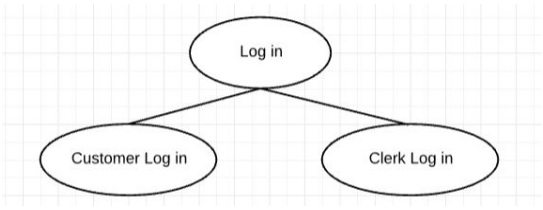
Team33_task designs

1. Task Decomposition and abstract code

   1.1 Login




```
//  if user chooses Clerk, read $Login and $Password
// if the query result is empty, the application reminds no this clerk; if the query result is not empty, and  Password == $Password, the application jumps to the clerk main menu; if the query result is not empty, and  Password != $Password, ask for input password again
//Using php to report not exist of user or password does not match
$query = mysql_query("Select username from clerk where username = '$username'")
if (mysql_num_rows($query)=0){
        echo 'user not exists!';}
$query_ps = mysql_query("Select username from clerk where username = '$username' and password = '$password'")
if (mysql_num_rows($query)>0 && mysql_num_rows($query_ps) =0{
        echo 'password is not correct';}


SELECT UserName, Fname, Lname
FROM CLERK
WHERE Username = $Login AND Password = $Password;

// if user chooses Customer, read $Login and $Password
// if the query result is empty, the application pops up create a profile screen;  if the query result is non-empty,  and Password == $Password, the application jumps to the customer main menu; if the query result is non-empty,  but  Password != $Password, the application prompts user wrong password and asks to  input the password again
//use php to check not exists of user or password not correct
$query = mysql_query("Select username from customer where username = '$username'")
if (mysql_num_rows($query)=0){
        echo 'user not exists!';}

$query_ps = mysql_query("Select username from customer where username = '$username' and password = '$password'")
if (mysql_num_rows($query)>0 && mysql_num_rows($query_ps) =0{
```

echo 'password is not correct';}

SELECT UserName, Fname, Lname
FROM CUSTOMER
WHERE Username = $Login AND Password = $Password;

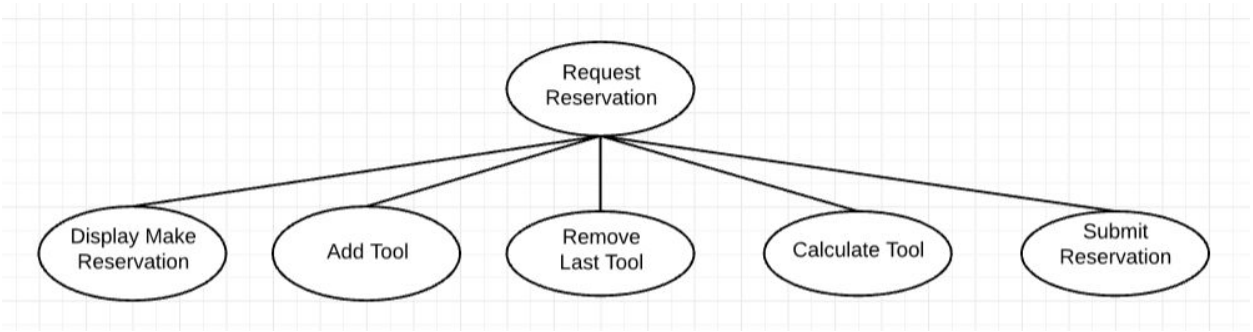
1.2 Create Profile

// The application read $Login, $Password, $Confirmed_password, $First_name, $Last_name, $Home_phone_area_code, $Home_phone_local_number, and $Work_phone_area_code, $Work_phone_local_number, $Address.

// When user click Submit, check whether $Password equal to $Confirm_password. If yes, move to the SQL command below; if not, prompt the user Conirm_password is not matched with Password, and input the Password and Confirm_password again.

INSERT INTO
CUSTOMER (Login, Password, Fname, Lname, Work_area_code, Work_local_number, Home_area_code, Home_local_number, Address)
VALUES ($Login, $Password, $First_name, $Last_name, $Work_phone_area_code, $Work_phone_local_number, $Home_phone_area_code, $Home_phone_local_number, $Address);

1.3 Request Reservation



//populate *Type of Tool* dropdowns
//read $Tool_Type, populate *Tool* dropdowns, if tool is not available, it won't be shown.
SELECT Tool_ID, Abbrdesc, Rental
FROM TOOL
WHERE is_Sold = 'N' **AND**
Tool_Type = $Tool_Type **AND**
tool_ID **NOT IN**
(SELECT tool.tool_id **FROM** tool, reserve, reservation

WHERE reservation.start_date< CURDATE() and reservation.end_date>CURDATE() and reserve.reservation_number = reservation.reservation_number and tool.tool_id = reserve.tool_id) **AND**
tool_id **NOT IN**
(**SELECT** tool_id from service_order
WHERE start_date< CURDATE() and end_date>CURDATE());

//If *Add More Tools* button is pushed, add a new line and populate *Type of Tool* dropdowns
//read $Tool_Type, populate *Tool* dropdowns,if tool is not available, it won't be shown.
SELECT Tool_ID, Abbrdesc, Rental
FROM TOOL
WHERE is_Sold = 'N' **AND**
Tool_Type = $Tool_Type **AND**
tool_ID **NOT IN**
(**SELECT** tool.tool_id **FROM** tool, reserve, reservation
WHERE reservation.start_date< CURDATE() and reservation.end_date>CURDATE() and reserve.reservation_number = reservation.reservation_number and tool.tool_id = reserve.tool_id) **AND**
tool_id **NOT IN**
(**SELECT** tool_id from service_order
WHERE start_date< CURDATE() **AND** end_date>CURDATE());

//If *Remove Last Tool* button is pushed, remove a line of Tool that was lastly added
//If *Calculate Total* button is pushed, read $Start_date, $End_date, $Tool_IDs is the collection of tool being selected.
//calculate Total Rental Price and Total Deposit Required based on $Start_date, $End_date and Rental and Deposit of $Tool_IDs.
SELECT abbrdesc **FROM TOOL**
WHERE tool_id in ($tool_ids);

SELECT sum(rental*(DATEDIFF($end_date,$start_date))), sum(deposit)
FROM TOOL
WHERE tool_id in ($tool_ids);

//If *Submit* button is pushed
//check the number of tools in the reservation first, if more than 50, stop and remind user that only up to 50 tools per reservation.
//if the number of tools in the reservation <= 50, generate a unique reservation No. as $reservation_number and assume customer username as $customer_username

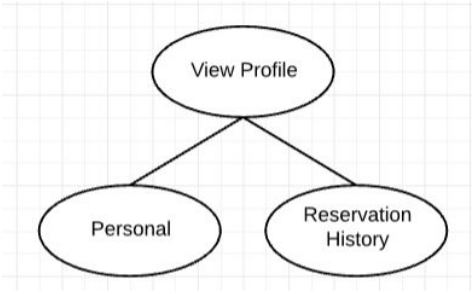
INSERT INTO RESERVATION (reservation_number, Start_date, End_date, Request_customer, Pickup_clerk, Dropoff_clerk, Pickup_creditcard_number, Pickup_creditcard_expdate)
VALUES ($reservation_number, $Start_date, $End_date, $Custtomer_Username, NULL, NULL, NULL, NULL);

for each $Tool_ID
INSERT INTO RESERVE (Reservation_number, Tool_ID)
VALUES ($RNumber, $Tool_ID);
end for

//show customer reservation summary,assume reservation number is $reservation_number

```
SELECT R.reservation_number, R.start_date, R.end_date, T.abbrdesc
from Reservation as R
inner join reserve on
R.reservation_number = reserve.reservation_number
inner join tool as T on
T.Tool_id = reserve.tool_ID
where R.reservation_number = $reservation_number;

SELECT sum(T.rental*(DATEDIFF(R.end_date,R.start_date))), sum(T.deposit)
from Reservation as R
inner join reserve on
R.reservation_number = reserve.reservation_number
inner join tool as T on
T.Tool_id = reserve.tool_ID
where R.reservation_number = $reservation_number;
```

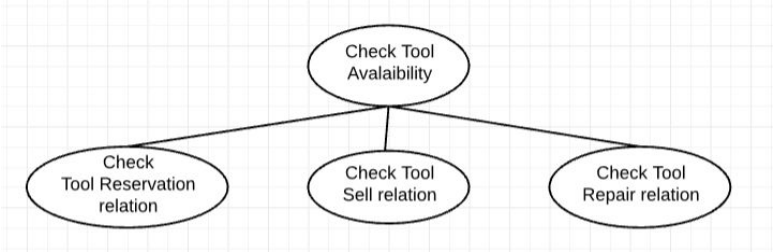
1.4 View Profile



//Application provide $Username
//find personal information
SELECT Username, Fname, Lname, Home_area_code, Home_local_number, Work_area_code, Work_local_number, Address
FROM CUSTOMER
WHERE Username = $customer_Username;

//Application provide $customer_Username
//find reservation history
SELECT R.reservation_number, T.abbrdesc, R.start_date, R.end_date,
T.rental*(DATEDIFF(R.end_date,R.start_date)), T.deposit, R.Pickup_clerk, R.Dropoff_clerk
FROM Reservation **as** R
inner join reserve **on**
R.reservation_number = reserve.reservation_number
inner join tool **as** T **on**
T.Tool_id = reserve.tool_ID
where R.Request_customer = $customer_username;

1.5 Check Tool Availability



//Application provide $tool_type, $start_date, $end_date
//Check tool reservation relation, sell relation, and repair relation

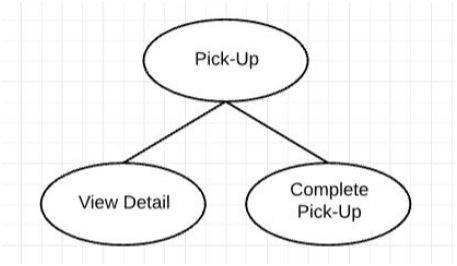
SELECT Tool_ID, Abbrdesc, Deposit, Rental
FROM TOOL
WHERE is_Sold = 'N' **and**
tool_type = $Tool_Type **and**
tool_ID **not in**
(**select** tool.tool_id **from** tool, reserve, reservation
where ((reservation.start_date< $start_date **and** reservation.end_date>$start_date) **or**
(reservation.start_date< $end_date **and** reservation.end_date>$end_date)) **and**
reserve.reservation_number = reservation.reservation_number and tool.tool_id =
reserve.tool_id) **and**
tool_id **not in**
(**select** tool_id **from** service_order
where

(start_date< $start_date **and** end_date>$start_date) **or** (start_date<$end_date **and** end_date>$end_date));

1.6 View Tool Details

```
//Application provide $tool_ID
//View tool details
SELECT T.Tool_ID, T.AbbrDesc, T.FullDesc, T.Purchase_Price, T.Deposit, T.Rental, A.ACCESSORIES
From TOOL AS T
LEFT JOIN ACCESSORIES AS A
ON T.Tool_ID = ACCESSORIES.Tool_ID
WHERE Tool_ID = $tool_ID;
```

1.7 Pick Up Reservation



//ViewDetail is the same as 1.6

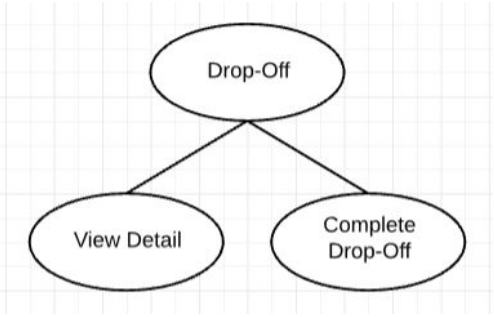
```
//Show pickup interface by $Reservation_number
        SELECT T.tool_ID, T.abbrdesc
        from TOOL as T
        inner join reserve on
        T.Tool_id = reserve.tool_ID
        inner join RESERVATION as R on
        R.reservation_number = reserve.reservation_number
        where R.reservation_number = $reservation_number;

        SELECT sum(T.rental*(DATEDIFF(R.end_date,R.start_date))), sum(T.deposit)
        from RESERVATION as R
        inner join RESERVE on
        R.reservation_number = reserve.reservation_number
        inner join tool as T on
        T.Tool_id = reserve.tool_ID
        where R.reservation_number = $reservation_number;
```

//Application provide $reservation_number, $credit_card_number, $Expiration_date, $clerk_name
//Complete PickUP
UPDATE RESERVATION
SET Pickup_creditcard_number = $credit_card_number,
Pickup_creditcard_expdate = $Expiration_date,
Pickup_clerk = $clerk_name
WHERE Reservation_Number = $Reservation_Number

//generate rental contract based on $reservation_number
SELECT R.Reservation_number, R.Pickup_clerk, R.Request_customer,
R.Pickup_creditcard_number, R.start_date, R.End_date,
T.tool_ID, T.abbrdesc
from Reservation as R
inner join reserve on
R.reservation_number = reserve.reservation_number
inner join tool as T on
T.Tool_ID = reserve.Tool_ID
where R.reservation_number = $reservation_number;

SELECT sum(T.rental*DATEDIFF(R.end_date,R.start_date)), sum(T.deposit)
from RESERVATION as R
inner join RESERVE on
R.reservation_number = reserve.reservation_number
inner join TOOL as T on
T.Tool_ID = reserve.Tool_ID
where R.reservation_number = $reservation_number;

1.8 Drop Off Reservation

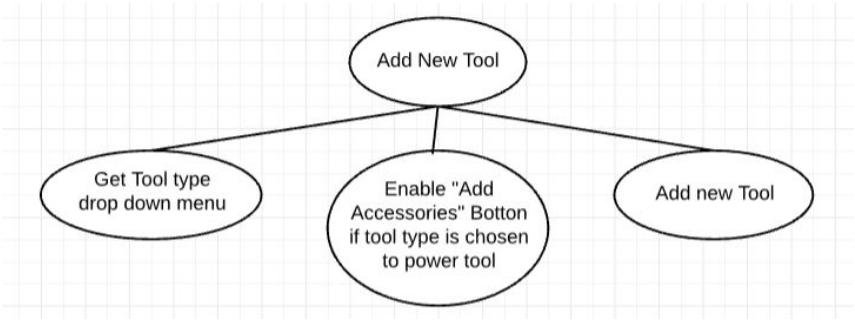


//ViewDetail is the same as 1.6

//Application provide $reservation_number, $clerk_name
//Complete Dropoff
UPDATE RESERVATION
SET Dropoff_clerk = $clerk_name
WHERE Reservation_Number = $Reservation_Number

//Generate receipt based on $Reservation_Number
SELECT R.Reservation_number, R.Pickup_clerk, R.Request_customer,
R.Pickup_creditcard_number, R.start_date, R.End_date,
sum(T.rental*DATEDIFF(R.end_date,R.start_date)), -sum(T.deposit),
sum(T.rental*DATEDIFF(R.end_date,R.start_date))
from RESERVATION as R
inner join RESERVE on
R.reservation_number = reserve.reservation_number
inner join TOOL as T on
T.Tool_ID = reserve.Tool_ID
where R.reservation_number = $Reservation_Number;

1.9 Add New Tool



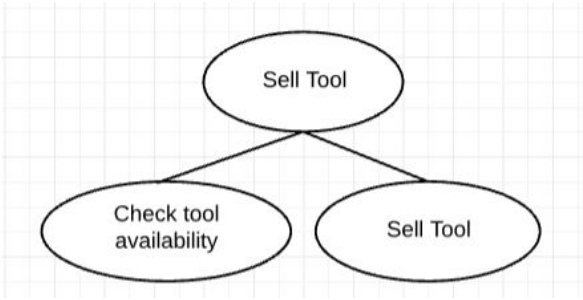
//populate *Tool Type* dropdown
//read $Abbrdesc, $Purchase_price, $Rental, $Deposit, $Fulldesc, $Type
//assume $Username of current clerk is managed by application
//the application generates $ID if the user clicks *Submit New Tool*

INSERT INTO TOOL (Tool_ID, Abbrdesc, Fulldesc, Tool_Type, Is_sold, Purchase_price, Deposit, Rental, Add_Clerk)
VALUES ($ID, $Abbrdesc, $Fulldesc, $Type, 'N', $Purchase_price, $Deposit, $Rental, $Username);

//if Tool Type was "power tool" and the user clicks *Add Accessories* button
//read $Accessory

for each $Accessory
INSERT INTO ACCESSORIES (Tool_ID, Accessories)
VALUES ($ID, $Accessory);
end for

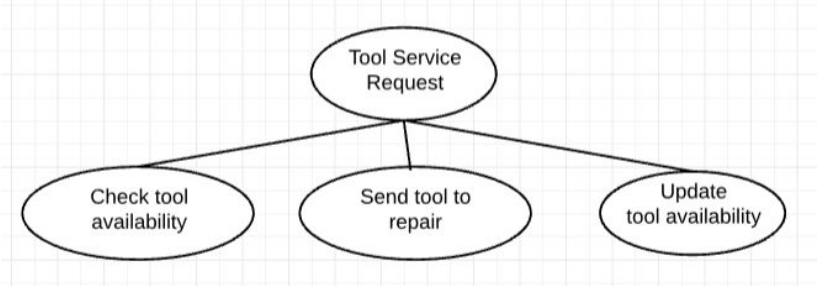
1.10 Sell Tool



//read $Tool_ID
//check if the tool is already sold
SELECT Tool_ID
FROM TOOL
WHERE is_Sold = 'N' **AND**
Tool_ID = $Tool_ID **AND**
tool_ID **NOT IN**
(SELECT tool.tool_id **FROM** tool, reserve, reservation
WHERE reservation.start_date< CURDATE() and reservation.end_date>CURDATE() and reserve.reservation_number = reservation.reservation_number and tool.tool_id = reserve.tool_id) **AND**
tool_id **NOT IN**
(**SELECT** tool_id from service_order
WHERE start_date< CURDATE() and end_date>CURDATE());

// if tool is available for sell
UPDATE TOOL
SET is_sold = 'Y'
Where Tool_ID = $ID;

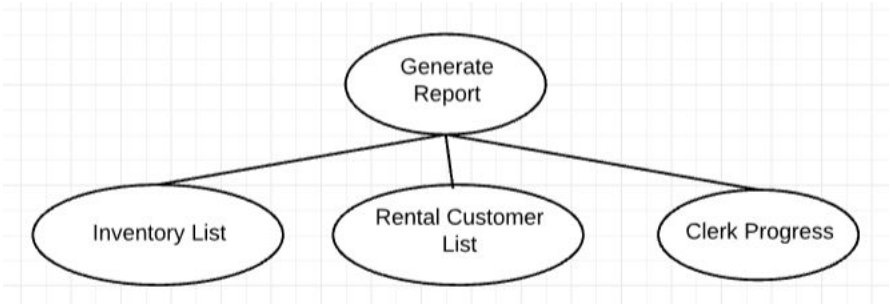
1.11 Request Service Order



//Check tool availability
SELECT Tool_ID
FROM TOOL
WHERE is_Sold = 'N' **and**
Tool_ID = $Tool_ID **and**
tool_ID **not in**
(**select** tool.tool_id **from** tool, reserve, reservation
where ((reservation.start_date< $start_date **and** reservation.end_date>$start_date) **or**
(reservation.start_date< $end_date **and** reservation.end_date>$end_date)) **and**
reserve.reservation_number = reservation.reservation_number **and** tool.tool_id =
reserve.tool_id) **and**
tool_id **not in**
(**select** tool_id **from** service_order
where
(start_date< $start_date **and** end_date>$start_date) **or** (start_date<$end_date **and**
end_date>$end_date));

//Application provide $tool_ID, $start_date, $end_date, $cost
//update service request
INSERT INTO Service_order(Tool_ID, Start_date, end_date, repair_cost)
VALUES($tool_ID, $start_date, $end_date, $cost)

1.12 Generate Reports




```
//Application provide $Month, $Year
//Generate inventory list
SELECT T.Tool_ID, T.Abbrdesc, sum(DATEDIFF(day, R1.Start_date, R1.End_date) * T.Rental) AS
profit,
sum(S.Repair_cost) + T.Purchase_price  AS cost, profit - cost AS total_profit
FROM RESERVATION AS R1 INNER JOIN RESERVE AS R2 ON R1.Reservation_Number =
R2.Reservation_Number INNER JOIN TOOL AS T ON T.ID = R2.Tool_ID INNER JOIN Service_Order
AS S
WHERE month(R1.Start_date) = $Month AND YEAR(R1.Start_date) = $Year AND
Month(S.Start_date) = $Month AND YEAR(S.Start_date) = $Year
GROUP BY R1.Tool_ID
ORDER BY total_profit DESC;

//Application provide $Month, $Year
//Generate rental customer list, assume, for example, if the start_date is in Feb, then the rental
belong to Feb.
SELECT R1.Request_customer, C.Fname, C.Lname, COUNT(R2.Tool_ID) AS num_tools
FROM RESERVATION AS R1 INNER JOIN CUSTOMER AS C ON R1.Request_customer =
C.Username
INNER JOIN RESERVE AS R2 ON R1.Reservation_number = R2.Reservation_number
WHERE month(R1.Start_date) = $Month AND YEAR(R1.Start_date) = $Year
GROUP BY R1.Request_customer
ORDER BY num_tools, C.Lname;


//Application provide $Month, $Year
//Generate Clerk progress
SELECT num_pickup, num_dropoff, num_pickup + num_dropoff AS num_total, clerk_name
FROM (
    SELECT COUNT(Pickup_Clerk) AS num_pickup, Pickup_Clerk AS clerk_name
    FROM RESERVATION
    WHERE Month(R.Start_date) = $Month AND Year(R.Start_date) = $Year
    UNION ALL
```

```
    SELECT COUNT(Dropoff_Clerk) AS num_dropoff, Dropoff_Clerk AS clerk_name
    FROM RESERVATION
    WHERE Month(R.Start_date) = $Month AND Year(R.Start_date) = $Year
)
GROUP BY clerk_name
ORDER BY num_total DESC;
```