

Table of Contents

[Data Types](#)

[Business Logic Constraints](#)

[Task Decomposition with Abstract Code](#)

Customers

[Login](#)

[Customer Main Menu](#)

[Registration](#)

[View Profile](#)

[Check Tool Availability](#)

[Make Reservation](#)

Clerks

[Clerk Main Menu](#)

[Clerk Login](#)

[Pickup Reservation](#)

[Drop-off Reservation](#)

[Add Tool](#)

[Repair Tool](#)

[View/Filter/Override Service Status](#)

[Sell Tool](#)

[View Sale Status](#)

[Generate Reports](#)

[Generate Clerk Report](#)

[Generate Customer Report](#)

[Generate Tool Report](#)

Data Types

User

Attribute	Data Type	Allow Null
Username	String	Not NULL
Email	String	Not NULL
First name	String	Not NULL
Middle name	String	NULL
Last name	String	Not NULL
Password	String	Not NULL

Customer

Attribute	Data Type	Allow Null
State	String	Not NULL
Street	String	Not NULL
City	String	Not NULL
Zip-code	String <digits only>	Not NULL
Zip-extension	String <digits only>	NULL
Home phone area code	String <digits only>	NULL
Home phone number	String <digits only>	NULL
Home phone extension	String <digits only>	NULL
Cell phone area code	String <digits only>	NULL
Cell phone number	String <digits only>	NULL
Cell phone extension	String <digits only>	NULL
Work phone area code	String <digits only>	NULL

Phase 1 Report | CS 6400 - Fall 2017 | Team 055

Work phone number	String <digits only>	NULL
Work phone extension	String <digits only>	NULL
Primary Phone	List <strings>	Not NULL
CVC 3-digit number	String <digits only>	Not NULL
Card number	String <digits only>	Not NULL
Name on card	String	Not NULL
Credit card expiration month	String <digits only>	Not NULL
Credit card expiration year	String <digits only>	Not NULL

Clerk

Attribute	Data Type	Allow Null
Employee number	Integer	Not NULL
Hire date	Date	Not NULL

Reservation

Attribute	Data Type	Allow Null
Start date	Date	Not NULL
End date	Date	Not NULL

Service Order

Attribute	Data Type	Allow Null
Service start date	Date	Not NULL
Service end date	Date	Not NULL
Repair cost	Number	Not NULL

All Tools

Attribute	Data Type	Allow Null
Status	String	Not NULL
Tool number	Integer	Not NULL
Original price	Number	Not NULL
Manufacturer	String	Not NULL
Width	Number	Not NULL
Weight	Number	Not NULL
Material	String	NULL
Length	Number	Not NULL
Sub-option	String	Not NULL
Sub-type	String	Not NULL
For sale date	Date	NULL
Power source	String	Not NULL
Sold Date	Date	NULL

Ladder

Attribute	Data Type	Allow Null
Step-count	Integer	NULL
Weight-capacity	Number	NULL
Step Ladder		
Pail-shelf	Boolean	NULL
Straight Ladder		
Rubber-feet	Boolean	NULL

Hand Tool

Attribute	Data Type	Allow Null
Screwdriver		
Screw-size	Integer	Not NULL
Socket		
Deep-socket	Boolean	NULL
Drive-size	Number	Not NULL
Sae-size	Number	Not Null
Plier		
Adjustable	Boolean	Not NULL
Gun		
Capacity	Integer	Not NULL
Gauge-rating	Integer	NULL
Hammer		
Anti-vibration	Boolean	Not NULL
Rachet		
Driver-Size	Integer	Not NULL

Power Tool

Attribute	Data Type	Allow Null
Max-rpm-rating	Number	NULL
Min-rpm-rating	Number	Not NULL
Amp-rating	Number	Not NULL
Volt-rating	Number	Not NULL

[Table of Contents](#)

Sander		
Dust-bag	Boolean	Not NULL
Drill		
Min-torque-rating	Number	Not NULL
Max-torque-rating	Number	NULL
Adjustable-clutch	Boolean	Not NULL
Saw		
Blade-size	Number	Not NULL
Air-compressor		
Tank size	Number	Not NULL
Pressure-rating	Number	NULL
Mixer		
Motor-rating	Number	Not NULL
Drum-size	Number	Not NULL
Generator		
Power-rating	Number	Not NULL

Accessory

Attribute	Data Type	Allow Null
Accessory-description	String	Not NULL

Cordless

Attribute	Data Type	Allow Null
Battery-type	List <strings>	Not NULL

Garden tool

[Table of Contents](#)

Phase 1 Report | CS 6400 - Fall 2017 | Team 055

Attribute	Data Type	Allow Null
Handle material	String	Not NULL
Pruner		
Blade-length	Number	Not NULL
Blade-material	String	NULL
Digger		
Blade-length	Number	Not NULL
Blade-width	Number	NULL
Striking		
Head-weight	Number	Not NULL
Rakes		
Tine-count	Integer	Not NULL
Wheelbarrow		
Wheel-count	Integer	Not NULL
Bin-material	String	Not NULL
Bin-volume	Number	NULL

Business Logic Constraints

Users:

1. To Login, users must choose Customer or Clerk to start.
2. All users will access the system through the same entry point on the Login page.
3. If no Customer exists in the system with that username, then a new registration interface will pop up prompting the user to create a new profile by registering prior to accessing the system. If the Customer tries to log in as a Clerk, an error message should be displayed.

Customers:

1. Phone number consisted of 3-digit area code and phone number with optional in HomePhone, CellPhone and Work Phone extensions.
2. The middle name is optional.
3. Every Customer is required to have exactly one primary phone number (may be either home, work, or cell).
4. Any Customer may optionally have more than one phone number: home, work, or cell.
5. Customers are required to select the primary phone prior to finishing the registration process.
6. One email address is only allowed to register for one account.
7. EachA customer can only make Reservation can include for no more than 10 tools.
8. Area code, and Phone Number and Phone Extension of telephone are separated by hyphen '-', in HomePhone, CellPhone and WorkPhone.
9. 5-digit zip-code and 4-digit zip-extension are separated by hyphen '-'.

Reservations:

1. Customer can use a new credit -card during the pick-up process, and the clerk is prompted to enter the customer's new credit card information which overwrites the existing credit card information.
2. The rental price and deposit price are derived from Reservation list.
3. All tools are only available for daily rental with 24 hour increment: 12:00:00 AM-11:59:59 PM12:00-11:59 pm.
4. Daily rental prices are 15% of the original purchase price roundup to nearest cents.
5. Deposit prices are 40% of the original purchase price rounded *up* to nearest cent.
6. Sale prices are 50% of the original purchase price rounded *up* to nearest cent.
7. Purchase price is *required* when the Clerk uses the Add-Tool interface.
8. A single Reservation can contain a maximum of ten tools.
9. Actions of add to/remove from cart is implemented thru a pop up interface.

Clerk:

1. The username, email and temporary password of clerks are determined by system administrator. The temporary password will be prompted to be changed at first-time login, and then normal password attribute is declared for Clerk entity.
2. The employer number and hire date collectively define the Clerk.

3. All email addresses for Clerks contain the domain name: '@tools4rent.com'

Tools:

1. Both short description and full description should be aggregated from tools' attributes.
2. Short description is derived from sub-option and sub-type of tools. For power tools only, the short description should also contain power-source.
3. Full-description is derived from tool attributes. The string should represent all applicable attributes for each specialized sub-type when present.
4. Full description should show dimensions first and manufacturer last.
5. Rent times are derived from number of "drop-off"s.
6. When rent times is over 50, change tool status to "for sale".

Task Decomposition with Abstract Code

Customers

Login

Task Decomp



Lock Type: Read-only on [Customer/Clerk](#)

Number of Locks: Single

Enabling Conditions: None

Frequency: Around 200 logins per day

Consistency (ACID): Not critical. Order is not critical

Subtasks: Mother Task is not needed. No decomposition needed.

Abstract Code

- User need to select one role from Customer or Clerk and the default one is Customer
- When User enters *username* and *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:
 - If Customer username and password record are not found:
 - Go to **Customer Registration** form.
 - Else if Customer username record is found but the password is incorrect:
 - 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:
 - If Clerk record is not found
 - Go back to **Login form**, with error message "There is no clerk with this username"
 - Else if the password is incorrect:
 - 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

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**.

Registration

Task Decomp



Lock Type: Write lock-Insertion on [Customer](#).

Number of Locks: Single.

Enabling Conditions: None.

Frequency: Around 66 registrations per day.

Consistency (ACID): Not critical.

Subtasks: Mother Task is not needed. No decomposition needed.

Abstract Code

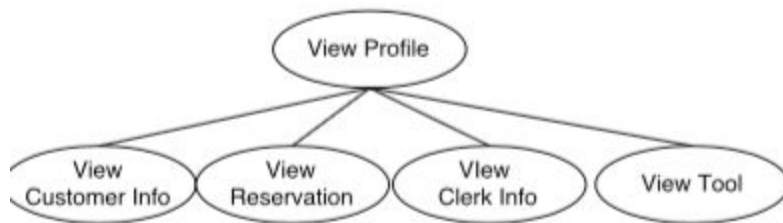
- When User enters *Username, First Name, Middle Name, Last Name, Home Phone, Work Phone, Cell Phone, Email Address, PassWord, Re-type Password, Street Address, Zip Code, Name on Credit card, Credit Number, CVC, validate data*.

[Table of Contents](#)

- User must select one *Primary Phone* from Home Phone, Work Phone and Cell Phone and the value should not be null for which phone type is selected; user select one *State* from all states; use select one *Expiration Month* from 12 months; user select one *Expiration Year*.
- If data validation is successful for all input field and all business logic is satisfied, then:
 - When **Register** button is clicked:
 - If the User username already exists: go back to **Reservation form**, with error message.
 - Else: Insert Customer. Go to **Customer Main Menu** page.
- Else: some inputs are invalid, **Register** button is disabled and display the **Reservation form** with error message.

View Profile

Task Decomp



Lock Type: 4 Read-only lookups on **Reservations**, **Tools**, **Customer** and **Clerk** information.

Number of Locks: Several different schema constructs are needed.

Enabling Conditions: All 4 are enabled by Customer Login.

Frequency: Low- All 4 have the same frequency.

Consistency (ACID): Not critical.

Subtasks: All tasks must be done, but can be done in parallel. Mother task is required to coordinate subtasks, order is not necessary.

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
 - **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

- **View Reservation** using Relationship “**Make**” between Customer and reservation to find Reservations made by the Customer 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

View Tools using relationship “**Add**” between tools and reservation to find Tools rented in this Reservation and display the tool name

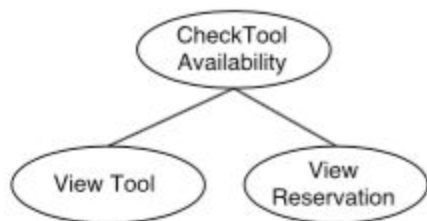
View Clerk using relationship “**Pick up**” and “**Drop off**” between clerk and reservation to find the clerk that picks up this Reservation and display clerk name (if not picked up yet, display not picked up yet)

Find the clerk that drops off this Reservation and display clerk name (if not dropped off yet, display not dropped off yet)

When ready, customer select next action from choices in **Customer Main Menu**.

Check Tool Availability

Task Decomp



Lock Types: Read-only 2 lookup of tool information and Reservation.

Number of Locks: Several different schema constructs are needed.

Enabling Conditions: Both are enabled by Customer Login.

Frequency: High frequency.

Consistency (ACID): Not critical.

Subtasks: All tasks must be done. Mother task is required to coordinate subtasks. **View tool** information subtask come first to find out tools whose current status is available, and then run **view Reservation**.

Abstract Code

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

If Customer does not select one type from (All tools, Hand Tool, Garden Tool, Ladder, Power Tool):

[Table of Contents](#)

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

While the **Search** button is not clicked, do nothing.

While the **Search** button is clicked, then:

- **View Tool**

- Find tools matching the entered type, power source, sub-type, keyword and whose status is not for-sale and sold. Tools that have been reserved during the selected dates are excluded in the results in case of overlapping.

- **If** the number of available tools is greater than 10:

Display the error message: "Specify a more unique search"

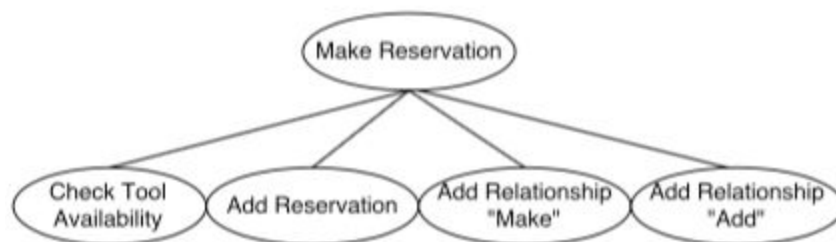
Else:

Display the tool number, Short Description, Rental Price and Deposit Price of each available tool

- When the short description of each available tool is clicked, display more details about the tool

Make Reservation

Task Decomp



Lock Type: One (read-only) lookup of available tools using **Check Tool Availability** task.

One (write-only) insertion of new Reservation.

Two (write-only) add make relationship type between customer and Reservation, add add relationship type between tools and Reservation.

Number of Locks: Several different schema constructs are needed.

Enabling Conditions: Both two are enabled by Customer Login.

Frequency: 200 per day.

[Table of Contents](#)

Consistency (ACID): Not critical.

Subtasks: All tasks must be done. Mother task is required to coordinate subtasks. **Check Tool Availability** need to run first to find available tools, and then run **Add Reservation**, add relationship **“Make”**, add relationship **“Add”** in one transaction.

Abstract Code

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

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

While the **Search** button is not clicked, do nothing.

While the **Search** button is clicked, then:

- Use **Check Tool Availability** task to display each available tool’s Tool ID, Description, Rental price, Deposit price
- When the **Add** checkbox of any available tool is clicked:
Display that tool to the temporary list in **Tools added to Reservation page**, and hide that tool in the original page
- When the **Remove** checkbox of any tool in the temporary list for Reservation is clicked:
show that tool in the available tools, and remove that tool in the temporary list in **Tools add to Reservation page**
- When the **Calculate Total** button is clicked,
 - **If** there is no tool in **Tools added to Reservation page**:
display the **Make Reservation form** with error message “please add tools”
 - **Else if** the number of tool in **Tools added to Reservation page** is greater than 10:
display the **Make Reservation form** with error message “reduce the number of tool in the current Reservation to 10”
 - **Else:**
pop up **Reservation Summary Form** with all tool information in **Tools added to Reservation page**
When **Submit** button is clicked:

- If the current status of tools in the Reservation is for-sale, sold or in-repair:
Display Make Reservation form with error message
- Else if any tool in the current Reservation has already been reserved during the specified time period:
Display Make Reservation form with error message
- Else:
Add Reservation, add relationship “Make” between customer and Reservation, **add relationship “Add”** between tools and Reservation, display the Reservation Confirmation Page

When **Reset** button is clicked:

Remove the Reservation in the memory, Display Make Reservation form.

Purchase Tool

Task Decomp



Lock Type: One (read-only) lookup of Tool information.

One (write-only) update tool information.

One (write-only) add buy relationship between customer and tool.

Number of Locks: Several different schemas construct are needed.

Enabling Conditions: All 3 are enabled by Customer Login.

Frequency: High - same frequency.

Consistency (ACID): Critical.

Subtasks: All tasks must be done. Mother task is required to coordinate subtasks. **View tool** information subtask come first to find out tools whose current status is for-sale, and then run **update tool** and **add relationship “Buy”** in one transaction.

Abstract Code

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:

Phase 1 Report | CS 6400 - Fall 2017 | Team 055

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

When the **Search** button is not clicked, do nothing.

When the **Search** button is clicked, then run **View Tool**:

- **View tool** matching the inputted keyword, tool type, power source, subtype and whose status is For-sale:
Display Tool number, current Status, Short Description, purchase price of each found tool.

When the description of each available tool is clicked:

- display more details about the tool.

When the **Purchase Tool** button of any available tool is clicked:

- add the tool to the temporary list for purchasing
- hide that tool in the available tools to purchase

When the **Remove** checkbox of any tool in the temporary list for purchasing is clicked:

- remove this tool from temporary list 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 temporary list for purchasing to **Purchase Summary Form** and display it with *current dates*, *total purchase price* and *information* about tools in the temporary list.

When **Submit** button is clicked:

- **If** the status of tools in the purchase is sold:
Display **Purchase Tool form** with error message
- **Else:**
update tool status to sold and **add buy relationship** between customer and tools, display the **Purchase Confirmation Page**

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

Task Decomp



Lock Types: Two (read-only) lookup of Reservation waiting for pick up and tool information.
One (write-only) update for tool information.
One (write-only) add another pick-up relationship between tool and clerk.
One (read-only) lookup of or one (write-only) update for customer information (credit card).

Number of Locks : Several different schema constructs are needed.

Enabling Conditions: All five are enabled by Clerk Login.

Frequency: High frequency - same.

[Table of Contents](#)

Consistency(ACID): Critical.

Subtasks: All tasks must be done. Mother task is required to coordinate subtasks. **View Reservation** need to run first to find tools waiting to be picked up, and then run **update tool, update/read customer information, add relationship “Pick-up”** in one transaction, finally run **view tool**.

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, Customer ID, start date and end date.
 - When clerk clicks a link, more detail about Reservation is displayed.
 - 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 all data in **Pickup Reservation page** to **Pick-up Reservation Confirmation page** and display this page.
- **If** Existing Credit Card is selected:
Read Customer. Populate the credit card information.
Else:
Display **Pickup Reservation Confirmation** form with room to add new Customer credit card information

When **Confirm Pick-up** button is clicked,

If New Credit Card is selected:

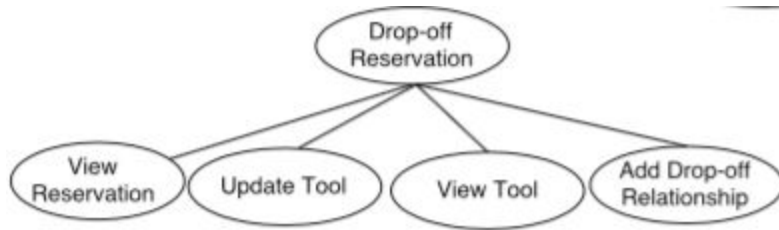
Update Customer information for Credit Card information

Find Reservation with Reservation id, **Add pick-up relationship** between clerk and Reservation, **Update Tool** to change tool status, and display Rental contract after pick-up page

when **print contract** button is clicked, this page is printed out
when Tool name is clicked, run **view tool** task

Drop-off Reservation

Task Decomp



Lock Types: Two (read-only) lookup of Reservation waiting for drop and tool information.

One (write-only) update for tool information.

One (write-only) add another drop-off relationship between tool and clerk.

Number of Locks: Different schema constructs are needed.

Enabling Conditions: This four are enabled by Clerk Login.

Frequency: Same - high frequency.

Consistency (ACID): Critical.

Subtasks: All tasks must be done. Mother task is required to coordinate subtasks. **view**

Reservation need to run first to find tools waiting to be dropped off, and then run **update tool**, **add relationship "drop-off"** in one transaction, finally run **view tool**.

Abstract Code

- Clerk clicked on **Drop-off Reservation** button from **Clerk Main Menu**:
- Run the **View Reservation** task: query for information about the Reservations
 - Find all Reservations have been picked up and have not been dropped off, and display Reservation ID, Customer, Pick-up clerk, deposit price and rental price.
 - 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 all data in **Drop-off Reservation** to **Drop-off Reservation Confirmation** form and Display:
 - When Tool name is clicked, run **view tool** task
 - When **Drop-off** button is clicked,
 - Find Reservation with Reservation id, **Update tool** to change tool status to available and **add drop-off relationship** between tool and clerk, pop final receipt

Add Tool

Task Decomposition



Lock Type: One (write-only) insert a new tool.

Schema Constructs: Single.

Enabling Conditions: Enabled by Clerk Login.

Frequency: 10x/week

Consistency(ACID): Not critical. **Order is** not critical.

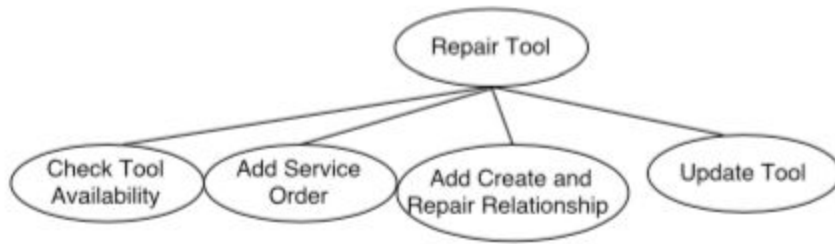
Subtasks: Mother task is not needed. Not necessary to be decomposed into subtasks.

Abstract Code

- 1) Clerk need to select tool type first, other input fields are disabled before selecting a tool type
 - **If** Power tool type is selected:
 - I. Display the power source drop-down menu, and clerk must select one of them before proceed to enter other input fields.
 - **If** Cordless is selected:
Display Battery type and D/C Volt Rating input fields
 - II. Display Power Tool Accessory input fields
- 2) Subtype dropdown menu are populated according to the selected tool type and power source (if power tool selected), and clerk must select one of them before proceed to enter other input fields.
- 3) Sub-option dropdown menu is populated and other tool detail information input fields are generated according to the tool type, power source and subtype selected before.
- 4) **If** required input fields are entered and data validation is successful:
Submit Button are now enable to click
Else:
Submit Button are disabled to click
- 5) When **Submit** Button is clicked, insert the new tool into database, and display **Clerk Main Menu**

Repair Tool

Task Decomposition



Lock Type: One (read-only) **Check Tool Availability (previous mother task)**.

Three (write-only) add new service order for a tool, add create relationship type between clerk and service order and add repair relationship type between service order and tools.

One (write-only) update tool information

Schema Constructs: Different schema constructs are needed.

Enabling Conditions: All 5 are enabled by Clerk Login.

Frequency: Low - same frequency.

Consistency(ACID): Critical. Insert new service order for a tool and update tool information must be done in a transaction.

Subtasks: All tasks must be done. Mother task is required to coordinate subtasks. **Check Tool Availability** need to run first to find tools available for repairing, and then run **add service order**, **add relationship "Create" and "Repair"**, **update tool** in one transaction.

Abstract Code

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

When **Search** button is Clicked do the following:

Check Tool Availability:

1. Find tools matching clerk inputs (start date, end date, customer search, type, power source, subtype) and tool's status is available.
2. For each available tool, display Tool id, Description, Rental Price, Deposit Price.
3. If Description of a tool is clicked, display detail information for a tool.

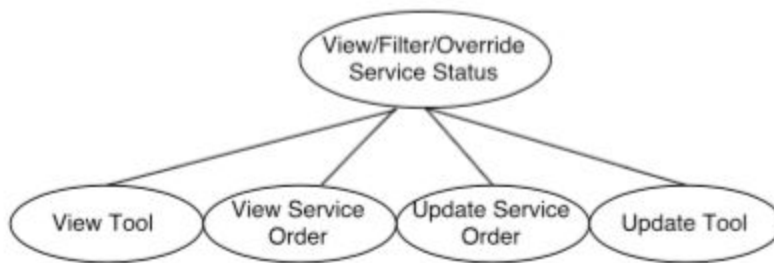
When **Confirm** button is Clicked do the following:

[Table of Contents](#)

- If data validation is successful:
 - If tool's status 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
Update Tool: change tool's status from available to in-repair
Display Repair Tool form.
- Else:
Display Repair Tool form with error message

View/Filter/Override Service Status

Task Decomposition



Lock Type: Two (read only) lookup service order and tools in repair.

Two (write only) update tool information and update service order information.

Number of Locks: Different schema constructs are needed.

Enabling Conditions: All 4 are enabled by Clerk Login.

Frequency: Low-same frequency.

Consistency (ACID): Critical. Update service order for a tool and update tool information must be done in a transaction.

Subtasks: All tasks must be done. Mother task is required to coordinate subtasks. **view tool** and **view service order** need to run first to find tools in repair and its corresponding service order, and then run **update tool**, **update service** in one transaction.

Abstract Code

When **Search** button is Clicked do the following:

1. **View Tool** to find tools matching clerk inputs (Type and Keyword) and tool's status is in-repair.

2. For each found tool, find its corresponding service order by comparing now() date with start and end date(**View service order**), and display service ID, Current Status, Tool ID, Description, Start Date, End Date, Repair Cost, Clerk.
3. If Description of a tool is clicked, display detail information for a tool.

When **Fix Now** button is Clicked do the following:

- If tool's status is not in-repair (in case other clerks click **Fix-Now** for this tool before you click **Fix-Now** button):

Display Repair Tool form with error message

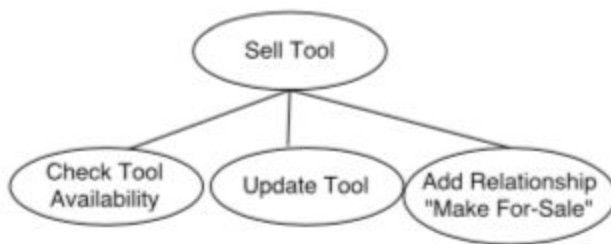
Else:

Update tool: change tool status from in-repair to available

Update service order: change service order End date to now, if start date is later than now, update the start date to now too.

Sell Tool

Task Decomposition



Lock: One (read-only) **check Tool Availability (previous mother task)**.

One (write-only) update tool information.

One (write-only) add relationship "Add For-sale".

Schema Constructs: Different schema constructs are needed.

Enabling Conditions: All of three are enabled by Clerk Login.

Frequency: 10x/week

Consistency(ACID): Critical.

Subtasks: All tasks must be done. Mother task is required to coordinate subtasks. **Check Tool Availability** to find available tools, and then run **update tool**, **add relationship "Add For-sale"** in one transaction.

Abstract Code

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

[Table of Contents](#)

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

When **Search** button is Clicked, do the following:

1. Run **Check Tool Availability** to find available tools
2. For each available tool, display Tool id, Description, Rental Price, Deposit Price
3. If Description of a tool is clicked, display detail information for a tool.

When **Sell Tool** button is Clicked do the following:

- **If** tool's status is not available (in case other clerks click **Sell Tool** for this tool before you click **Sell Tool** button):

Display Sell Tool form with error message

Else:

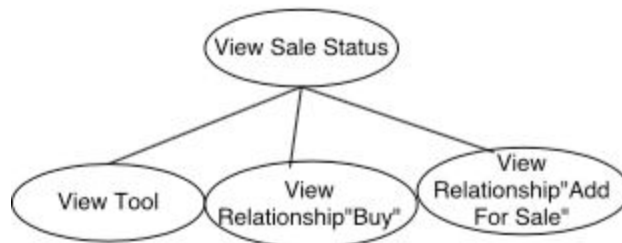
Update tool: change tool status from available to for-sale.

Change tool's for-sale date to now.

Add relationship "Add For-sale"

View Sale Status

Task Decomposition



Lock Type: Three (read-only). Lookup tools whose status is for-sale or sold; lookup of relationship **"Buy"**; lookup of relationship **"Add for sale"**.

Schema Constructs: Different schema constructs are needed.

Enabling Conditions: Enabled by Clerk Login.

Frequency: Low-same frequency.

Consistency (ACID): Not critical.

Subtasks: Mother task is required to coordinate subtasks. First, run **View tools**, and then **View relationship "Buy"** and relationship **"Add for sale"**

Abstract Code

When **Search** button is Clicked do the following:

View Tools in Sold or For-Sale:

1. **View tools** to find tools matching clerk inputs (Type and Keyword) and tool's status is sold or for-sale
2. For each found tool, current status, tool id, description, **View relationship "Buy"** to find customer (if sold), sale price, sale date (if sold) and **View relationship "Add for sale"** to find Clerk ID.
3. If Description of a tool is clicked, display detail information for a tool.

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

Task Decomposition



Lock Type:

- 1) Look up Clerk information.
- 2) Look up Reservations to count number of pickups or drop-off for each clerk.

Number of Locks: Different schema constructs are needed.

Enabling Conditions: Enabled by Clerk Login.

[Table of Contents](#)

Frequency: Low - same frequency.

Consistency (ACID): Not critical. Order is not critical

Subtasks: Mother task is required to coordinate subtasks. First, run **View Clerk**, and then **View Reservation**

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, and count how many Reservation is picked up/dropped off by that clerk
- And add Reservation picked up and dropped off by this clerk together and display the sum

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

Task Decomposition



Lock Types:

- 1) Look-up Customer information.
- 2) Look-up Reservations to count Reservation made by this customer.

Number of Locks: Different schema constructs are needed.

Enabling Conditions: Enabled by Clerk Login.

Frequency: Low - same frequency.

Consistency (ACID): Not critical.

Subtasks: Mother task is required to coordinate subtasks. First, run **View Customer Profile**, and then **View Reservation**.

Abstract Code

[Table of Contents](#)

Run **View Customer** task:

- For each customer, display Customer ID, First Name, Middle Name, Last Name, Email, Primary phone
- **If View Profile** button is clicked, display customer detail

Run **View Reservation** task:

- Group Reservation by customer attribute, and count how many Reservation and how many tools are rented by that customer and display them

When **Back to Report Menu button** is clicked:

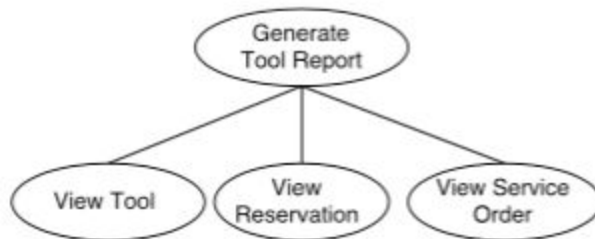
Display **Select a Report** page

When **Reload Results** is clicked:

Reload **Customer Report** page

Generate Tool Report

Task Decomposition



Lock Types:

- 1) Look-up Tool information.
- 2) Look-up Reservations.
- 3) Lock-up service order.

Number of Locks: Different schema constructs are needed.

Enabling Conditions: Enabled by Clerk Login.

Frequency: Low - same frequency.

Consistency (ACID): Not critical.

Subtasks: Mother task is required to coordinate subtasks. First, run **View Tool**, and then **View Reservation** and **View service order**.

Abstract Code

When **Search** Button is clicked:

Run **View Tool** task:

- 1) Find Tools matching the clerk's input (Type, keyword).
- 2) Display Tool Id, Current Status, Description, rental profit, sale date.
- 3) If Description of a tool is clicked, display detail information of the tool.

[Table of Contents](#)

Run **View Reservation, and service order** task:

For each tool,

- 1) According to tool's status, find date in Reservation/service order and display.
- 2) Find service order with tool id, and calculate its repair cost. Display the sum of repair cost and original price.
- 3) Use rental profit to reduce total cost to get total profit.
 if the current **status** is **sold**:
 total profit += sold price of tool
Then display total profit.