit into $Length

```
INSERT INTO Tool (original_price, power_source, sub_option, sub_type, width, length, weight, type) VALUES ('$original_price', '$power_source', '$sub_option', '$sub_type', '$Width', '$Length', '$weight', '$type');
```

$Assume application can keep surrogate key of new inserted Tool as $ToolID

```
INSERT INTO Step (ToolID, step_count, weight_capacity, pail_shelf) VALUES ('$ToolID', '$step_count', '$weight_capacity', '$pail_shelf');
```

Power Tools

ELSE If $type = 'power_tool', $power_source = 'A/C'' and $sub_type = 'Sander':

sub-option dropdown is populated with 'finish', 'sheet', 'belt', 'random orbital' and user chooses one as $sub_option

display enter fields $original_price, $manufacturer, $width, $width_fraction, $width_unit, $length, $length_fraction, $length_unit, $weight, $min_rpm_rating, $max_rpm_rating, $volt_rating, $amp_rating, $dust_bag, $AccessoryName, $accessory_quantity

when user entered all fields and click *confirm* button,

$Assume application can combine $width, $width_fraction, $width_unit into $Width, and $length, $length_fraction, $length_unit into $Length

```
INSERT INTO Tool (original_price, power_source, sub_option, sub_type, width, length, weight, type) VALUES ('$original_price', '$power_source', '$sub_option', '$sub_type', '$Width', '$Length', '$weight', '$type');
```

$Assume application can keep surrogate key of new inserted Tool as $ToolID

```
INSERT INTO PowerTool (ToolID, min_rpm_rating, max_rpm_rating, volt_rating,
amp_rating) VALUES ('$ToolID', '$min_rpm_rating', '$max_rpm_rating', '$volt_rating',
'$amp_rating');
```

If $AccessoryName.size() != 0: {
    For each $AccessoryName: {

```
INSERT INTO Accessory (ToolID, AccessoryName, accessory_quantity) VALUES ('$ToolID',
'$AccessoryName', '$accessory_quantity');
```

    }
}

```
INSERT INTO Sander(ToolID, dust_bag) VALUES ('$ToolID', '$dust_bag');
```

ELSE If $type = 'power_tool', $power_source = 'D/C'' and $sub_type = 'Sander':

sub-option dropdown is populated with 'finish', 'sheet', 'belt', 'random orbital' and user chooses one as $sub_option

display enter fields $original_price, $manufacturer, $width, $width_fraction, $width_unit, $length, $length_fraction, $length_unit, $weight, $min_rpm_rating, $max_rpm_rating, $volt_rating, $amp_rating, $dust_bag, $AccessoryName, $accessory_quantity, $battery_type

when user entered all fields and click *confirm* button,

$Assume application can combine $width, $width_fraction, $width_unit into $Width, and $length, $length_fraction, $length_unit into $Length

```
INSERT INTO Tool (original_price, power_source, sub_option, sub_type, width, length,
weight, type) VALUES ('$original_price', '$power_source', '$sub_option', '$sub_type',
'$Width', '$Length', '$weight', '$type');
```

$Assume application can keep surrogate key of new inserted Tool as $ToolID

```
INSERT INTO PowerTool (ToolID, min_rpm_rating, max_rpm_rating, volt_rating,
amp_rating) VALUES ('$ToolID', '$min_rpm_rating', '$max_rpm_rating', '$volt_rating',
'$amp_rating');
```

If $AccessoryName.size() != 0: {
    For each $AccessoryName: {

```
INSERT INTO Accessory (ToolID, AccessoryName, accessory_quantity) VALUES ('$ToolID', '$AccessoryName', '$accessory_quantity');
```

```
                    }
                }
```

```
INSERT INTO Cordless (battery_type) VALUES ('$battery_type');
```

$Assume application can keep surrogate key of new inserted cordless as $CordlessID

```
INSERT INTO Sander(ToolID, dust_bag, CordlessID) VALUES ('$ToolID', '$dust_bag', '$CordlessID');
```

ELSE If $type = 'power_tool', $power_source = 'A/C'' and $sub_type = 'Drill':

sub-option dropdown is populated with 'driver', 'hammer' and user chooses one as $sub_option

display enter fields $original_price, $manufacturer, $width, $width_fraction, $width_unit, $length, $length_fraction, $length_unit, $weight, $min_rpm_rating, $max_rpm_rating, $volt_rating, $amp_rating, $adjustable_clutch, $min_torque_rating, $max_torque_rating, $AccessoryName, $accessory_quantity

when user entered all fields and click ***confirm*** button,

$Assume application can combine $width, $width_fraction, $width_unit into $Width, and $length, $length_fraction, $length_unit into $Length

```
INSERT INTO Tool (original_price, power_source, sub_option, sub_type, width, length, weight, type) VALUES ('$original_price', '$power_source', '$sub_option', '$sub_type', '$Width', '$Length', '$weight', '$type');
```

$Assume application can keep surrogate key of new inserted Tool as $ToolID

```
INSERT INTO PowerTool (ToolID, min_rpm_rating, max_rpm_rating, volt_rating, amp_rating) VALUES ('$ToolID', '$min_rpm_rating', '$max_rpm_rating', '$volt_rating', '$amp_rating');
```

```
If $AccessoryName.size() != 0: {
        For each $AccessoryName: {
```

```
INSERT INTO Accessory (ToolID, AccessoryName, accessory_quantity) VALUES ('$ToolID',
'$AccessoryName', '$accessory_quantity');
```

```
                }
        }
```

```
INSERT INTO Drill(ToolID, adjustable_clutch, min_torque_rating, max_torque_rating)
VALUES ('$ToolID', '$adjustable_clutch', '$min_torque_rating', '$max_torque_rating');
```

ELSE If $type = 'power_tool', $power_source = 'D/C'' and $sub_type = 'Drill':

sub-option dropdown is populated with 'driver', 'hammer' and user chooses one as $sub_option

display enter fields $original_price, $manufacturer, $width, $width_fraction, $width_unit, $length, $length_fraction, $length_unit, $weight, $min_rpm_rating, $max_rpm_rating, $volt_rating, $amp_rating, $adjustable_clutch, $min_torque_rating, $max_torque_rating, $AccessoryName, $accessory_quantity, $battery_type

when user entered all fields and click *confirm* button,

$Assume application can combine $width, $width_fraction, $width_unit into $Width, and $length, $length_fraction, $length_unit into $Length

```
INSERT INTO Tool (original_price, power_source, sub_option, sub_type, width, length,
weight, type) VALUES ('$original_price', '$power_source', '$sub_option', '$sub_type',
'$Width', '$Length', '$weight', '$type');
```

$Assume application can keep surrogate key of new inserted Tool as $ToolID

```
INSERT INTO PowerTool (ToolID, min_rpm_rating, max_rpm_rating, volt_rating,
amp_rating) VALUES ('$ToolID', '$min_rpm_rating', '$max_rpm_rating', '$volt_rating',
'$amp_rating');
```

```
If $AccessoryName.size() != 0: {
        For each $AccessoryName: {
INSERT INTO Accessory (ToolID, AccessoryName, accessory_quantity) VALUES ('$ToolID',
'$AccessoryName', '$accessory_quantity');
                }
        }
```

```
INSERT INTO Cordless (battery_type) VALUES ('$battery_type');
```

$Assume application can keep surrogate key of new inserted cordless as $CordlessID

```
INSERT INTO Drill(ToolID, dust_bag, CordlessID) VALUES ('$ToolID', '$adjustable_clutch', '$min_torque_rating', '$max_torque_rating', '$CordlessID');
```

ELSE If $type = 'power_tool', $power_source = 'A/C'' and $sub_type = 'Saw':

sub-option dropdown is populated with 'circular', 'reciprocating', 'jig' and user chooses one as $sub_option

display enter fields $original_price, $manufacturer, $width, $width_fraction, $width_unit, $length, $length_fraction, $length_unit, $weight, $min_rpm_rating, $max_rpm_rating, $volt_rating, $amp_rating, $blade_size, $AccessoryName, $accessory_quantity

when user entered all fields and click *confirm* button,

$Assume application can combine $width, $width_fraction, $width_unit into $Width, and $length, $length_fraction, $length_unit into $Length

```
INSERT INTO Tool (original_price, power_source, sub_option, sub_type, width, length, weight, type) VALUES ('$original_price', '$power_source', '$sub_option', '$sub_type', '$Width', '$Length', '$weight', '$type');
```

$Assume application can keep surrogate key of new inserted Tool as $ToolID

```
INSERT INTO PowerTool (ToolID, min_rpm_rating, max_rpm_rating, volt_rating, amp_rating) VALUES ('$ToolID', '$min_rpm_rating', '$max_rpm_rating', '$volt_rating', '$amp_rating');
```

```
If $AccessoryName.size() != 0: {
        For each $AccessoryName: {
INSERT INTO Accessory (ToolID, AccessoryName, accessory_quantity) VALUES ('$ToolID', '$AccessoryName', '$accessory_quantity');
        }
}
```

```
INSERT INTO Saw(ToolID, blade_size) VALUES ('$ToolID', '$blade_size');
```

ELSE If $type = 'power_tool', $power_source = 'D/C'' and $sub_type = 'Saw':

sub-option dropdown is populated with 'circular', 'reciporacating', 'jig' and user chooses one as $sub_option

display enter fields $original_price, $manufacturer, $width, $width_fraction, $width_unit, $length, $length_fraction, $length_unit, $weight, $min_rpm_rating, $max_rpm_rating, $volt_rating, $amp_rating, $blade_size, $AccessoryName, $accessory_quantity, $battery_type

when user entered all fields and click *confirm* button,

$Assume application can combine $width, $width_fraction, $width_unit into $Width, and $length, $length_fraction, $length_unit into $Length

```
INSERT INTO Tool (original_price, power_source, sub_option, sub_type, width, length, weight, type) VALUES ('$original_price', '$power_source', '$sub_option', '$sub_type', '$Width', '$Length', '$weight', '$type');
```

$Assume application can keep surrogate key of new inserted Tool as $ToolID

```
INSERT INTO PowerTool (ToolID, min_rpm_rating, max_rpm_rating, volt_rating, amp_rating) VALUES ('$ToolID', '$min_rpm_rating', '$max_rpm_rating', '$volt_rating', '$amp_rating');
```

If $AccessoryName.size() != 0 {
For each $AccessoryName: {

```
INSERT INTO Accessory (ToolID, AccessoryName, accessory_quantity) VALUES ('$ToolID', '$AccessoryName', '$accessory_quantity');
```

}
}

```
INSERT INTO Cordless (battery_type) VALUES ('$battery_type');
```

$Assume application can keep surrogate key of new inserted cordless as $CordlessID

```
INSERT INTO Saw(ToolID, blade_size, CordlessID) VALUES ('$ToolID', '$blade_size',
'$CordlessID');
```

ELSE If $type = 'power_tool', $power_source = 'A/C'' and $sub_type = 'Air-compressor':

sub-option dropdown is populated with 'reciprocating' and user chooses one as $sub_option

display enter fields $original_price, $manufacturer, $width, $width_fraction, $width_unit, $length, $length_fraction, $length_unit, $weight, $min_rpm_rating, $max_rpm_rating, $volt_rating, $amp_rating, $tank_size, $pressure_rating, $AccessoryName, $accessory_quantity

when user entered all fields and click *confirm* button,

$Assume application can combine $width, $width_fraction, $width_unit into $Width, and $length, $length_fraction, $length_unit into $Length

```
INSERT INTO Tool (original_price, power_source, sub_option, sub_type, width, length,
weight, type) VALUES ('$original_price', '$power_source', '$sub_option', '$sub_type',
'$Width', '$Length', '$weight', '$type');
```

$Assume application can keep surrogate key of new inserted Tool as $ToolID

```
INSERT INTO PowerTool (ToolID, min_rpm_rating, max_rpm_rating, volt_rating,
amp_rating) VALUES ($'ToolID', '$min_rpm_rating', '$max_rpm_rating', '$volt_rating',
'$amp_rating');
```

```
If $AccessoryName.size() != 0 {
        For each $AccessoryName: {
INSERT INTO Accessory (ToolID, AccessoryName, accessory_quantity) VALUES ('$ToolID',,
'$AccessoryName', '$accessory_quantity');
        }
}
```

```
INSERT INTO AirCompressor(ToolID, tank_size, pressure_rating) VALUES ($ToolID,
'$tank_size', '$pressure_rating');
```

ELSE If $type = 'power_tool', $power_source = 'Gas'' and $sub_type = 'Air-compressor':

sub-option dropdown is populated with 'reciprocating' and user chooses one as $sub_option

display enter fields $original_price, $manufacturer, $width, $width_fraction, $width_unit, $length, $length_fraction, $length_unit, $weight, $min_rpm_rating, $max_rpm_rating, $volt_rating, $amp_rating, $tank_size, $pressure_rating, $AccessoryName, $accessory_quantity

when user entered all fields and click *confirm* button,

$Assume application can combine $width, $width_fraction, $width_unit into $Width, and $length, $length_fraction, $length_unit into $Length

```
INSERT INTO Tool (original_price, power_source, sub_option, sub_type, width, length, weight, type) VALUES ('$original_price', '$power_source', '$sub_option', '$sub_type', '$Width', '$Length', '$weight', '$type');
```

$Assume application can keep surrogate key of new inserted Tool as $ToolID

```
INSERT INTO PowerTool (ToolID, min_rpm_rating, max_rpm_rating, volt_rating, amp_rating) VALUES ('$ToolID', '$min_rpm_rating', '$max_rpm_rating', '$volt_rating', '$amp_rating');
```

```
If $AccessoryName.size() != 0 {
        For each $AccessoryName: {
INSERT INTO Accessory (ToolID, AccessoryName, accessory_quantity) VALUES ('$ToolID', '$AccessoryName', '$accessory_quantity');
        }
}
```

```
INSERT INTO AirCompressor(ToolID, tank_size, pressure_rating) VALUES ('$ToolID', '$tank_size', '$pressure_rating');
```

ELSE If $type = 'power_tool', $power_source = 'A/C'' and $sub_type = 'Mixer':

sub-option dropdown is populated with 'concrete' and user chooses one as $sub_option

display enter fields $original_price, $manufacturer, $width, $width_fraction, $width_unit, $length, $length_fraction, $length_unit, $weight, $min_rpm_rating, $max_rpm_rating, $volt_rating, $amp_rating, $motor_rating, $drum_size, $AccessoryName, $accessory_quantity

when user entered all fields and click *confirm* button,

$Assume application can combine $width, $width_fraction, $width_unit into $Width, and $length, $length_fraction, $length_unit into $Length

```
INSERT INTO Tool (original_price, power_source, sub_option, sub_type, width, length, weight, type) VALUES ('$original_price', '$power_source', '$sub_option', '$sub_type',, '$Width', '$Length', '$weight', '$type');
```

$Assume application can keep surrogate key of new inserted Tool as $ToolID

```
INSERT INTO PowerTool (ToolID, min_rpm_rating, max_rpm_rating, volt_rating, amp_rating) VALUES ('$ToolID', '$min_rpm_rating', $max_rpm_rating', '$volt_rating', '$amp_rating');
```

If $AccessoryName.size() != 0 {
    For each $AccessoryName: {

```
INSERT INTO Accessory (ToolID, AccessoryName, accessory_quantity) VALUES ('$ToolID', '$AccessoryName', '$accessory_quantity');
```

    }
}

```
INSERT INTO Mixer(ToolID, motor_rating, drum_size) VALUES ('$ToolID', '$motor_rating', '$drum_size');
```

ELSE If $type = 'power_tool', $power_source = 'Gas'' and $sub_type = 'Mixer':

sub-option dropdown is populated with 'concrete' and user chooses one as $sub_option

display enter fields $original_price, $manufacturer, $width, $width_fraction, $width_unit, $length, $length_fraction, $length_unit, $weight, $min_rpm_rating, $max_rpm_rating, $volt_rating, $amp_rating, $motor_rating, $drum_size, $AccessoryName, $accessory_quantity

when user entered all fields and click *confirm* button,

$Assume application can combine $width, $width_fraction, $width_unit into $Width, and $length, $length_fraction, $length_unit into $Length

```
INSERT INTO Tool (original_price, power_source, sub_option, sub_type, width, length, weight, type) VALUES ('$original_price', '$power_source', '$sub_option', '$sub_type', '$Width', '$Length', '$weight', '$type');
```

$Assume application can keep surrogate key of new inserted Tool as $ToolID

```
INSERT INTO PowerTool (ToolID, min_rpm_rating, max_rpm_rating, volt_rating, amp_rating) VALUES ('$ToolID', '$min_rpm_rating', '$max_rpm_rating', '$volt_rating', '$amp_rating');
```

If $AccessoryName.size() != 0 {
    For each $AccessoryName: {

```
INSERT INTO Accessory (ToolID, AccessoryName, accessory_quantity) VALUES ('$ToolID', '$AccessoryName', '$accessory_quantity');
```

    }
}

```
INSERT INTO Mixer(ToolID, motor_rating, drum_size) VALUES ('$ToolID', '$motor_rating', '$drum_size');
```

ELSE If $type = 'power_tool', $power_source = 'A/C'' and $sub_type = 'Generator':

sub-option dropdown is populated with 'electric' and user chooses one as $sub_option

display enter fields $original_price, $manufacturer, $width, $width_fraction, $width_unit, $length, $length_fraction, $length_unit, $weight, $min_rpm_rating, $max_rpm_rating, $volt_rating, $amp_rating, $power_rating, $AccessoryName, $accessory_quantity

when user entered all fields and click *confirm* button,

$Assume application can combine $width, $width_fraction, $width_unit into $Width, and $length, $length_fraction, $length_unit into $Length

```
INSERT INTO Tool (original_price, power_source, sub_option, sub_type, width, length, weight, type) VALUES ('$original_price', '$power_source', '$sub_option', '$sub_type', '$Width', '$Length', '$weight', '$type');
```

$Assume application can keep surrogate key of new inserted Tool as $ToolID

```
INSERT INTO PowerTool (ToolID, min_rpm_rating, max_rpm_rating, volt_rating,
amp_rating) VALUES ('$ToolID', '$min_rpm_rating', '$max_rpm_rating', '$volt_rating',
'$amp_rating');
```

```
If $AccessoryName.size() != 0 {
        For each $AccessoryName: {
INSERT INTO Accessory (ToolID, AccessoryName, accessory_quantity) VALUES ('$ToolID',
'$AccessoryName', '$accessory_quantity');
        }
}
```

```
INSERT INTO Generator(ToolID, power_rating) VALUES ('$ToolID', '$power_rating');
```

Repair Tool

Abstract Code

Assum application keep $ClerkID

Include **Basic Keyword Search Function**, wait user for entering *Start Date, End Date, Keyword, Type, Power Source, SubType*

When ***Search*** button is Clicked do the following, then: Search function return $type, $power_source, $sub_type, $keyword, and $start_date, $end_date from user entered

- Call **Check Tool Availability** task to display each available tool's $Tool ID, $Description(a link containing the $ToolID, $power_source, $sub_option, call helper method **display tool detail** when called), $Rental price and $Deposit price

Wait for clert to enter $ToolID and $service_cost
When ***Confirm*** button is Clicked do the following:

- **If** data validation is successful:
  - If tool is unavailable: (in case other clerks submit service order for this tool before you hit ***confirm*** button)
    Display Repair Tool form with error message
  - **Else:**
    Add Service: add a new service order into the database

```
INSERT INTO ServiceOrder ( service_start_date, service_end_date,
repair_cost, ClerkID, ToolID) VALUES ('$service_start_date',
'$service_end_date','$repair_cost', '$ClerkID', '$ToolID');
```

If err, reload the page

Display **Repair Tool form.**

- Else:
  Display Repair Tool form with error message

View/Filter/Override Service Status

Abstract Code

Include **Advance Keyword Search Function,** wait user for entering *Keyword, Type*

when user click search button, application can get $type, $clerk_username, $sub_option, $min_cost, $max_cost

Find service order matching the entered $type, $clerk_username, $sub_option, $min_cost, $max_cost. Display Service ID, Status(All is in-repair), ToolID, Short Description(a link containing the $ToolID, $power_source, $sub_option, call helper method **display tool detail** when called), service_start_date, service_end_date, repair cost and clerk username.

```
SELECT ServiceOrderID, ToolID,  power_source, sub_option, sub_type, service_start_date,
service_end_date, repair_cost, username FROM ServiceOrder NATURAL JOIN Tool
NATURAL JOIN Clerk WHERE service_end_date > CURDATE() AND type LIKE '%$type%'
AND username LIKE '%$clerk_username%' AND repair_cost >= $min_cost AND repair_cost
<= $max_cost AND sub_option LIKE '%$sub_option%';
```

When ***Fix Now*** button is Clicked, application can get the $ServiceOrderID and do the following:

Update service order: change service order end date to now,

```
UPDATE ServiceOrder SET service_end_date = CURDATE() WHERE
ServiceOrderID = '$ServiceOrderID';
```

If err, reload the page

Sell Tool

Abstract Code

Assume application keep the $ClerkID

Include **Basic Keyword Search Function**, wait user for entering *Sale Date, Keyword, Type, Power Source, SubType*

When ***Search*** button is Clicked, then: Search function return $type, $power_source, $sub_type, $keyword, and $start_date, $end_date from user entered. Do the following:

- Call **Check Tool Availability** task to display each available tool's $Tool ID, $Description(a link containing the $ToolID, $power_source, $sub_option, call helper method **display tool detail** when called), $Rental price and $Deposit price

When ***Sell Tool*** button is Clicked, application can get the $ToolID and do the following:

Update tool. Change tool's for-sale date to now.

```
INSERT INTO ToolForSale (for_sale_date, ClerkID,ToolID) VALUES (CURDATE(), '$ClerkID', '$ToolID');
```

If err, reload the page

View Sale Status

Abstract Code

Include **Advance Keyword Search Function**, wait user for entering *Keyword, Type*

When ***Search*** button is Clicked do the following: application can get $type, $clerk_username, $sub_option, $min_sale, $max_sale

View Tools in Sold or For-Sale:
Find service order matching the entered $type, $customer_username, $sub_option, $min_sale, $max_sale. Display Sale ID, Current Status, ToolID, Short Description(a link containing the $ToolID, $power_source, $sub_option, call helper method **display tool detail** when called), Customer_username, sale_date and ClerkID.

```
SELECT SaleID, ClerkID, ToolID, Customer.username, (0.5*original_price) AS sale_price, power_source, sub_option, sub_type, CASE WHEN sold_date IS NOT NULL THEN sold_date ELSE for_sale_date END AS date, CASE WHEN sold_date IS NOT NULL THEN 'Sold' ELSE 'For-Sale' END AS current_status FROM ToolForSale NATURAL JOIN Tool NATURAL JOIN Clerk LEFT OUTER JOIN Customer ON ToolForSale.CustomerID = Customer.CustomerID  WHERE type LIKE '%$type%' AND Customer.username LIKE '%$customer_username%' AND 0.5*original_price >= $min_sale AND 0.5*original_price <= $max_sale AND sub_option LIKE '%$sub_option%';
```

Generate Reports

Abstract Code

- Show "***Clerk Report***", "***Customer Report***", "***Tool Inventory Report***" options/tabs.
- Upon:
    - Click ***Clerk Report*** button - jump to **Generate Reports** task.
    - Click ***Customer Report*** button - jump to **Generate Clerk Report** task.
    - Click ***Tool Inventory Report*** button - jump to **Generate Customer Report** task.

Generate Clerk Report

Abstract Code

Run **View Clerk** task

- For each clerk, display Clerk ID, First Name, Middle Name, Last Name, Email, Hire Date

Run **View Reservation** task:

- Group Reservation by pick up clerk/ drop-off clerk attribute, count how many Reservation is picked up/dropped off by that clerk from this month

```
SELECT ClerkID, first_name, middle_name, last_name, email, hire_date,
COUNT(R1.PickClerkID) AS number_of_pickups, COUNT(R2.DropClerkID) AS
number_of_dropoffs,  COUNT(R1.PickClerkID) + COUNT(R2.DropClerkID) AS total FROM
Clerk LEFT OUTER JOIN Reservation AS R1 ON R1.PickClerkID =Clerk.ClerkID LEFT OUTER
JOIN Reservation AS R2 ON R2.DropClerkID = Clerk.ClerkID WHERE MONTH(R1.start_date)
= MONTH(CURDATE()) OR MONTH(R2.end_date) = MONTH(CURDATE()) GROUP BY
Clerk.ClerkID;
```

When ***Back to Report Menu*** button is clicked:
    Display **Select a Report** page
When ***Reload Results*** is clicked:
    Reload **Clerk Report** page

Generate Customer Report

Abstract Code

Run **View Customer** task:
For each customer, display Customer ID, First Name, Middle Name, Last Name, Email, Primary phone, group Reservation by customer attribute, and count how many Reservation and how many tools are rented by that customer in last month and display them

```
SELECT Customer.CustomerID, first_name, middle_name, last_name, email,
CONCAT(area_code, '-', phone_number, '-', extension) AS Phone,
COUNT(Reservation.reservationID) AS total_reservation,  COUNT(AddRes.ToolID) AS
total_tools_rented FROM  Customer NATURAL JOIN Phone LEFT OUTER JOIN Reservation
ON Customer.CustomerID = Reservation.CustomerID LEFT OUTER JOIN AddRes ON
Reservation.reservationID = AddRes.reservationID WHERE (MONTH(start_date) =
(MONTH(CURDATE()) - 1) AND YEAR(start_date) = YEAR(CURDATE())) OR
(MONTH(start_date)= 12 AND MONTH(CURDATE()) = 1 AND
YEAR(start_date)=(YEAR(CURDATE()) - 1))  GROUP BY Customer.CustomerID;
```

- **If *View Profile*** button(which contains $CustomerID) is clicked, call **View Profile**

When ***Back to Report Menu*** button is clicked:
    Display **Select a Report** page
When ***Reload Results*** is clicked:
    Reload **Customer Report** page

Generate Tool Report

Abstract Code

Include **Advance Keyword Search Function**, wait user for entering *Keyword, Type*

when user click search button, application can get $type, $sub_option

Find Tools matching the clerk's input (Type, keyword)

```
SELECT ToolID FROM Tool WHERE type LIKE '%$type%' AND sub_option LIKE
'%$sub_option%';
```

Answer = []; (array, JavaScript)

Assume application can get a $list_of_ToolID from above query

For each $ToolID in the $list_of_ToolID: {

    Find all reservation record about for this $ToolID
    result1 =

```
SELECT DATEDIFF(end_date, start_date) AS number_of_days, DATEDIFF(end_date, start_date)*0.15*original_price AS rental_profit FROM Tool NATURAL JOIN AddRes NATURAL JOIN Reservation WHERE ToolID = '$ToolID';
```

Find all sale record about for this $ToolID
result2 =

```
SELECT for_sale_date, sold_date FROM Tool NATURAL JOIN ToolForSale WHERE ToolID = '$ToolID';
```

Find all service record about for this $ToolID
result3 =

```
SELECT repair_cost, service_end_date FROM Tool NATURAL JOIN ServiceOrder WHERE ToolID = '$ToolID' ORDER BY service_end_date DESC ;
```

Find tool information about for this $ToolID
result4 =

```
SELECT original_price, power_source, sub_option, sub_type FROM Tool WHERE ToolID = '$ToolID';
```

```
temp = {}
$total_rental_price = 0;
$total_cost = result4[0].original_price
$sale_profit = 0;

//calculate total rental profit
If result1.size() != 0 {
        For each $tuple in result1: {
                $total_rental_price +=$tuple.rental_profit;
        }
}

//calculate total repair cost
If result3.size() != 0 {
        For each $tuple in result3: {
                $total_cost += $tuple.repair_cost;
        }
}
```

```
        // first judge in sale or not, then judge if in repair or not, last judge if in-rent or not
        If result2.size() != 0 {
                If result2[0].CustomerID == null {
                        temp.current_status = 'For-Sale';
                        temp.date = result2[0].for_sale_date;
                } Else {
                        temp.current_status = 'Sold';
                        temp.date = result2[0].sold_date;
                        $sale_profit += 0.5 *  result4[0].original_price;
                }
        Else if  result3.size()! = 0 && result3[0].service_end_date > $current_date: {
        temp.current_status = 'In-Repair';
                        temp.date = result3[0].service_end_date;
                }
        Else if  result1.size()!=0 {
                For each $tuple in result1 {
                        if  $tuple.start_date <= $current_date <=  $tuple.end_date {
                                temp.current_status = 'Rented'
                                temp.date = $tuple.end_date;
                                break;
                        }
                }
        }
        Else {
                temp.current_status = 'Available'
                temp.date = null
        }

        temp.rental_profit = $total_rental_price
        temp.total_cost = $total_cost
        temp.total_profit = $total_rental_price + $sale_profit - $total_cost
        temp.description = result4[0].Description
        temp.toolID = $ToolID

        Answer.add(temp);
}
```

Then pass Answer to display page, and display required information.

[Helper-method]

Display tool detail

Abstract Code

$Assume application can get $ToolID and $sub_option information, display ToolID, Tool Type, Short Description, Full Description, Deposit price, Rental_price

If $sub_option = 'Sander'

```
SELECT weight, length, width, material, manufacturer, (0.4*original_price) AS deposit_price, (0.15*original_price) AS rental_price, power_source, sub_option, sub_type, min_rpm_rating, max_rpm_rating, volt_rating, amp_rating, AccessoryName, accessory_quantity, dust_bag, battery_type FROM Tool NATURAL JOIN PowerTool NATURAL JOIN Sander LEFT OUTER JOIN Accessory ON PowerTool.ToolID = Accessory.ToolID LEFT OUTER JOIN Cordless ON Cordless.CordlessID = Sander.CordlessID WHERE Tool.ToolID = '$ToolID';
```

Else if $sub_option = 'Saw'

```
SELECT weight, length, width, material, manufacturer, (0.4*original_price) AS deposit_price, (0.15*original_price) AS rental_price, power_source, sub_option, sub_type, min_rpm_rating, max_rpm_rating, volt_rating, amp_rating, AccessoryName, accessory_quantity, blade_size, battery_type FROM Tool NATURAL JOIN PowerTool NATURAL JOIN Saw LEFT OUTER JOIN Accessory ON PowerTool.ToolID = Accessory.ToolID LEFT OUTER JOIN Cordless ON Cordless.CordlessID = Saw.CordlessID WHERE Tool.ToolID = '$ToolID';
```

Else if $sub_option = 'Drill'

```
SELECT weight, length, width, material, manufacturer,  (0.4*original_price) AS deposit_price, (0.15*original_price) AS rental_price, power_source, sub_option, sub_type, min_rpm_rating, max_rpm_rating, volt_rating, amp_rating, AccessoryName, accessory_quantity, adjustable_clutch, min_torque_rating, max_torque_rating, battery_type FROM Tool NATURAL JOIN PowerTool NATURAL JOIN Drill LEFT OUTER JOIN Accessory ON PowerTool.ToolID = Accessory.ToolID LEFT OUTER JOIN Cordless ON Cordless.CordlessID = Drill.CordlessID WHERE Tool.ToolID = '$ToolID';
```

Else if $sub_option = 'Air_compressor'

```
SELECT weight, length, width, material, manufacturer,  (0.4*original_price) AS
deposit_price, (0.15*original_price) AS rental_price, power_source, sub_option,
sub_type, min_rpm_rating, max_rpm_rating, volt_rating, amp_rating, AccessoryName,
accessory_quantity, tank_size, pressure_rating FROM Tool NATURAL JOIN PowerTool
NATURAL JOIN AirCompressor LEFT OUTER JOIN Accessory ON PowerTool.ToolID =
Accessory.ToolID WHERE Tool.ToolID = '$ToolID';
```

Else if $sub_option = 'Mixer'

```
SELECT weight, length, width, material, manufacturer,  (0.4*original_price) AS
deposit_price, (0.15*original_price) AS rental_price, power_source, sub_option,
sub_type, min_rpm_rating, max_rpm_rating, volt_rating, amp_rating, AccessoryName,
accessory_quantity, motor_rating, dum_size FROM Tool NATURAL JOIN PowerTool
NATURAL JOIN Mixer LEFT OUTER JOIN Accessory ON PowerTool.ToolID =
Accessory.ToolID WHERE Tool.ToolID = '$ToolID';
```

Else if $sub_option = 'Generator'

```
SELECT weight, length, width, material, manufacturer,  (0.4*original_price) AS
deposit_price, (0.15*original_price) AS rental_price, power_source, sub_option,
sub_type, min_rpm_rating, max_rpm_rating, volt_rating, amp_rating, AccessoryName,
accessory_quantity, power_rating FROM Tool NATURAL JOIN PowerTool NATURAL JOIN
Generator LEFT OUTER JOIN Accessory ON PowerTool.ToolID = Accessory.ToolID WHERE
Tool.ToolID = '$ToolID';
```

Else if $sub_option = 'Step'

```
SELECT weight, length, width, material, manufacturer,  (0.4*original_price) AS
deposit_price, (0.15*original_price) AS rental_price, power_source, sub_option,
sub_type, step_count, weight_capacity, pail_shelf FROM Tool NATURAL JOIN Step
WHERE Tool.ToolID = '$ToolID';
```

Else if $sub_option = 'Straight'

```
SELECT weight, length, width, material, manufacturer,  (0.4*original_price) AS
deposit_price, (0.15*original_price) AS rental_price, power_source, sub_option,
sub_type, step_count, weight_capacity, rubber_feet FROM Tool NATURAL JOIN Straight
WHERE Tool.ToolID = '$ToolID';
```

Else if $sub_option = 'Screwdriver'

```
SELECT weight, length, width, material, manufacturer,  (0.4*original_price) AS
deposit_price, (0.15*original_price) AS rental_price, power_source, sub_option,
sub_type, screw_size FROM Tool NATURAL JOIN Screwdriver WHERE Tool.ToolID =
'$ToolID';
```

Else if $sub_option = 'Rachet'

```
SELECT weight, length, width, material, manufacturer,  (0.4*original_price) AS
deposit_price, (0.15*original_price) AS rental_price, power_source, sub_option,
sub_type, drive_size FROM Tool NATURAL JOIN Rachet WHERE Tool.ToolID = '$ToolID';
```

Else if $sub_option = 'Plier'

```
SELECT weight, length, width, material, manufacturer, (0.4*original_price) AS
deposit_price, (0.15*original_price) AS rental_price, power_source, sub_option,
sub_type, adjustable FROM Tool NATURAL JOIN Plier WHERE Tool.ToolID = '$ToolID';
```

Else if $sub_option = 'Hammer'

```
SELECT weight, length, width, material, manufacturer, (0.4*original_price) AS
deposit_price, (0.15*original_price) AS rental_price, power_source, sub_option,
sub_type, anti_vibration FROM Tool NATURAL JOIN Hammer WHERE Tool.ToolID =
'$ToolID';
```

Else if $sub_option = 'Gun'

```
SELECT weight, length, width, material, manufacturer, (0.4*original_price) AS
deposit_price, (0.15*original_price) AS rental_price, power_source, sub_option,
sub_type, capacity, gauge_rating FROM Tool NATURAL JOIN Gun WHERE Tool.ToolID =
'$ToolID';
```

Else if $sub_option = 'Socket'

```
SELECT weight, length, width, material, manufacturer, (0.4*original_price) AS
deposit_price, (0.15*original_price) AS rental_price, power_source, sub_option,
sub_type, sae_size, drive_size, deep_socket FROM Tool NATURAL JOIN Socket WHERE
Tool.ToolID = '$ToolID';
```

Else if $sub_option = 'Wrench'

```
SELECT weight, length, width, material, manufacturer, (0.4*original_price) AS
deposit_price, (0.15*original_price) AS rental_price, power_source, sub_option,
sub_type FROM Tool NATURAL JOIN Wrench WHERE Tool.ToolID = '$ToolID';
```

Else if $sub_option = 'Striking'

```
SELECT weight, length, width, material, manufacturer, (0.4*original_price) AS
deposit_price, (0.15*original_price) AS rental_price, power_source, sub_option,
sub_type, handle_material, head_weight FROM Tool NATURAL JOIN Striking WHERE
Tool.ToolID = '$ToolID';
```

Else if $sub_option = 'Pruner'

```
SELECT weight, length, width, material, manufacturer, (0.4*original_price) AS
deposit_price, (0.15*original_price) AS rental_price, power_source, sub_option,
sub_type, handle_material, blade_material, blade_length FROM Tool NATURAL JOIN
Pruner WHERE Tool.ToolID = '$ToolID';
```

Else if $sub_option = 'Rakes'

```
SELECT weight, length, width, material, manufacturer, (0.4*original_price) AS
deposit_price, (0.15*original_price) AS rental_price, power_source, sub_option,
sub_type, handle_material, tine_count FROM Tool NATURAL JOIN Rakes WHERE
Tool.ToolID = '$ToolID';
```

Else if $sub_option = 'Digger'

```
SELECT weight, length, width, material, manufacturer, (0.4*original_price) AS
deposit_price, (0.15*original_price) AS rental_price, power_source, sub_option,
```

```
sub_type, handle_material, blade_material, blade_length FROM Tool NATURAL JOIN
Digger WHERE Tool.ToolID = '$ToolID';
```

Else if $sub_option = 'Wheelbarrows'

```
SELECT weight, length, width, material, manufacturer, (0.4*original_price) AS
deposit_price, (0.15*original_price) AS rental_price, power_source, sub_option,
sub_type, handle_material, bin_material, bin_volume, wheel_count FROM Tool
NATURAL JOIN Wheelbarrows WHERE Tool.ToolID = '$ToolID';
```

Basic Keyword Search Function

If Customer does not select one *type* from (All tools, Hand Tool, Garden Tool, Ladder, Power Tool):
    Power Source and Sub-Type Drop Down, and ***Search*** button are disabled
Else:
    power source is populated with hard-code list.
    If Customer does not select one type from power source drop down:
        Sub-Type Drop Down, and ***Search*** button are disabled
    Else:
        Sub-Type Drop Down is populated with hard code list (based on type and power source)
        If customer does not select one type from sub type drop down:
            Search button is disabled
Return $type, $power_source, $sub_type, $keyword

Advance Keyword Search Function

Assume application can get $clerk_username, $sub_option, $min_cost, $max_cost and from keyword that customer input, and $type that user choose