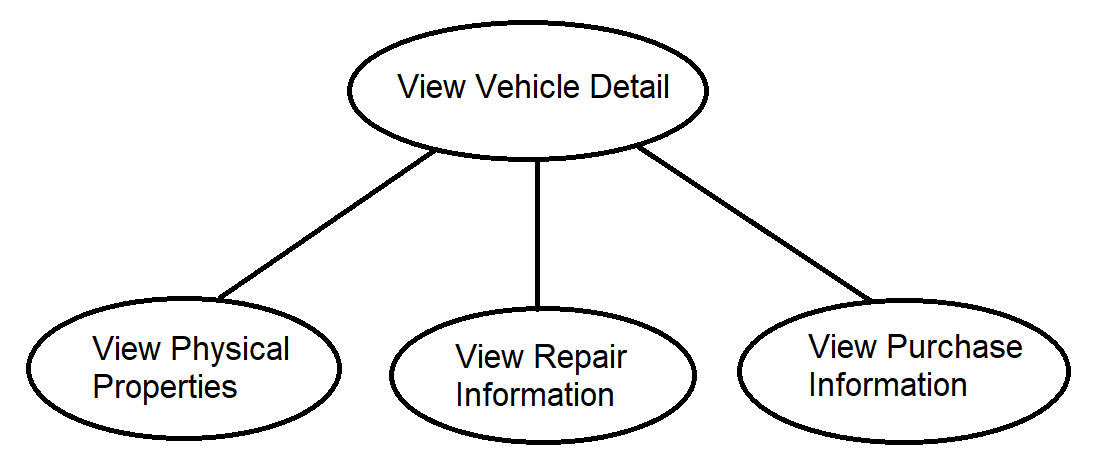
1 Vehicle detail form

Task Decomposition



Subtasks: View physical properties; view repair; view transactions

Physical properties attributes: VIN, Vehicle type, Model Year, Manufacturer, Model, Colors, Mileage, Sales Price, Description

Repair: Condition, Repair status (no repair, pending, in progress, complete), Repair description, Repair vendor(name, address, phone), Start date, End date, Repair cost, and the Recall number and manufacture (if applicable)

Transaction: Purchase price

All subtasks look up;

Enabling conditions: click on vehicle details button after selecting a single car

Frequency: 3 subtasks have different frequency;

Consistency is not essential, order

A mother task is needed

Constraints:

Repair status is either no repair, pending, in progress or complete;

If repair status = complete, no change in repair is allowed;

Repair end date is later than start date;

Repair cost >= 0;

If repair status = ‘no repair’, repair cost = 0 and no change in repair is allowed;

Mileage >= 0;

Year is later than 1900 -1 -1

Sales price = purchase price \* 125% + total repair cost \* 110%;

Datatype: see add vehicle and add repair form;

Abstract Code:

Run the ‘vehicle detail’ task: select a single vehicle in search screen and click the button ‘view vehicle detail’:

View vehicle physical properties;

IF login as company owner: view vehicle repair and transaction properties;

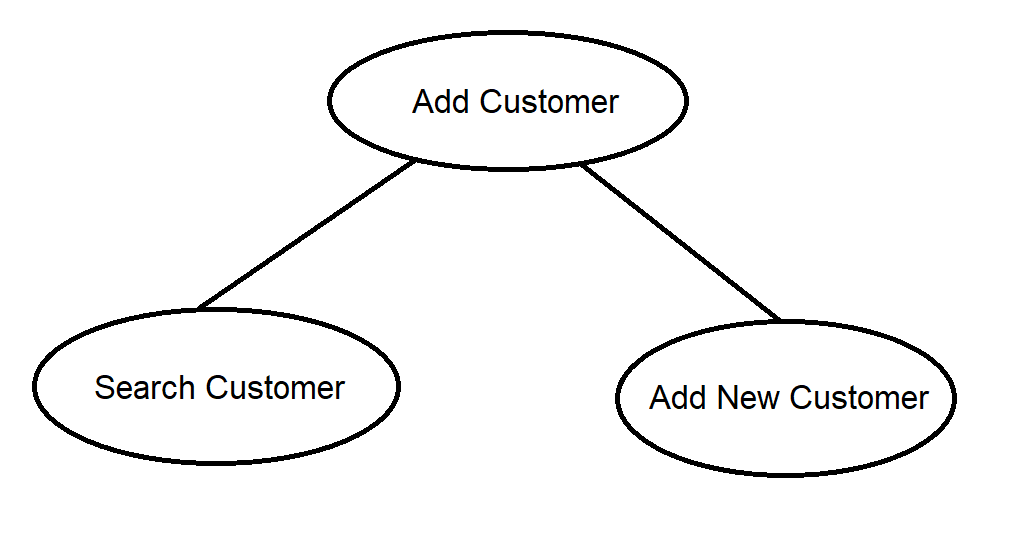
IF login as clerk: view vehicle repair and transaction properties;

Show dropdown menu in ‘repair status’ column; choose can change the repair status;

IF ‘repair status’ = complete: no further change is allowed; dropdown menu disappeared.

IF ‘add repair’: direct to ‘add repair’ task

2 Add customer



Task decomposition

Subtasks are look up or insert;

Enabling conditions: click on ‘add customer’ button after salesperson select a vehicle and click ’sell vehichle’, or after clerk click ‘add vehicle’

Frequency: subtasks have different frequency;

A mother task is needed

Consistency: Search run first, then insert;

Constraints:

A customer is either a person or a business

Datatype:

Address: String, not null;

Email: String, nullable;

Phone: Integer, not null;

Name: String, not null;

DL: String, not null;

TIN: String, not null;

Business\_name: String, not null;

Primary\_contact\_name: String, not null;

Primary\_contact\_title: String, not null;

Abstract Code:

Run ‘add customer’ task: salesperson or clerk click the button ‘add customer’:

Search: enter driver’s license (DL) or tax identification number (TIN):

IF input in current DL or TIN list: view customer profile;

ELSE: dropdown menu to select either person or company;

IF DL is selected: display new customer form for person (address, email, phone\_number, first\_name, last\_name);

IF SAVE: add customer;

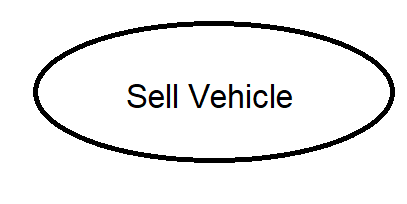
IF CANCEL: display new customer form for person;

IF tax identification number is selected: display new customer form for business (address, email, phone\_number, business\_name, primary\_contact\_name, primary\_contact\_title);

IF SAVE: add customer;

IF CANCEL: display new customer form for company;

3 Sale order form



Task decomposition

Lookup vs (insert, delete or update)?

Lookup and update

How many schema constructs involved?

little

Are enabling conditions consistent across tasks? (some enabled can run first)

consistent

Are frequencies consistent across tasks? (can split high and low frequently running tasks)

Is consistency essential?

Is mother tasks needed to control subtasks?

Datatype:

Sales date: date

Constraints:

Sales date should be later than vehicle purchase date;

If a vehicle sold status is sold: no further modification is allowed.

Abstract Code:

Run ‘sale order form’ when salesperson click ‘sell vehicle’ button in vehicle detail screen after searching by VIN:

IF ‘confirm sale’: pop out window and enter sales date; update vehicle detail;

IF ‘cancel’: view vehicle detail