HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

KHOA CÔNG NGHỆ THÔNG TIN 1

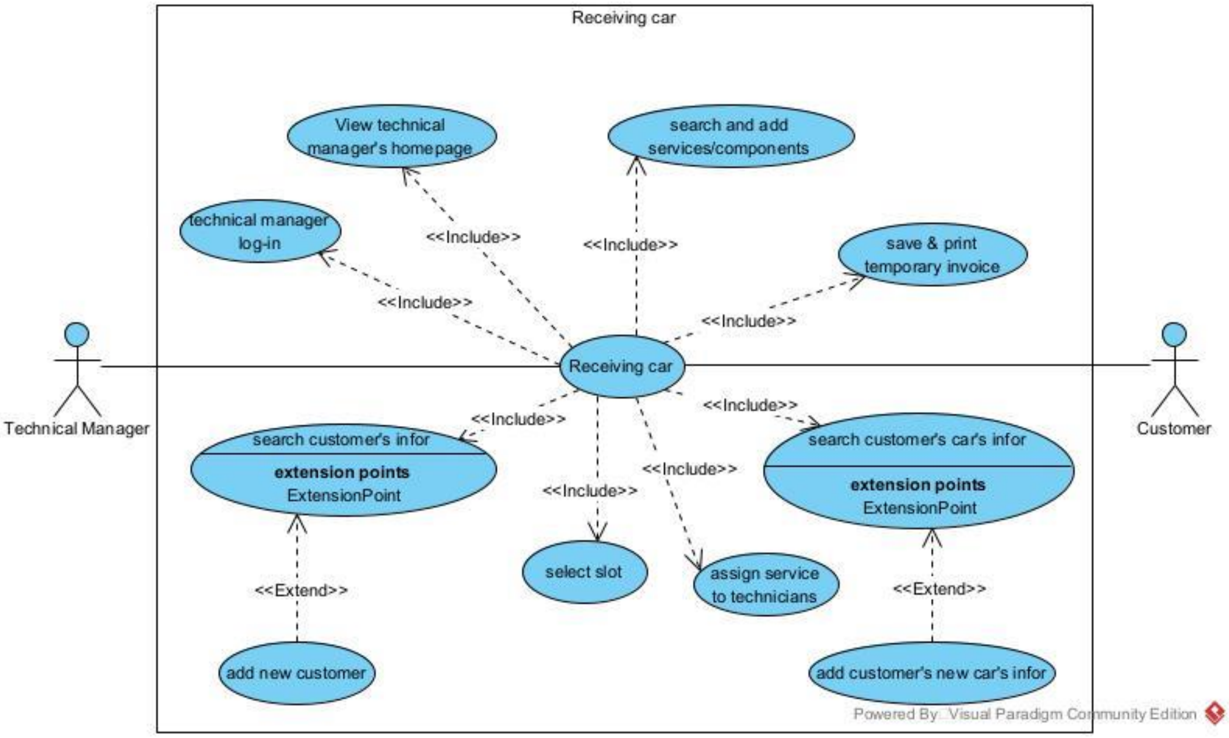
MÔN: NHẬP MÔN CÔNG NGHỆ PHẦN MỀM

Lớp **:** E22CQCN02-B
Nhóm bài tập lớn: 02
Đề tài bài tập lớn: Garage car service management
Danh sách nhóm : 1. Trần Xuân Kiên - B22DCVT269
2. Trịnh Diệu Linh - B22DCVT310
3. Vũ Ngọc Linh - B22DCDT179
4. Trần Nhật Minh - B22DCVT349
Module **:** Receiving car
Yêu cầu **:** Báo cáo tổng hợp

Hà Nội – 2025

A. Requirement

1. Detailed use case



Use case description:

- Technical manager log-in: log-in for technical manager
- View technical manager homepage: this enables the technical manager to view the homepage after log into the system
- Search and add services/components: this enables the technical manager to search the service/component, enter quantity and add to the invoice
- Search customer’s information: this enables the technical manager to search the customer by name
- Add new customer: this enables the technical manager to add a new customer if the customer is new
- Search car’s information: this enables the technical manager to search for the car’s information by plate number
- Add new car: this enables the technical manager to add a new car if the car has not be brought to the garage before
- Assign service to technicians: this enables the technical manager to assign added services to technicians to do
- Select slot: this enables the user to select the slot to put the car in
- Save temporary invoice: this enables the technical manager to save all the invoice information to the system and print out to attach the car

2. Standard scenario

<table>
<tr><th>Scenario</th><th>Receiving car</th></tr>
<tr><td>Actor(s)</td><td>Technical manager, customer</td></tr>
<tr><td>Pre-condition</td><td>Technical manager has an account to access the system</td></tr>
<tr><td>Post-condition</td><td>The car is received and has an attached temporary invoice</td></tr>
<tr><td>Main events</td><td>1. A technical manager Nguyen Van Anh login with username = techmanager, password = 123 into the system to receive a car bring by customer
2. The system displays the technical manager’s homepage which includes an option to select receiving car function
3. The technical manager selects the receiving car function
4. The system shows the search customer UI which includes:
- Input: full name
- Button: search, add new
5. The technical manager asks the customer’s name
6. The customer answers his name is Tran Binh An
7. The technical manager enters customer’s name = Tran Binh An to the search bar and clicks search button
8. A list of all clients whose name contains the keyword “Tran Binh An” is listed as follow:
<table>
<tr><th>ID</th><th>Name</th><th>Phone number</th><th>Address</th><th>Note</th></tr>
<tr><td>1</td><td>Tran Binh An</td><td>0897653458</td><td>Nam Dinh</td><td></td></tr>
<tr><td>2</td><td>Nguyen Tran Binh An</td><td>0112233445</td><td>Ha Dong</td><td></td></tr>
<tr><td>2</td><td>Tran Binh An</td><td>0442233445</td><td>Thanh Oai</td><td></td></tr>
<tr><td>3</td><td>Tran Binh An Nhien</td><td>1223344556</td><td>Hoai Duc</td><td></td></tr>
</table>
9. Technical manager asks customer if the information in the first row is correct
10. Customer confirms
11. Technical manager selects the first row
12. The interface of searching customer’s car’s information appears</td></tr>
</table>

with:

- Input: Plate number
- Button: search, add new

13. The technical manager asks customer about plate number of the car he/she has brought
14. The customer answers that the plate number of the car is 30G-123.12
15. The technical manager inserts the plate number 30G-123.12 into the text field and click search
16. The result with a list of car with plate number contains the keyword = 30G-123.12 appears as follow:

ID	Plate number	Name	Brand	Type
1	30G-123.12	Mazda 6	Mazda	4 seats
2	30G-123.124	BMW SE 2023	BMW	4 seats

17. The technical manager selects the result and clicks next
18. The interface to add services/components to customer's order appears including

- Input: Name of service/component, quantity
- Button: search, add to invoice, next

19. The technical manager tells the customer the component that the car needs to change the oil and asks if the customer agrees to do the service
20. The customer agrees
21. The technical manager inserts the key word "oil" into the text field and click the search button
22. The list of service/component which name includes the key word "tire" appears as follow

ID	Name	Price	Description	Estimated time (min)
1	Oil change	100 000đ	Change engine oil	60
2	Oil filter replacement	199 000đ	Replace oil filter	60

23. The technical manager selects the first row and inserts the quantity is 1 then click add to invoice, repeats the process until adding all the required services/components then click next
24. The interface to assign added services to technicians appears as

below:

Select service: [Select ▼] Select date: 2025-05-20

Button: next day, search timeslot

Time slot: [Select ▼]

ID	Name	Select

- Button: next

25. Technical manager clicks the combo box select service and select one service, keep the current date and click search

26. The timeslots that are counted from current time, have free technician and have the duration equals to estimated time of selected service appears in the combo box select timeslot, the list of free technicians in the first timeslot will appear as below

Time slot: [08:00 – 09:00]

ID	Name	Select
1	Nguyen Duc Canh	□
2	Le Phong	□

27. The technical manager selects the first technician to do the selected service and repeats the process from choosing service until assign all the services to technicians, then clicks next

28. The interface to select slot to put the car in appears with a list of slots that are free between the earliest starting time of the added services and the latest ending time of the added services

ID	Slot name
1	A01
2	A02
3	B01
4	B02
5	C01
6	C02

- Button: go back

29. The technical manager select the first slot

30. The confirmation invoice interface appears with detail information of the invoice:

- Invoice code: 12
- Created at: 08:00 20/05/2025
- Customer name: Tran Binh An
- Customer phone number: 0897653458
- Car ‘s name: Mazda 6
- Car’s plate number: 30G-123.12
- Received by: Nguyen Van Anh
- Slot: A01

No	Service / Component	Unit price	Quantity	Item Total	Technician
1	Oil change	100 000đ	1	100 000đ	Nguyen Duc Canh
2	Battery replacement	300 000đ	1	300 000đ	Le Phong, Pham Thi Ca

- Total: 400 000đ

And 2 button: Cancel, Confirm

31. The technical manager reads the total bill to the client and asks to confirm

32. The customer confirms the invoice

33. The technical manager clicks the confirm button

34. The system displays a successful message

35. Technical manager clicks ok and informs the success to the customer

36. The system display the technical manager’s homepage

3. Entity classes of analysis phase

Step 1: Describe the Receiving car module

The system assists the technical manager in receiving cars brought in by customers and creating a temporary invoice linked to each car. The technical manager can search for a customer by name or add a new customer if they are not already in the system. Next, the technical manager can search for the received car by its license plate number. If the car has previously been brought, its information will be available; otherwise, the technical manager can add a new car for the customer. The technical manager can then add the required services or components by searching for their names and selecting the correct items, insert the quantity and click add to invoice then click next. The interface to assign services to technicians appears, the technical manager selects the service, the default date is current and clicks search timeslot, if there are no free technicians can click the next day button to search for free timeslot and technicians in the next day. After assigning all the services the technical manager click next to choose the slot that is free from the starting time of the first service and ending time of the last one. After choosing slot the confirm interface with detailed information of the temporary invoice appears, technical manager can click confirm or cancel.

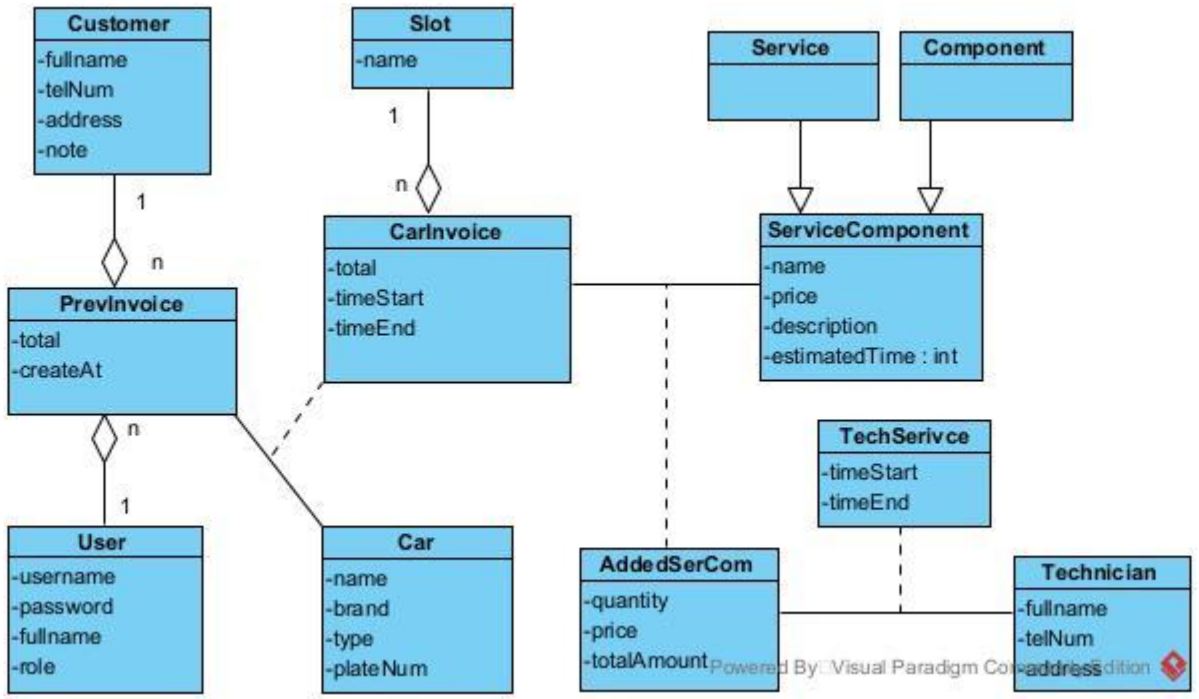
Step 2+3: Noun extraction and evaluate

- System: an abstract noun → reject
- Technical manager → class User: username, password, fullname, role
- Car → class Car: plate number, name, brand, type
- Customer → class Customer: name, phone number, address, note
- Customer's name: attribute → reject
- License plate number: attribute → reject
- Car's information: attribute → reject
- Service → class Service: name, price, description, estimatedTime
- Component → class Component: name, price, description, estimatedTime
- Temporary invoice → class PrevInvoice: createdAt, total
- Technician → class Technician: fullname, telNum, address
- Slot → class Slot: name
- Interface: an abstract noun → reject

Step 4+5: Quantity and relationship among classes

- A user can create many temporary invoices, a temporary invoice is created by one user, so the relationship between User - PrevInvoice is 1-n
- A car can have many temporary invoices for many visits, an invoice can be for more than one car so Car - PrevInvoice is n-n. We create an entity CarInvoice between them
- Service and Component have many attributes in common and each time searching will search for both service and component so we create an entity ServiceComponent. Service and Component will inherit from this entity
- A car at one visit can require many services or repairing parts, a service/component can be done for many cars so CarInvoice - ServiceComponent is n-n. We create a class AddedSerCom between these two entities

- An added service can require many technicians to do, a technician can do many services so Technician-AddedSerCom is n-n. We create a class TechService between these two entities
- A car is put in one slot, a slot can be used for many cars at different times so CarInvoice -Slot is n-1



4. Classes diagram of analysis phase

Analysis receiving car module:

- Enter the system → Log-in interface appears → Need a class LoginView
  - Input for username → inUsername
  - Input for password → inPassword
  - A button to login → subLogin
- Enter the username + password → the system must check if the login is correct → need a method check login:
  - Name: checkLogin()
  - Input: username, password (class User)
  - Output: boolean
  - Owner class: User
- Once the login is successful → the technical manager's homepage appears → need a class TechMngHomeview which has at least:
  - An option to choose Receiving car → subReceiveCar

- Choose the option to receive car → The search client's information UI appears → need a class SearchClientView
    - Input for fullname → inFullname
    - A button to search → subSearch
    - A button to add → subAdd
    - A list of customers whose name includes the inserted name, can click to choose the correct one → outsubListClient
- Enter customer's information to search → The system must search all customers whose name contains the entered fullname → need a method to search client by name
    - Name: searchClient()
    - Input: key words
    - Output: a list of customers
    - Owner class: Customer
- The results are returned to (and displayed on) the SearchClientView.
- The technical manager chooses the correct customer, if the customer is new, click button add, the AddClientView interface appears including :
    - Input for fullname → inFullname
    - Input for phone number → inTelNum
    - Input for address → inAddress
    - Input for not → inNote
    - A button to confirm adding → subAdd
    - A button to cancel → subCancel

  And a method to add new customer:
    - Name: addClient()
    - Input: an object of customer
    - Output: a new customer
    - Owner class: Customer
- The search customer's car's information appears → need a class SearchClientCar
    - Input for plate number → inPlate
    - A button to search → subSearch
    - A button to add → subAdd
    - A list of cars whose plate includes the inserted string, can click to choose the correct one → outsubListCar

- Enter customer’s car’s plate number to search → The system must search all car whose plate contains the entered string → need a method to search car by plate number
    - Name: searchCar()
    - Input: a string of plate number
    - Output: a list of cars
    - Owner class: Car
- The results are returned to (and displayed on) the SearchClientCar.
- The technical manager chooses the correct car, if the car is new, click button add and AddNewCar interface appears, including:
    - Input for plate number → outinPlate
    - Input for car’s name → outinName
    - Input for car’s brand → outinBrand
    - Input for car’s type → outinType
    - A button to add → subAdd
    - A button to cancel → subCancel

  And a method to add new car:
    - Name: addCar()
    - Input: an object of car
    - Output: a new car
    - Owner class: Car
- The search and add service/component appears → need a class SearchSerCom
    - Input for service/component’s name → inSerComName
    - A button to search → subSearch
    - A list of services/components whose name includes the inserted string, can click to choose the correct one → outinResultList
    - Input for quantity → inQuantity
    - A button to add the selected service/component and quantity to invoice → subAddToInvoice
    - A list of added services/components → outAddedList
    - A button to click next after adding all service/component → subNext
- Enter service/component’s name to search → The system has to search all service/component whose name contains the entered words → need a method to search service/component by name
    - Name: searchByName()
    - Input: key words

- Output: a list of service/component
- Owner class: ServiceComponent

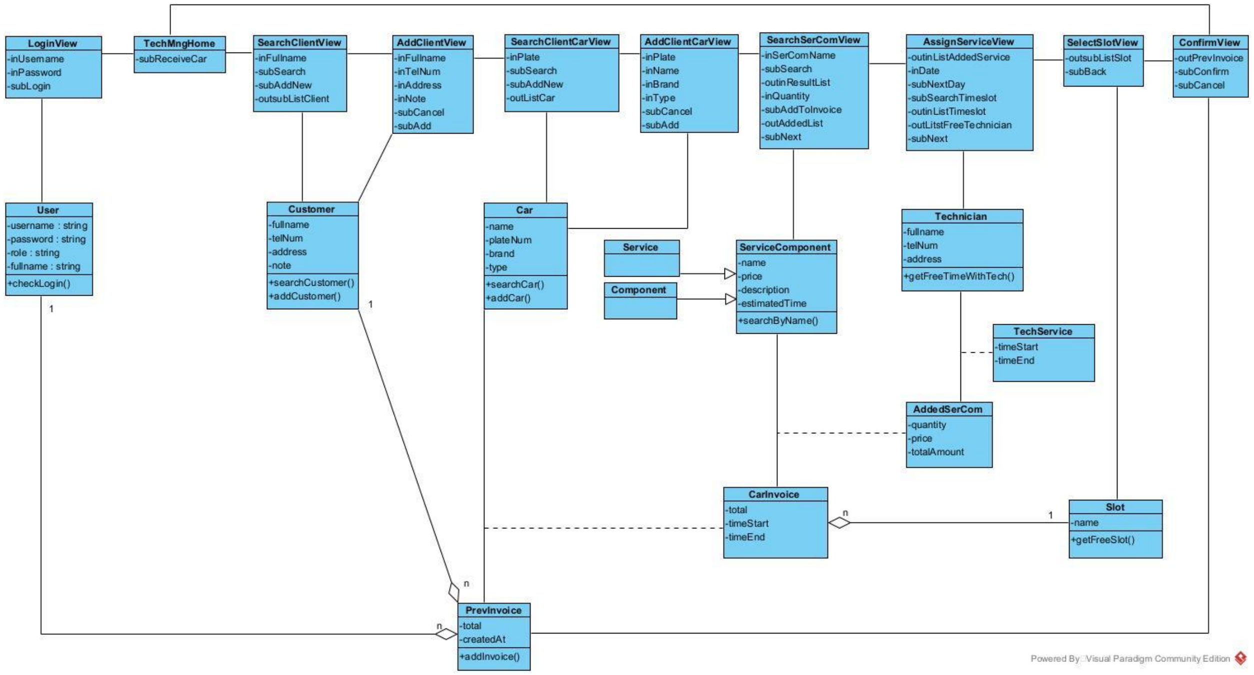
- The results are returned to (and displayed on) the SearchSerCom.
- The interface to assign services to technicians appears → need a class AssignTechnician including:
  - A dropdown menu with added service to choose the service to assign → outinListAddedService
  - Input for date which is searching timeslot in → inDate
  - A button to increase date if there are no free technicians for selected day → subNextDay
  - A button to search for timeslot in selected date and long enough to do the selected service → subSearchTimeSlot
  - A dropdown menu with list of result timeslots and can select the timeslot → outinListTimeslot
  - A list of free technicians in selected timeslot → outinListFreeTechnician
  - A button to click next after assigning all the services to technicians → subNext
- The technical manager clicks the search free timeslot button → need a method to find all the free timeslot in that working day and list of technicians which are free in that timeslot → need a method

  + name: getFreeTimeWithTech()
  + input: a date and estimated time of service
  + output: a list of free timeslot and list of free technicians in a timeslot
  + owner class: Technician
- After assign all the services, technical manager clicks next, the interface to select slot to put the car in appears → SelectSlotView

  + a list of free slot → outsubListSlot
  + a button to back to assign service interface to choose other timeslot to repair if there are no free slot → subBack
- To find free slots from the starting time of the first service and ending time of the last service, need a method to find free slot

  + name: getFreeSlot()
  + input: time start and time end
  + output: a list of free slot
  + owner class: Slot

- After choosing the slot, the confirm interface appears → need a class ConfirmView
    - + display all information about the temporary invoice → outPrevInvoice
    - + a confirm button → subConfirm.
    - + a cancel button → subCancel
- The technical manager chooses to confirm after having the aggregation from the customer → The system has to save the invoice into the database → need a method:
    - + name: addInvoice()
    - + input: an object of PrevInvoice
    - + output: none or boolean
    - + Owner class: PrevInvoice
- After saving to the database, the system returns to the TechMngHome.



5. Sequence diagram of analysis phase

Scenario version 2

1) The customer brings car to fix
2) The technical manager logins from the LoginView
3) The class LoginView calls the class User to process.
4) The class User executes the function checkLogin()
5) The class User returns result to the LoginView
6) The class LoginView calls the class TechMngHome.

7) The TechMngHome shows to the technical manager

8) The technical manager selects the receiving car function from the homepage

9) The TechMngHome calls SearchClientView

10) The SearchClientView shows to the technical manager

11) The technical manager asks the customer’s name

12) Customer answers his/her name

13) Technical Manager enter customer’s name and click search button of SearchClientView

14) The SearchClientView calls Customer class

15) The Customer class executes function searchClient()

16) The Customer class returns results to SearchClientView

17) SearchClientView shows results to technical manager

18) The technical manager selects the right customer on SearchClientView

19) SearchClientView calls SearchClientCar

20) The SearchClientCar shows to the technical manager

21) The technical manager asks the customer’s car’s plate number

22) Customer answers his/her car’s plate number

23) Technical Manager enter customer’s car’s plate number and click search button of SearchClientCar

24) The SearchClientCar calls CustomerCar class

25) The CustomerCar class executes function searchCar()

26) The CustomerCar class returns results to SearchClientCar

27) SearchClientCar shows results to technical manager

28) The technical manager selects the right car

29) SearchClientCar calls SearchSerCom

30) The SearchSerCom shows to the technical manager

31) The technical manager tells the customer about the required service/component

32) Customer agrees

33) Technical Manager enter service/component’s key word and click search button of SearchSerCom

34) The SearchSerCom calls ServiceComponent class

35) The ServiceComponent class executes function searchByName()

36) The ServiceComponent class returns results to SearchSerCom

37) SearchSerCom shows results to technical manager

38) The technical manager clicks on the right service/component, enters quantity and click add to invoice, click next after adding all the required services/components

39) SearchSerCom calls AssignServiceView

40) The AssginServiceView is shown to technical manager

41) Technical manager select one service, keep the current date and clicks search timeslot

42) The AssignServiceView calls Technician class

43) The Technician class executes function() getFreeTimeWithTech()

44) The Technician class returns results to AssignServiceView

45) AssignServiceView shows results to technical manager

46) Technical manager selects technicians to do the service, click next after assigning all the services to technicians

47) AssginServiceView calls the SelectSlotView

48) The SelectSlotView calls class Slot

49) The class Slot executes function getFreeSlot()

50) The class Slot returns results to SelectSlotView

51) The SelectSlotView is shown to technical manager with list of free slot

52) Technical manager select one slot

53) The SelectSlotView calls ConfirmView

54) The class ConfirmView displays all temporary invoice’s information to the technical manager

55) The technical manager repeats these information to the client and requires the client to confirm.

56)The client confirms the invoice’s information.

57)The technical manager clicks the confirm button.

58)The class ConfirmView call the class PrevInvoice to process

59)The class PrevInvoice calls the method addInvoice().

60) The PrevInvoice shows result to ConfirmView

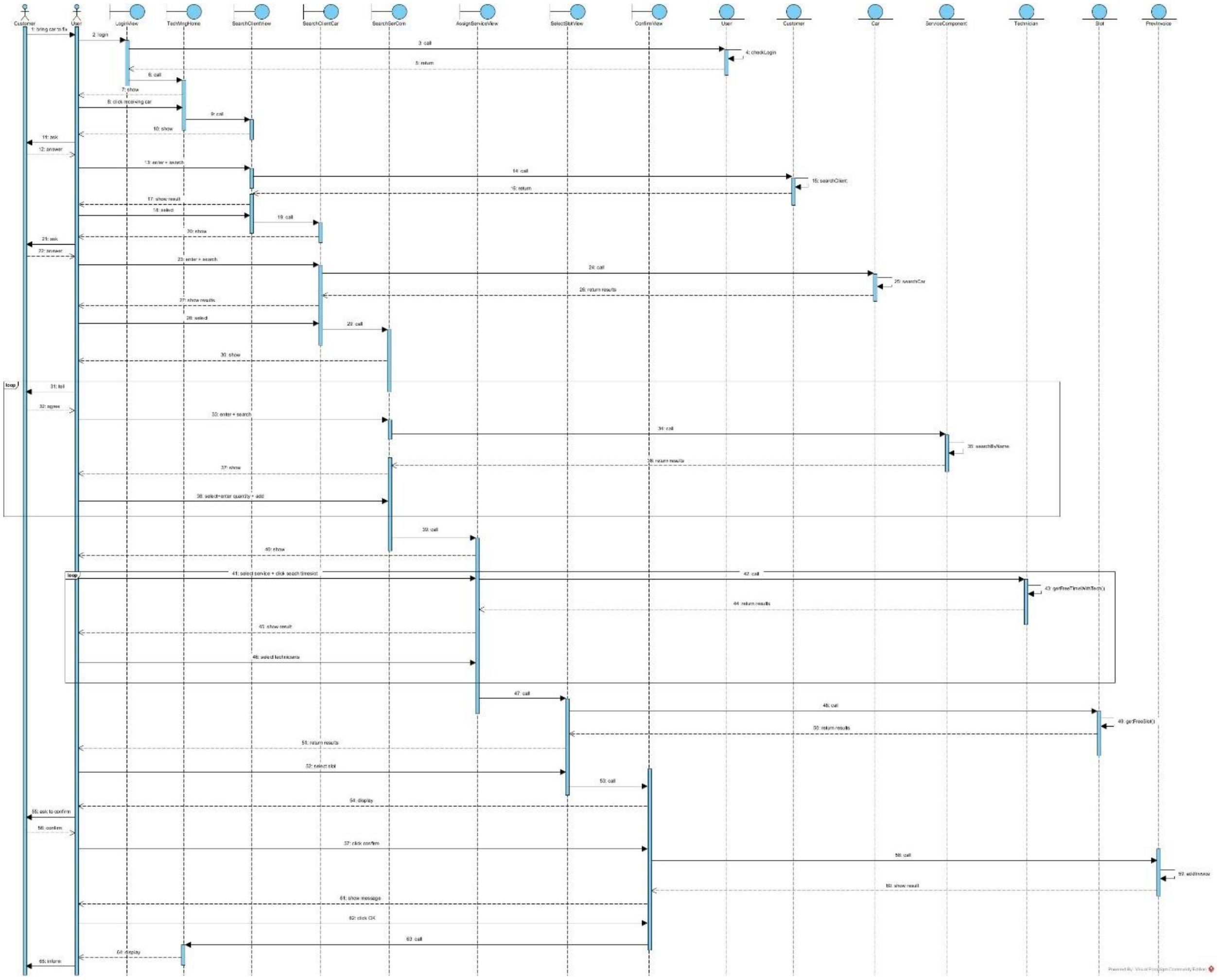
61) The ConfirmView shows successful message to technical manager

62) Technical manager click OK button

63) The ConfirmView calls TechMngHomw

64) TechMngHome displays itself to technical manager

65) The technical manager informs the success to customer



6. Entity classes of design phase

Step 1: Add ID attribute for classes: Customer, Car, User, PrevInvoice, CarInvoice, Slot, SerCom, AddedSerCom, TechService, Technician

Step 2: Add type for attribute for all classes

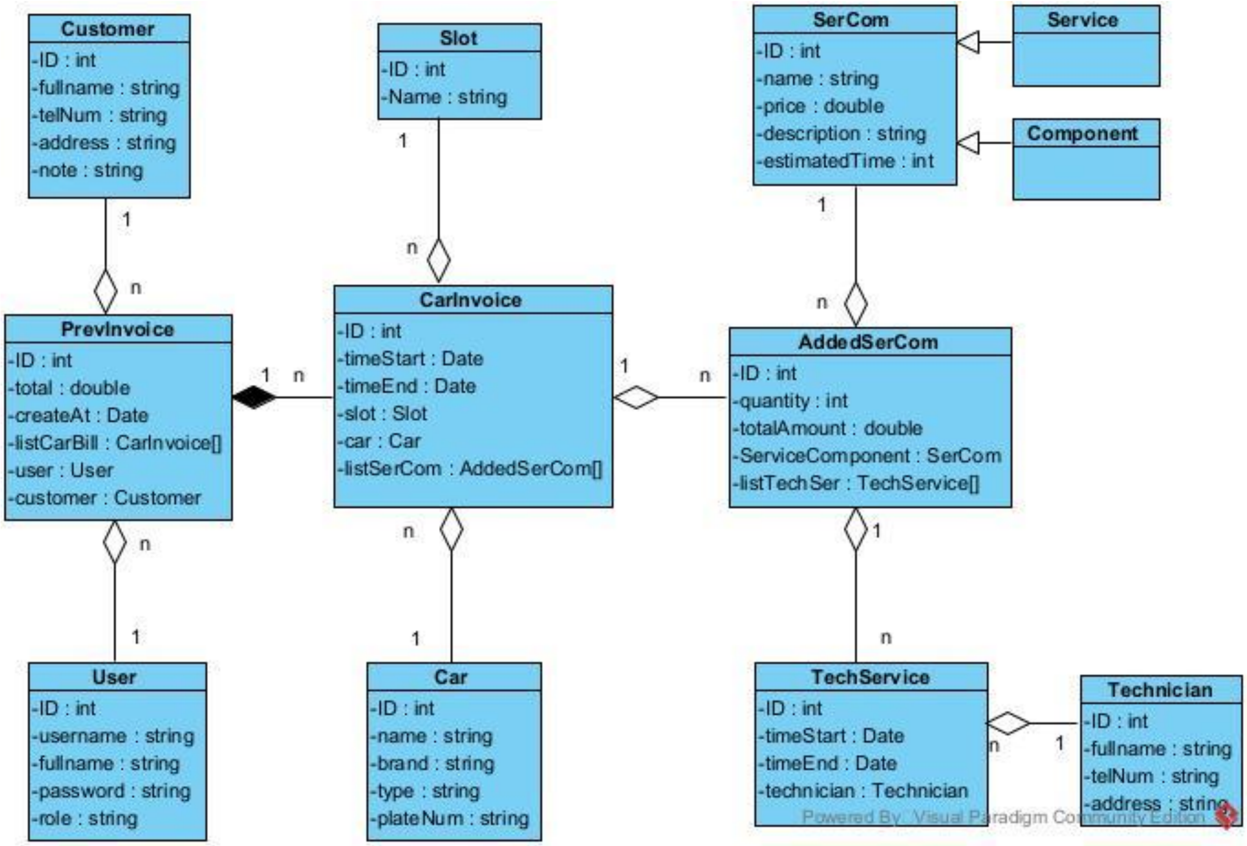
Step 3: Convert association relationships to correspond aggregation/composition relationships:

- Car + PrevInvoice → CarInvoice: Car is a component of CarInvoice, CarInvoice is a component of PrevInvoice
- CarInvoice + ServiceComponent → AddedSerCom: AddedSerCom is a component of CarInvoice, ServiceComponent is a component of AddedSerCom
- AddedSerCom + Technician → TechService: Technician is a component of TechService, TechService is a component of AddedSerCom

Step 4: Add object attributes:

- User is a component of PrevInvoice, of type 1-n → PrevInvoice has a User

- CarInvoice is a component of PrevInvoice, of type n-1 → PrevInvoice has a list of CarInvoice
- Car is a component of CarInvoice, of type 1-n → CarInvoice has a Car
- Customer is a component of PrevInvoice, of type 1-n → PrevInvoice has a Customer
- Slot is a component of CarInvoice, of type 1-n → CarInvoice has a Slot
- AddedSerCom is a component of CarInvoice, of type n-1 → CarInvoice has a list of AddedSerCom
- ServiceComponent is a component of AddedSerCom, of type 1-n → AddedSerCom has a ServiceComponent
- Technician is a component of TechService, of type 1-n → TechService has a Technician
- TechService is a component of AddedSerCom, of type n-1 → AddedSerCom has a list of TechService



7. Database design

Step 1: Create a table for each entity

- Customer → tblCustomer
- PrevInvoice → tbl PrevInvoice

- User → tblUser
- Slot → tblSlot
- CarInvoice → tblCarInvoice
- Car → tblCar
- ServiceComponent → tblSerCom
- AddedSerCom → tblAddedSerCom
- TechService → tblTechService
- Technician → tblTechnician
- Service → tblService
- Component → tblComponent

Step 2: For each attribute of entities which is not an object, convert to attribute of correspond table

- tblCustomer: ID, fullname, telNum, address, note
- tblPrevInvoice: ID, createdAt
- tblUser: ID, username, password, fullname, role
- tblSlot: ID, name
- tblCarInvoice: ID, timeStart, timeEnd
- tblCar: ID, name, brand, type, plateNumber
- tblSerCom: ID, name, price, description, estimatedTime
- tblAddedSerCom: ID, quantity
- tblTechService: ID, timeStart, timeEnd
- tblTechnician: ID, fullname, telNum, address
- tblService: ID
- tblComponent: ID

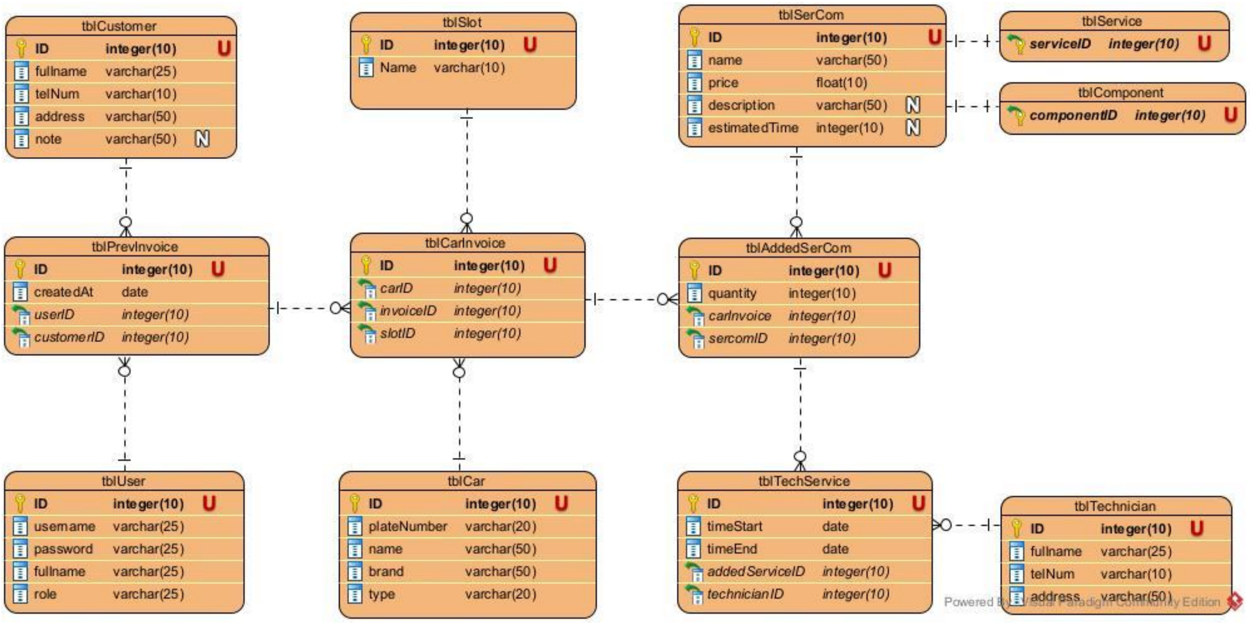
Step 3: Convert the cardinality relationships between entity classes into cardinality relationships between database tables

- 1 tblCustomer – n tblPrevInvoice
- 1 tblUser – n tblPrevInvoice
- 1 tblSlot – n tblCarInvoice
- 1 tblPrevInvoice – n tblCarInvoice
- 1 tblCar – n tblCarInvoice
- 1 tblSerCom – n tblAddedSerCom
- 1 tblAddedSerCom – n tblTechService
- 1 tblTechnician – n tblTechService
- 1 tblSerCom – 1 tblService
- 1 tblSerCom – 1 tblComponent

Step 4: Config the key columns for tables

- For each table has ID attribute → set as primary key
- Foreign key:
  + 1 tblCustomer – n tblPrevInvoice → tblPrevInvoice has foreign key customerID
  + 1 tblUser – n tblPrevInvoice → tblPrevInvoice has foreign key userID
  + 1 tblSlot – n tblCarInvoice → tblCarInvoice has foreign key slotID
  + 1 tblPrevInvoice – n tblCarInvoice → tblCarInvoice has foreign key invoiceID
  + 1 tblCar – n tblCarInvoice → tblCarInvoice has foreign key carID
  + 1 tblSerCom – n tblAddedSerCom → tblAddedSerCom has foreign key sercomID
  + 1 tblAddedSerCom – n tblTechService → tblTechService has foreign key addedServiceID
  + 1 tblTechnician – n tblTechService → tblTechService has foreign key technicianID
  + 1 tblSerCom – 1 tblService → tblService has foreign key serviceID
  + 1 tblSerCom – 1 tblComponent → tblComponent has foreign key componentID

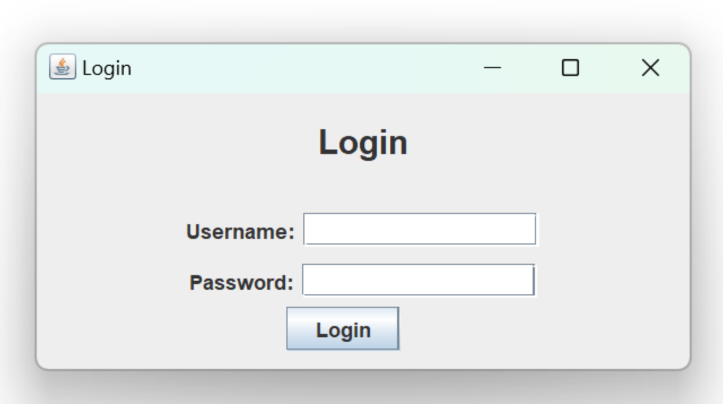
Step 5: Eliminate duplicate or inherit attributes



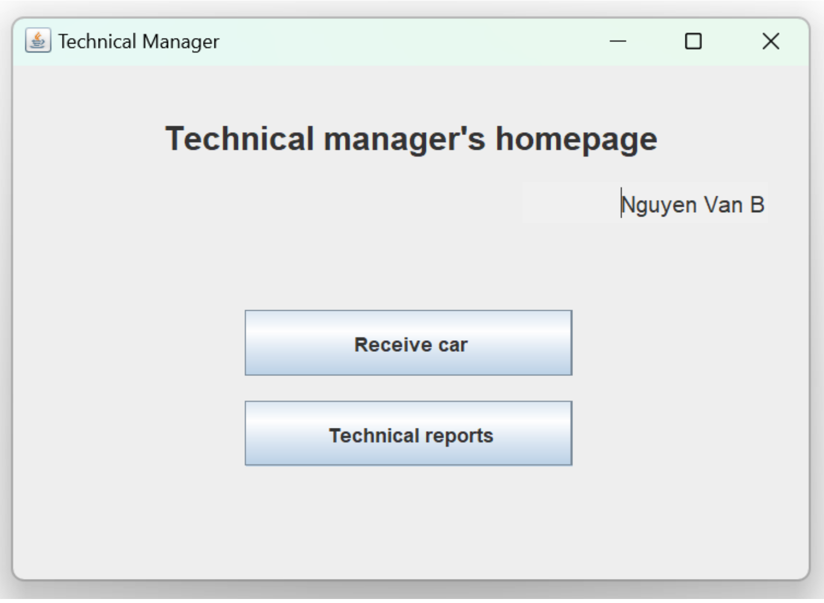
8. UI design and class diagram of design phase

8.1. UI design

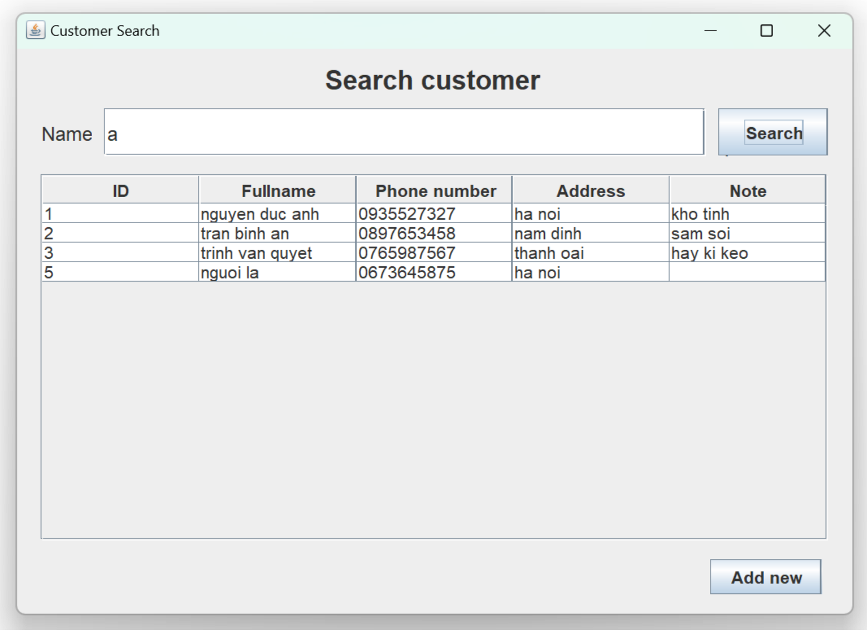
- LoginFrm: have two text fields to insert username and password, one button to login



- TechMngHomeFrm: homepage for technical manager, have two function: receive car and view technical reports, a label to display the user’s fullname

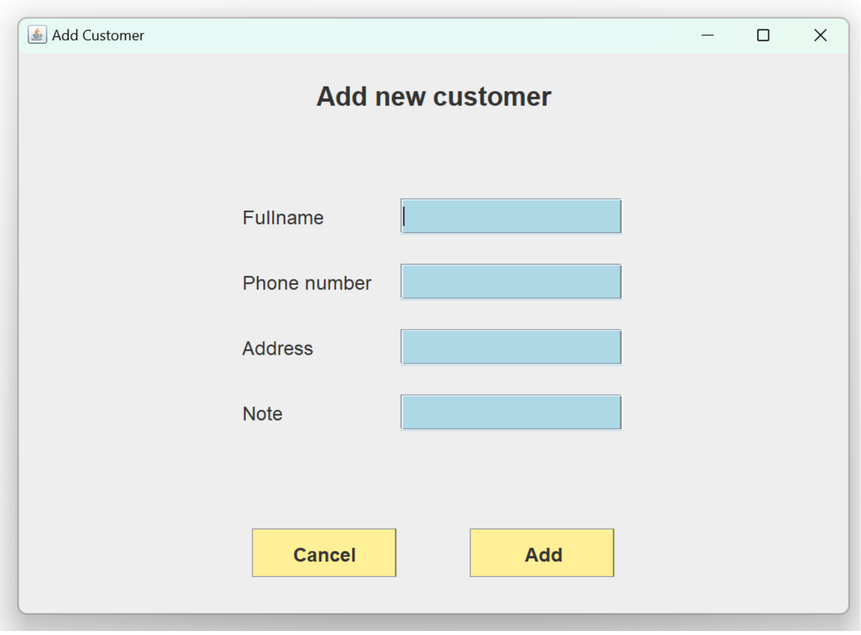


- SearchCustomerFrm: have a text field to insert customer name and button to click search, a table to show searching results and a button to add new customer



ID	Fullname	Phone number	Address	Note
1	nguyen duc anh	0935527327	ha noi	kho tinh
2	tran binh an	0897653458	nam dinh	sam soi
3	trinh van quyet	0765987567	thanh oai	hay ki keo
5	nguoi la	0673645875	ha noi	

- AddCustomerFrm: interface to add a new customer with text fields to insert customer's fullname, phone number, address, note. A button to add and a button to cancel the adding process



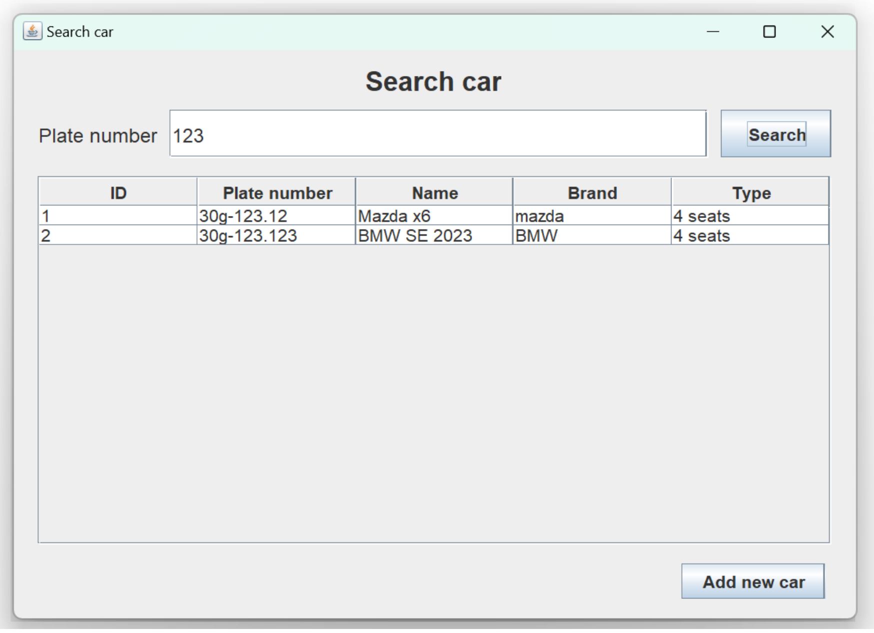
Fullname

Phone number

Address

Note

- SearchCarFrm: have a text field to insert car's plate number and button to click search, a table to show searching results and a button to add new car



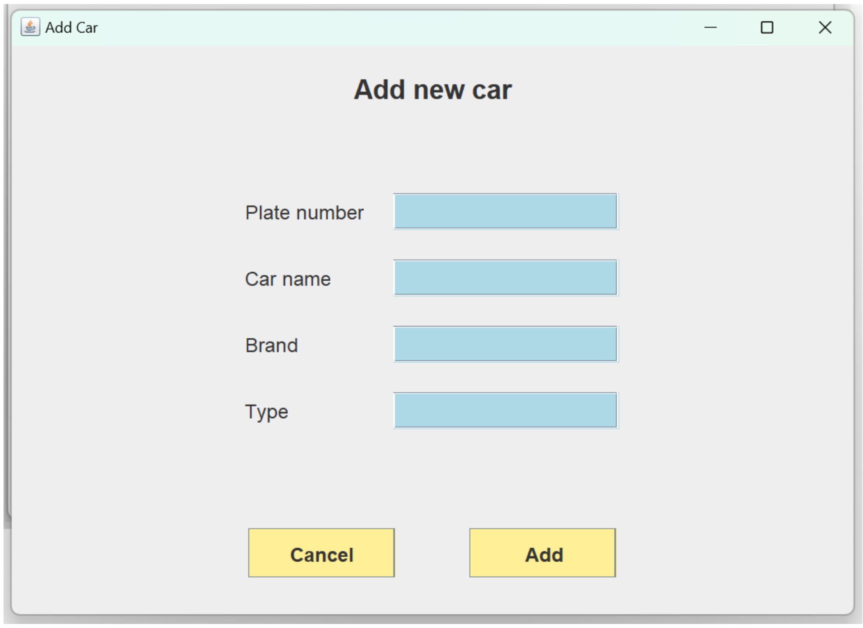
Search car

Search car

Plate number 123

ID	Plate number	Name	Brand	Type
1	30g-123.12	Mazda x6	mazda	4 seats
2	30g-123.123	BMW SE 2023	BMW	4 seats

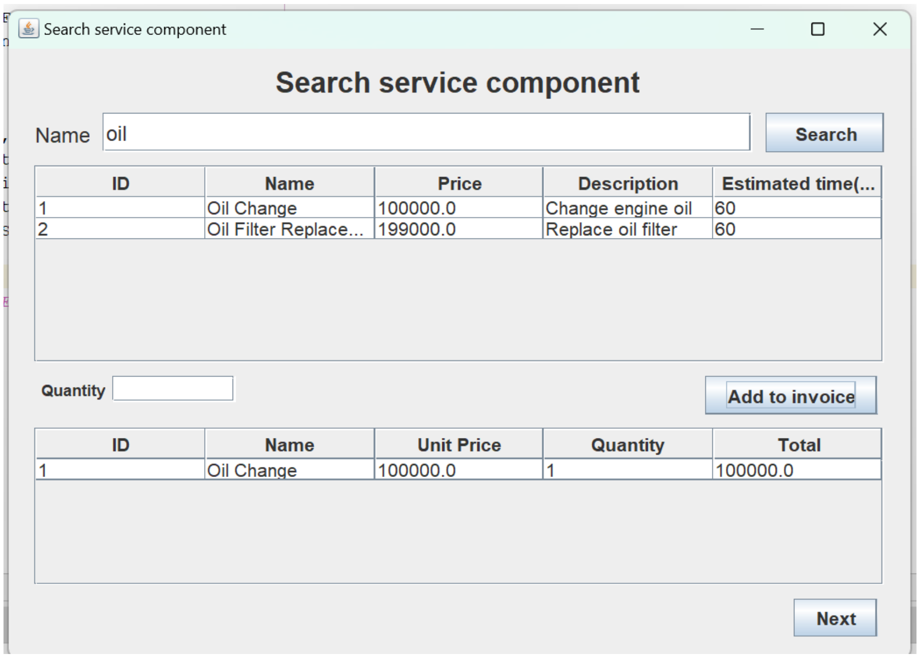
- AddCarFrm: interface to add a new car with text fields to insert car’s plate number, name, brand, type. A button to add and a button to cancel the adding process



Add Car

Add new car

- SearchSerComFrm: interface to search service/component, insert quantity and add to invoice.



Search service component

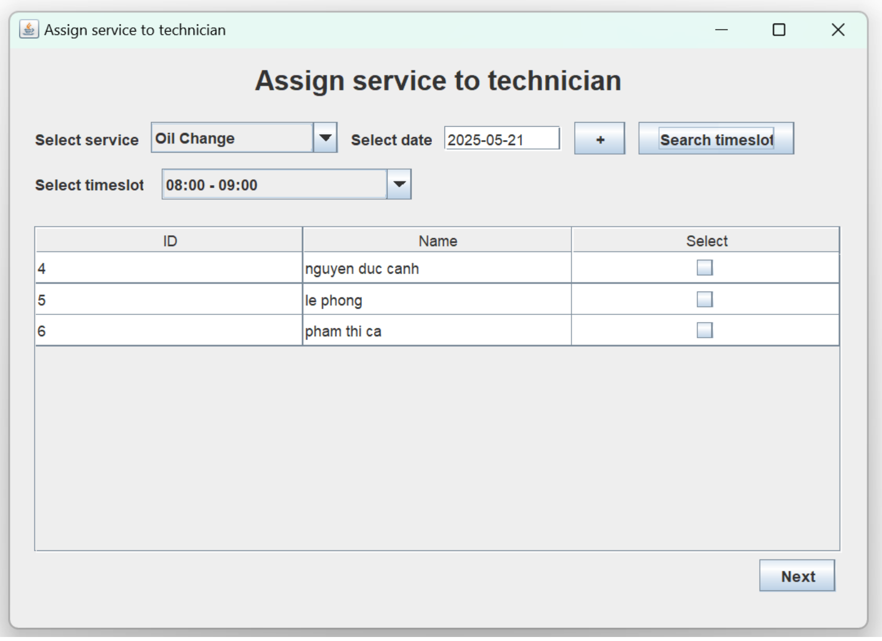
Name: oil

ID	Name	Price	Description	Estimated time(...
1	Oil Change	100000.0	Change engine oil	60
2	Oil Filter Replace...	199000.0	Replace oil filter	60

Quantity:

ID	Name	Unit Price	Quantity	Total
1	Oil Change	100000.0	1	100000.0

- AssignServiceFrm: a combo box to select the service which has been added before, a text field with the default value as the current date, and a button to add days if the current date has no available free timeslot. Search timeslot button to search for timeslot in the day selected which is equal to estimated time of selected service and a table of free technicians in the selected timeslot. A button to click next when assign all the services to technicians



Assign service to technician

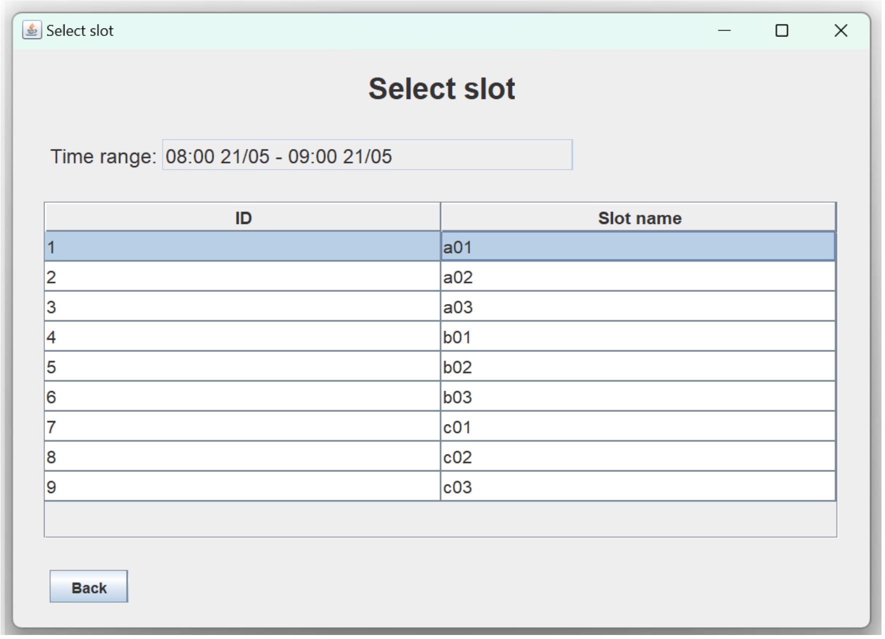
Select service: Oil Change

Select date: 2025-05-21

Select timeslot: 08:00 - 09:00

ID	Name	Select
4	nguyen duc canh	☐
5	le phong	☐
6	pham thi ca	☐

- SelectSlotFrm: a text field to show the start and end time of the car repair, counting from the earliest starting time of a service and latest ending time. A list to show free slot between the time range, can select one row to select slot. A button to go back if there is no free slot



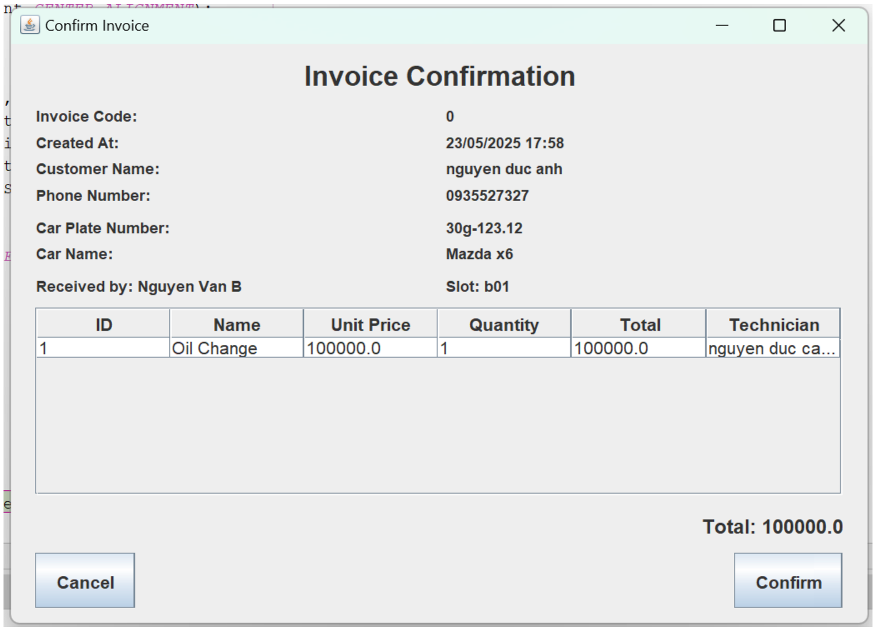
Select slot

Select slot

Time range: 08:00 21/05 - 09:00 21/05

ID	Slot name
1	a01
2	a02
3	a03
4	b01
5	b02
6	b03
7	c01
8	c02
9	c03

- ComfirmFrm: the interface displays the details of temporary invoice, a button to confirm and other to cancel



Confirm Invoice

Invoice Confirmation

Invoice Code:	0
Created At:	23/05/2025 17:58
Customer Name:	nguyen duc anh
Phone Number:	0935527327
Car Plate Number:	30g-123.12
Car Name:	Mazda x6
Received by: Nguyen Van B	Slot: b01

ID	Name	Unit Price	Quantity	Total	Technician
1	Oil Change	100000.0	1	100000.0	nguyen duc ca...

Total: 100000.0

8.2. Class diagram of design phase

View Classes:

- LoginFrm is the interface to login. It needs a text field to enter the username, a text field to enter password, and a button to login.

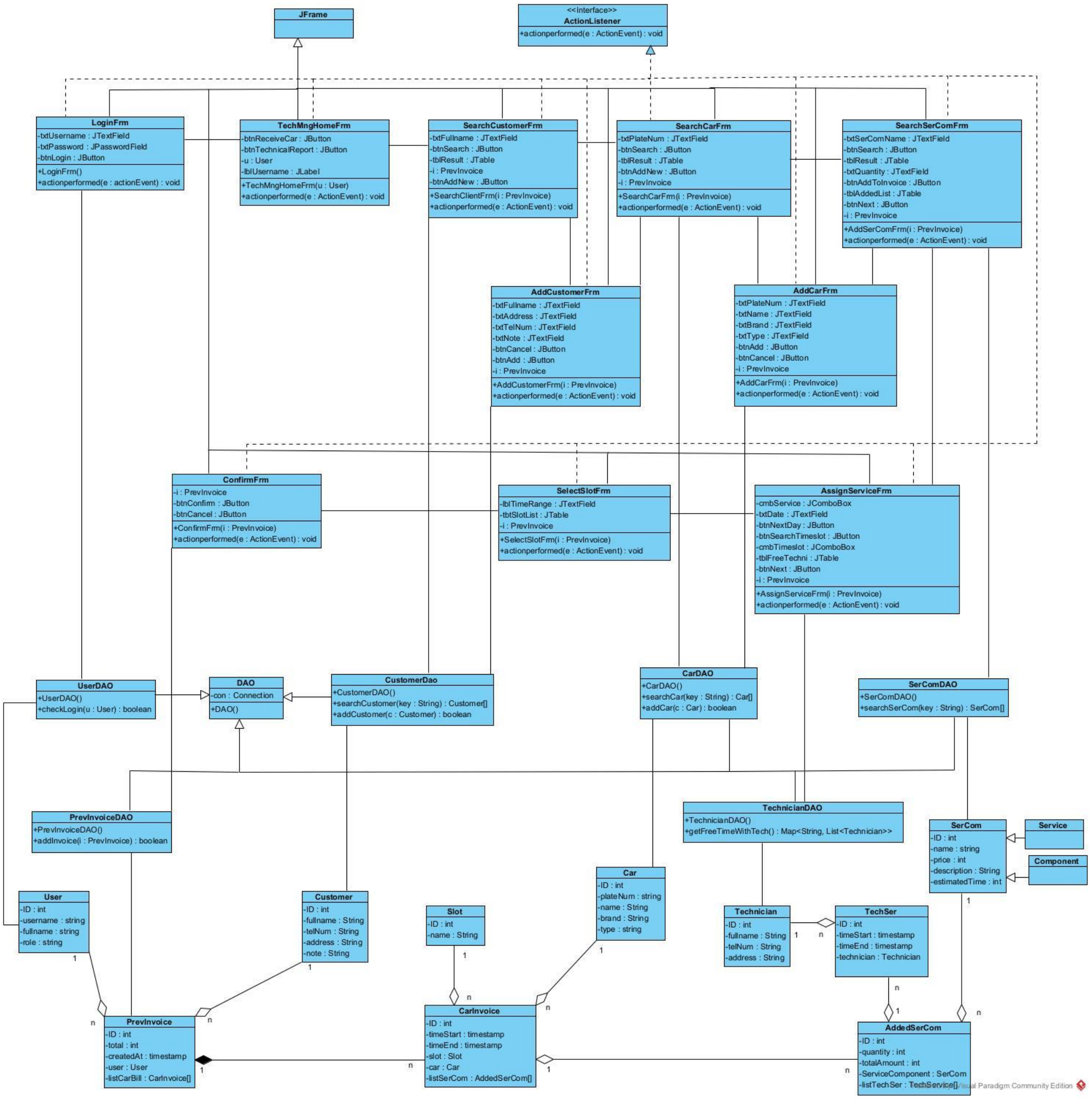
- TechMngHomeFrm is the home interface for the technical manager. It needs at least a button to go to the receive car function.
- SearchCustomerFrm is the interface to search and select the customer. It needs a text field to enter the keyword to search client by name, a button to search, a table to show the list of founded customers, and a button to add new client if the customer does not exist in the database.
- AddCustomerFrm is the interface to add new customer. It needs the text fields to enter the customer's information: name, address, phone number, note and a button to save, another button to cancel
- SearchCarFrm is the interface to search and select the car brought by customer. It needs a text field to enter the keyword to search car by plate number, a button to search, a table to show the list of founded cars, and a button to add new car if the car does not exist in the database
- AddCarFrm is the interface to add new car. It needs the input text fields to enter the car's information: plate number, name, brand, type and a button to save, another button to cancel
- SearchSerComFrm is the interface to search and add new service or component to repair the car. It needs a text field to search for service/component by keyword and a button to search, a table to show the list of founded services and components. A text field to insert quantity and click add to invoice, a table of list of added services/components. A button to click next.
- AssignServiceFrm is the interface to assign technicians to services. It has a dropdown menu to select the service, a text field displays the date and button to increase the date if needed when current date has no free technicians. A dropdown menu to select the timeslot and a table of list of free technicians in the selected timeslot
- SelectSlotFrm is the interface to select slot to put the car in. It has a table to show list of slots that are free and a text field to display the time range when the car starts and finishes the repair.
- ConfirmFrm is the interface to confirm invoice information. It displays

Control (DAO) classes:

- DAO is a general class of DAO. It has only the construction to connect to the database and provides the common connection for all inherited DAO classes in the system.
- UserDAO is the class for manipulating with database related to the User object. In this module, it needs a method to verify whether the login information is correct or not, it is checkLogin() method.

- CustomerDAO has two methods:
  - searchCustomer(): to search customer whose name contains the entered key
  - addCustomer(): to add new customer into the database
- CarDAO has two methods:
  - searchCar(): to search car which has plate number contains the entered key
  - addCar(): to add new car into the database
- SerComDAO has a method to search service/component which name contains the entered key searchSerCom()
- TechSerDAO has two methods:
  - getFreeTimeSlot(): to get a list of free time slot that has the duration bigger or equal to estimatedTime of the chosen service
  - addTechSer(): to add new TechSer into the database
- TechnicianDAO has a method to find list free timeslots and the list of free technicians in each timeslot: getFreeTimeslotWithTech ()
- SlotDAO has a method to find list of slots that are free between the time start of the earliest service and time end of the latest service: getFreeSlot()
- PrevInvoiceDAO has a method to add new invoice addInvoice()

Entity classes: Customer, User, PrevInvoice, Car, CarInvoice, Slot, SerCom, Technician, AddedSerCom, TechSerivce



9. Sequence diagram of design phase

Scenario version 3

1. The customer brings car to fix
2. The technical manager enters username, password and clicks on the login button on LoginFrm
3. The method actionPerformed() of LoginFrm is called
4. The method actionPerformed() calls User to create an User object
5. The class User packs the information into an User object
6. The class User returns User object to the method actionPerformed()
7. The method actionPerformed() calls method checkLogin() of the class UserDAO

8. The method checkLogin() checks the login information
9. The method checkLogin() calls the class User set more two attributes name, role
10. The class User calls its method setName(), setRole()
11. The class User returns the User object to the method checkLogin()
12. The method checkLogin() returns the results to the actionPerfomed()
13. The method actionPerformed() calls the class TechMngHomeFrm
14. The constructor TechMngHomeFrm() is called
15. The TechMngHomeFrm is shown to the technical manager
16. The technical manager clicks on the receive car button
17. The method actionPerformed() is called
18. The method actionPerformed() calls the SearchCustomerFrm
19. The constructor SearchCustomerFrm() is called
20. The interface SearchCustomerFrm is shown to technical manager
21. The Technical manager asks the customer name
22. The customer answers the technical manager
23. The technical manager enters the customer’s name and clicks search
24. The method actionPerformed() of class SearchCustomerFrm is called
25. The method actionPerformed() calls the method searchCustomer() of the class CustomerDAO
26. The method searchCustomer() executes
27. The method searchCustomer() calls the class Customer to pack the results
28. The class Customer packs Customer objects
29. The class Customer returns the packed objects to the method searchCustomer()
30. The method searchCustomer() returns the results to the method actionPerformed()
31. The method actionPerformed() displays the results on the SearchCustomerFrm to the technical manager
32. The technical manager clicks on the row corresponding to the right customer
33. The method actionPerformed() of the class SearchCustomerFrm is called
34. The method actionPerformed() calls the class PrevInvoice to add customer information to it
35. The PrevInvoice calls the method setCustomer()
36. The class PrevInvoice returns the packed object to the method actionPerformed()
37. The method actionPerformed() calls the class SearchCarFrm
38. The constructor SearchCarFrm() is called
39. The interface SearchCarFrm if shown to the technical manager
40. The technical manager asks the customer about the car’s plate number
41. The customer answers the technical manager

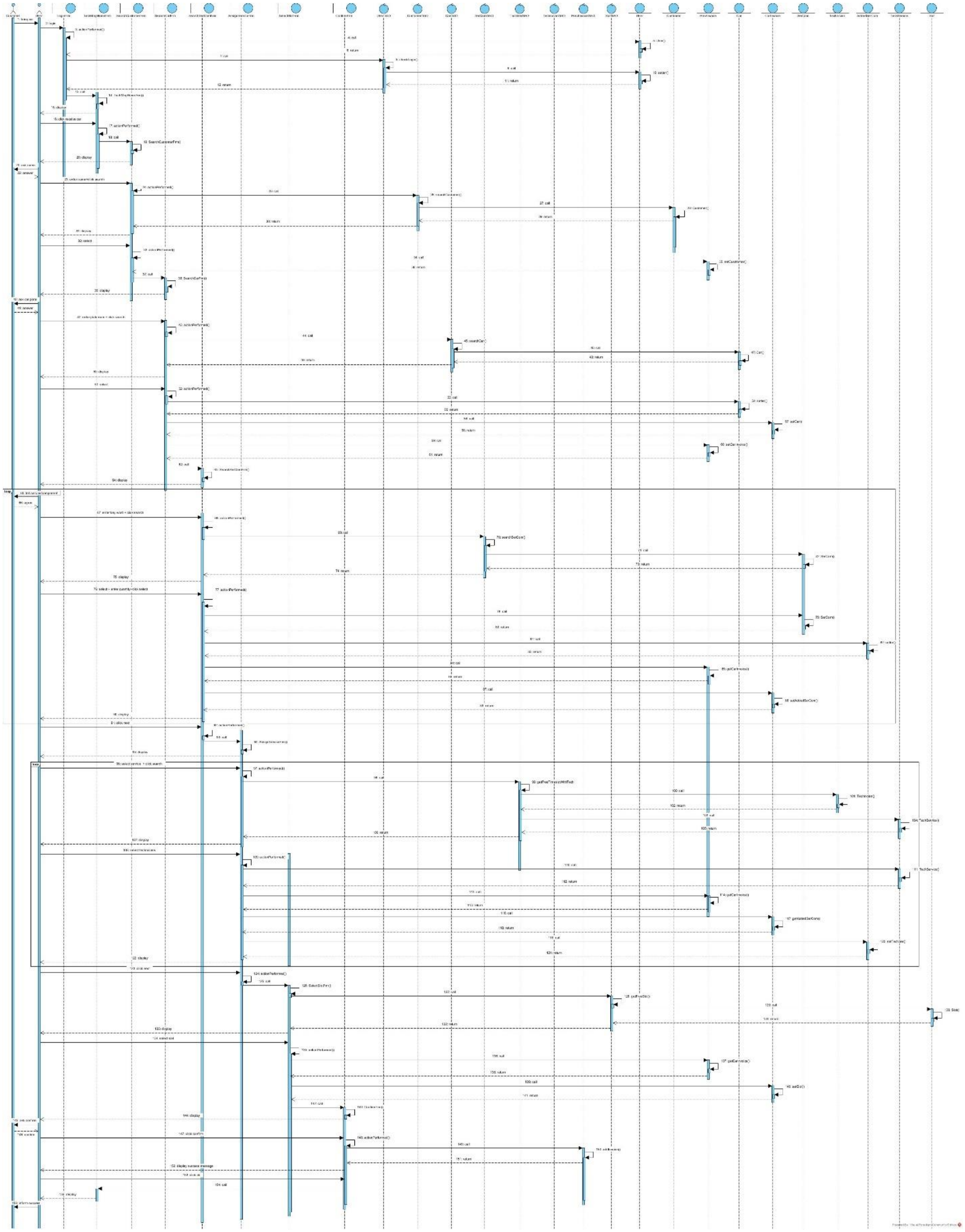
42. The technical manager enters the car’s plate number and click search
43. The method actionPerformed() of class SearchCarFrm is called
44. The method actionPerformed() calls the method searchCar() of the class CarDAO
45. The method searchCar() executes
46. The method searchCar() calls the class Car to pack the results
47. The class Car packs result objects
48. The class Car returns the packed object to the method searchCar()
49. The method searchCar() return results to the method actionPerformed()
50. The method actionPerformed() displays results on the SearchCarFrm to the technical manager
51. The technical manager clicks on the row corresponding to the right car
52. The method actionPerformed() of class SearchCarFrm is called
53. The method actionPerformed() calls the class Car to pack a car object
54. The class Car calls setter() methods
55. The class Car returns result to actionPerformed()
56. The actionPerformed() calls class CarInvoice to set a new car
57. The function setCar() of CarInvoice executes
58. The CarInvoice returns result to actionPerformed()
59. The method actionPerformed() calls PrevInvoice to set a new car invoice
60. The method setCarInvoice() executes
61. The class PrevInvoice returns to actionPerformed()
62. The method actionPerformed() calls the class SearchSerComFrm
63. The constructor SearchSerComFrm() is called
64. The SearchSerComFrm is shown to the technical manager
65. The technical manager tells customer about the required service/component and its quantity
66. The customer agrees
67. The technical manager enters the key word and click search
68. The method actionPerformed() of class SearchSerComFrm is called
69. The method actionPerformed() calls the method searchSerCom() of the class SerComDAO
70. The method searchSerCom() executes
71. The method searchSerCom() calls the class ServiceComponent to pack the results
72. The class ServiceComponent packs the ServiceComponent objects
73. The class ServiceComponent returns the packed object to the method searchSerCom()
74. The method searchSerCom() returns results to the method actionPerformed()

75. The method actionPerformed() displays results on the SearchSerComFrm to the technical manager
76. The technical manager selects the correct service/component, inserts quantity and click add to invoice
77. The method actionPerformed() of class SearchSerComFrm is called
78. The method actionPerformed() calls the class ServiceComponent to pack an object
79. The ServiceComponent packs the object
80. The ServiceComponent returns result to actionPerformed()
81. The actionPerformed() calls class AddedSerCom to pack a new object
82. The class AddedSerCom calls methods setQuantity() and setSerCom()
83. The class AddedSerCom returns result to actionPerformed()
84. The actionPerformed() calls class PrevInvoice
85. The class PrevInvoice calls the method getCarInvoice()
86. The method getCarInvoice() returns CarInvoice object to actionPerformed()
87. The actionPerformed() calls class CarInvoice to add the new AddedSerCom object to the listSerCom[]
88. The CarInvoice calls the methods setAddedSerCom()
89. The class CarInvoice returns result to actionPerformed() of SearchSerComFrm
90. The class SearchSerComFrm displays result to technical manager
91. The technical manager clicks next button
92. The method actionPerformed() of SearchSerComFrm is called
93. The method actionPerformed() calls the class AssignServiceFrm
94. The constructor AssignServiceFrm() is called
95. The interface AssignServiceFrm is shown to the technical manager
96. The technical manager selects the service, keeps the current date and clicks search timeslot
97. The method actionPerformed() of class AssignServiceFrm is called
98. The method actionPerformed() calls the method getFreeTimeslotWithTech() of the class TechnicianDAO
99. The method getFreeTimeslotWithTech() executes
100. The method getFreeTimeslotWithTech() calls class Technician to pack objects
101. The class Technician packs the objects
102. The class Technician returns results to getFreeTimeslotWithTech()
103. The method getFreeTimeslotWithTech() calls class TechService
104. The class TechService packs the objects
105. The class TechService returns results to getFreeTimeslotWithTech()

106. The method getFreeTimeslotWithTech() calculates the timeslots which have free technicians and long enough to do the service, then return results to actionPerformed()
107. The actionPerformed() displays results with free timeslots and free technicians in each timeslot to technical manager
108. The technical manager selects technicians
109. The method actionPerformed() of AssignServiceFrm is called
110. The method actionPerformed() calls the class TechService to create a new object
111. The class TechService packs the object
112. The class TechService returns result to actionPerformed()
113. The class actionPerformed() calls class PrevInvoice to get CarInvoice
114. The method getCarInvoice() executes
115. The method getCarInvoice() returns result to actionPerformed()
116. The method actionPerformed() calls class CarInvoice to get the added service
117. The method getAddedSerCom() of class CarInvoice executes
118. The class CarInvoice returns result to actionPerformed()
119. The method actionPerformed() calls class AddedSerCom to set a new TechService object to the list of TechService
120. The class AddedSerCom calls method setTechSer()
121. The class AddedSerCom returns result to actionPerformed() of AssignServiceFrm
122. The AssignServiceFrm displays results to technical manager
123. The technical manager click next button
124. The method actionPerformed() of AssignServiceFrm is called
125. The method actionPerformed() calls the class SelectFlotFrm
126. The constructor SelectSlotFrm() is called
127. The SearchSlotFrm() calls the method getFreeSlot() of SlotDAO
128. The method getFreeSlot() executes
129. The method getFreeSlot() calls class Slot to pack the objects
130. The class Slot packs the object
131. The class Slot returns results to getFreeSlot()
132. The method getFreeSlot() return result to SearchSlotFrm()
133. The interface SelectSlotFrm is shown to technical manager to select slot
134. The technical manager selects a slot
135. The method actionPerformed() of SelectSlotFrm is called
136. The method actionPerformed() calls the method getCarInvoice() of class PrevInvoice
137. The method getCarInvoice() executes

138. The method getCarInvoice() returns CarInvoice object to actionPerformed()
139. The method actionPerformed() calls class CarInvoice to set a new slot
140. The class CarInvoice calls method setSlot()
141. The class CarInvoice returns result to actionPerformed()
142. The method actionPerformed() calls the class ConfirmFrm
143. The cconstructor ConfirmFrm() is called
144. The interface ConfirmFrm is shown to the technical manager
145. The technical manager repeats the invoice information to the client and asks him to confirm.
146. The customer confirms it.
147. The technical manager clicks on the confirm button.
148. The method actionPerformed() of the class ConfirmFrm is called.
149. The method actionPerformed() calls the method addInvoice() of the class PrevInvoiceDAO.
150. The method addInvoice() executes.
151. The method addInvoice() returns the turn to the method actionPerformed()
152. The method actionPerformed() displays a success message
153. The technical manager clicks on the OK button of the message.
154. The method actionPerformed() recalls the interface TechMngHomeFrm.
155. The interface TechMngHomeFrm is shown to the technical manager.
156. The technical manager confirms the success to the customer.

9.2. Sequence diagram of design phase



10. Black-box testing

10.1. Black-box test plan

No.	Module	Test case
1	Receiving car	Customer and car already existed, the garage has free technicians and free slots
2		Customer doesn’t exist, car exists, the garage has free technicians and free slots
3		Customer exists, car does not exist, the garage has free technicians and free slots
4		Customer and car don’t exist, the garage has free technicians and free slots
5		Customer and car already existed, the garage has no free technicians but has free slots
6		Customer doesn’t exist, car exists, the garage has no free technicians but has free slots
7		Customer exists, car does not exist, the garage has no free technicians but has free slots
8		Customer and car don’t exist, the garage has no free technicians but has free slots
9		Customer and car already existed, the garage has free technicians but no free slots
10		Customer doesn’t exist, car exists, the garage has free technicians but no free slots
11		Customer exists, car does not exist, the garage has free technicians but no free slots
12		Customer and car don’t exist, the garage has free technicians but no free slots
13		Customer and car already existed, the garage has no free technicians and no free slots
14		Customer doesn’t exist, car exists, the garage has no free technicians and no free slots
15		Customer exists, car does not exist, the garage has no free technicians and no free slots
16		Customer and car don’t exist, the garage has no free technicians and no free slots
17		Add a service/component two times to the invoice

10.2. Standard test case

Database before testing

- tblUse

ID	username	password	fullname	role
1	techmanager	123	Nguyen Van Anh	Technical Manager
2	manager	1234	nguyen duc hoa	Manager
3	thucashier	321	Le Thi Thu	Cashier

- tblCustomer

ID	fullname	telNum	address	note
1	Tran Binh An	0897653458	Nam Dinh	Kho tinh
2	Nguyen Tran Binh An	0112233445	Ha Dong	Sam soi
3	Tran Binh An Nhien	1223344556	Thanh Oai	
4	Trinh Van Quyet	0765987567	Thanh Oai	Hay ki keo

- tblCar

ID	PlateNumber	Name	Brand	Type
1	30G-123	Mazda G6	Mazda	4 seats
2	30G-1234	Toyota X5	Toyota	7 seats
3	27G-123	Mazda G6	Mazda	4 seats

- tblSlot

ID	Name
1	A01
2	A02
3	A03
4	B01
5	B02
6	B03
7	C01
8	C02

9	C03

- tblTechnician

ID	fullname	telNum	address
1	Nguyen Duc Canh	0579367892	Nam Dinh
2	Le Phong	0456834694	Hanoi
3	Pham Thi Ca	0964385735	Yen bai
4	Tran Van Hieu	0912345678	Yen Nghia
5	Nguyen Thi Lan	0934567890	Hoai Duc
6	Le Van Minh	0987654321	Chuong My
7	Pham Van Nam	0901122334	Ho Chi Minh
8	Bui Thi Hoa	0977008899	Can Tho

- tblSerCom

ID	Name	Price	Description	EstimatedTime
1	Oil Change	100000	Change engine oil	60
2	Tire Rotation	200000	Rotate tires for even wear	90
3	Tire Replacement	200000	Replace all 4 tires	90
4	Brake Inspection	250000	Check brake condition	30
5	Brake Pads Replacement	150000	Change front brake pads	45
6	Battery Check	100000	Test battery health	30
7	Coolant Flush	250000	Flush and refill coolant	40
8	Transmission Fluid Change	300000	Replace transmission fluid	60
9	Engine Diagnostic	150000	Run full engine check	20
10	Alternator Replacement	120000	Replace faulty alternator	90
11	Spark Plug Replacement	300000	Install new spark plugs	60
12	Oil Filter Replacement	199000	Replace oil filter	60

13	Battery Replacement	300000	Replace with new battery	90
14	Air Filter Replacement	500000	Replace engine air filter	60
15	Cabin Air Filter Replace	350000	Replace cabin filter	120
16	Fuel Filter Replacement	350000	Change fuel filter	90

- tblService

ID
1
2
3
4
5
6
7
8
9
10
11

- tblComponent

ID
12
13
14
15
16

- tblTechService

ID	timeStart	timeEnd	addedServiceID	technicianID
1	2025-05-23 08:00	2025-05-23 09:30	1	1
2	2025-05-23 08:00	2025-05-23 09:30	1	2
3	2025-05-23 08:35	2025-05-23 09:35	2	3
4	2025-05-23 08:35	2025-05-23 10:05	3	4

-tblCarInvoice

ID	carID	invoiceID	slotID
1	3	1	1
2	2	2	2

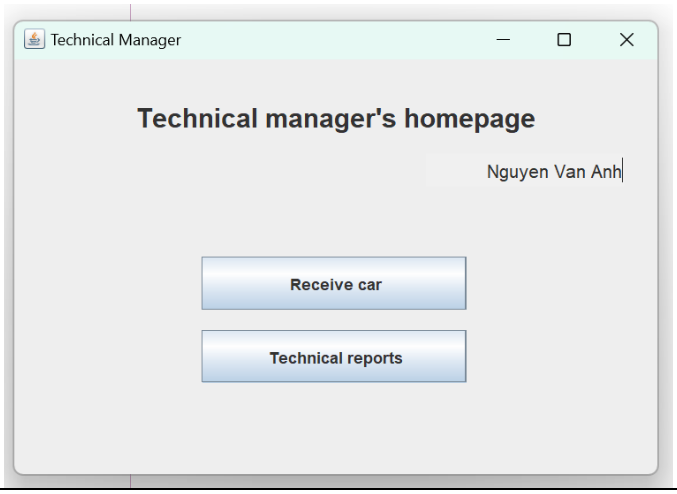
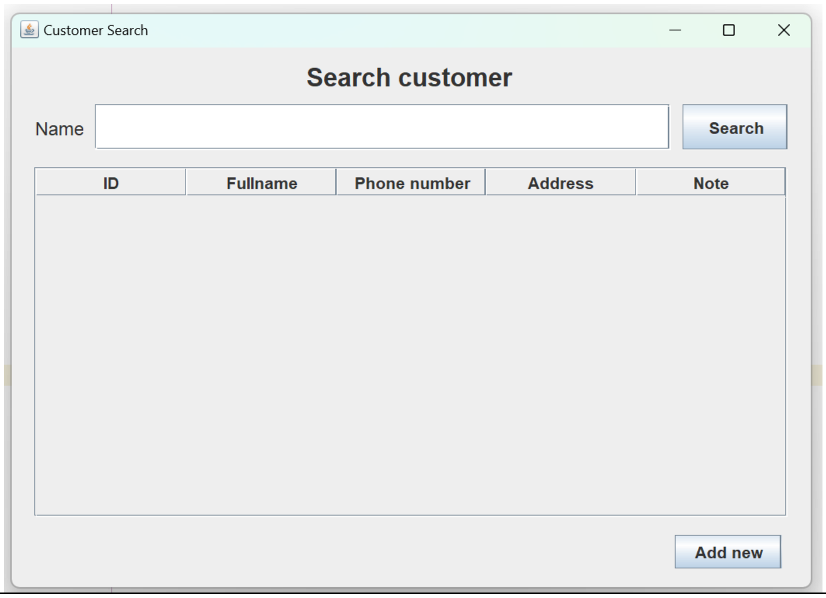
- tblPrevInvoice

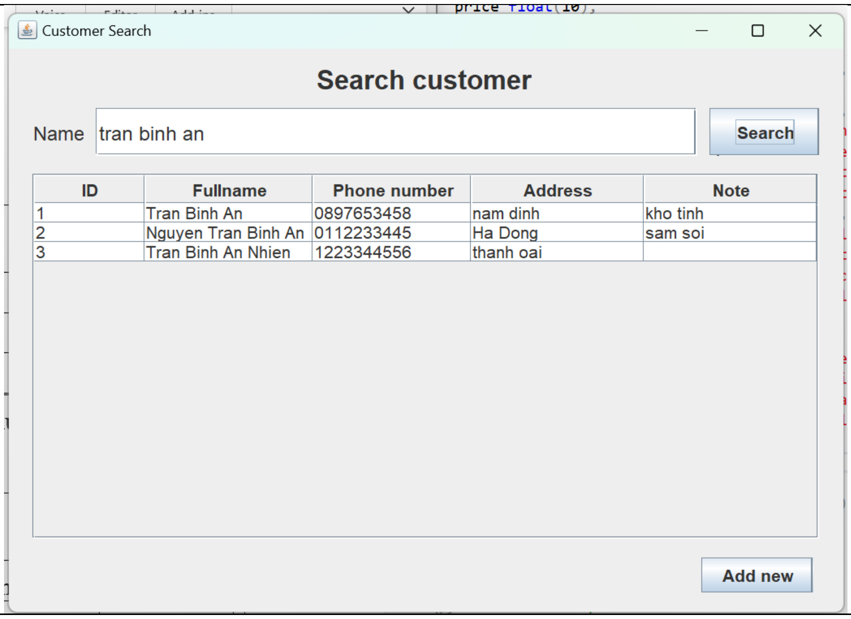
ID	createdAt	userID	customerID
1	2025-05-23 07:58	1	3
2	2025-05-23 08:30	1	2

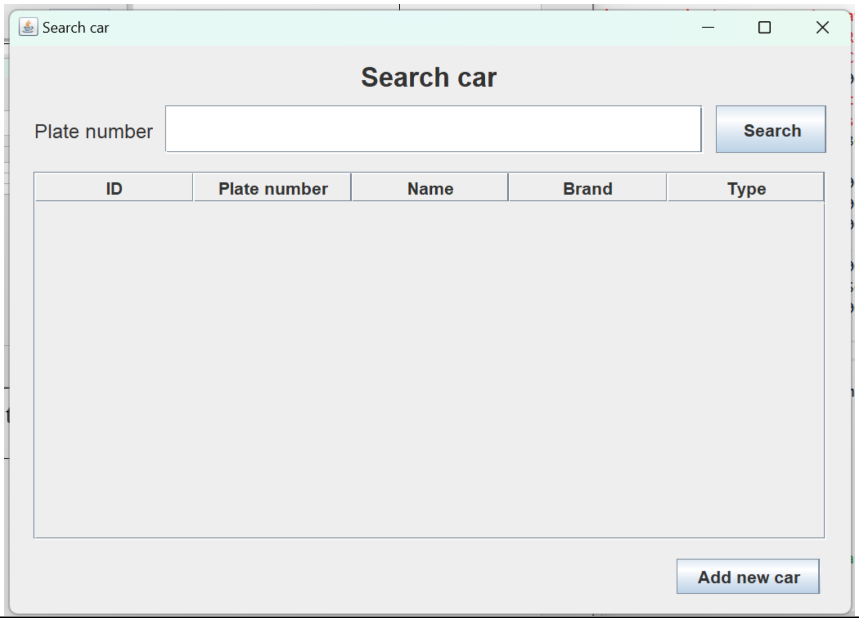
- tblAddedSerCom

ID	quantity	carInvoiceID	sercomID
1	1	1	2
2	2	2	1
3	1	2	10

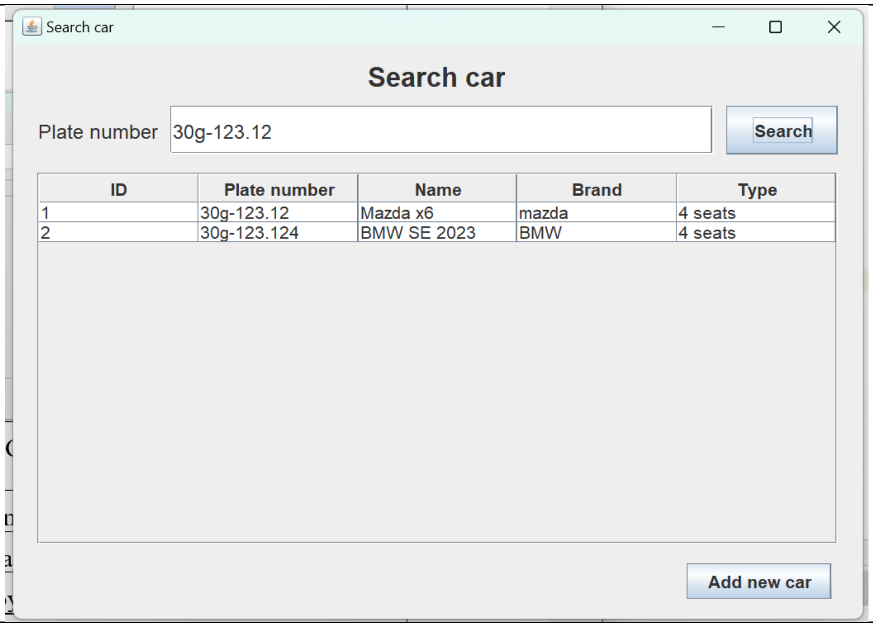
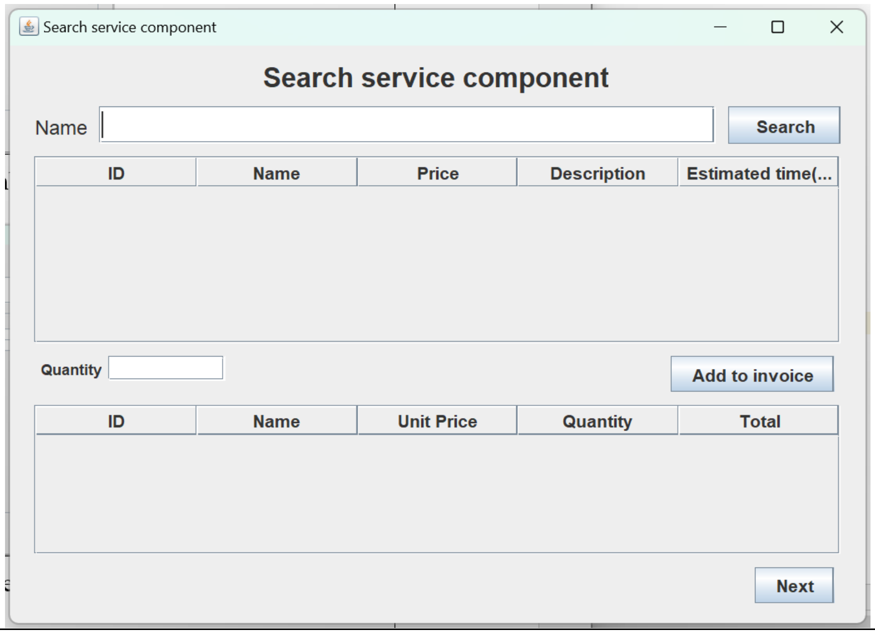
Testing scenario and expected results

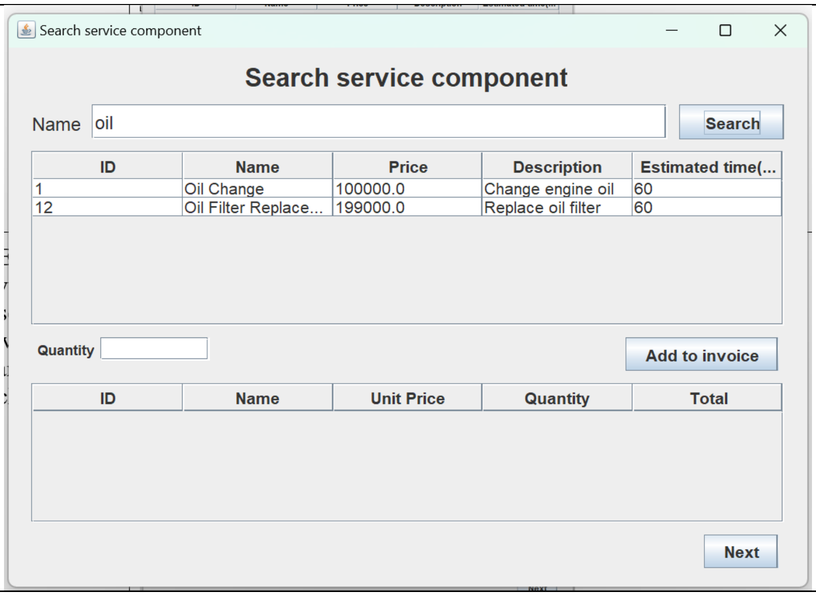
Scenario	Expected results
1. Login with username = techmanager, password = 123	- Technical manager homepage appears  Technical Manager Technical manager's homepage Nguyen Van Anh Receive car Technical reports
2. Click receive car	- Search customer UI appears  Customer Search Search customer Name Search ID Fullname Phone number Address Note Add new
3. Enter name= Tran Binh An, click search	- Table result with customer’s name = “Tran Binh An” appears

<table>
<tr>
<td></td>
<td>


Name: tran binh an
<table>
<tr><th>ID</th><th>Fullname</th><th>Phone number</th><th>Address</th><th>Note</th></tr>
<tr><td>1</td><td>Tran Binh An</td><td>0897653458</td><td>nam dinh</td><td>kho tinh</td></tr>
<tr><td>2</td><td>Nguyen Tran Binh An</td><td>0112233445</td><td>Ha Dong</td><td>sam soi</td></tr>
<tr><td>3</td><td>Tran Binh An Nhien</td><td>1223344556</td><td>thanh oai</td><td></td></tr>
</table>
</td>
</tr>
<tr>
<td>4. Select the first row</td>
<td>
- Search car UI appear


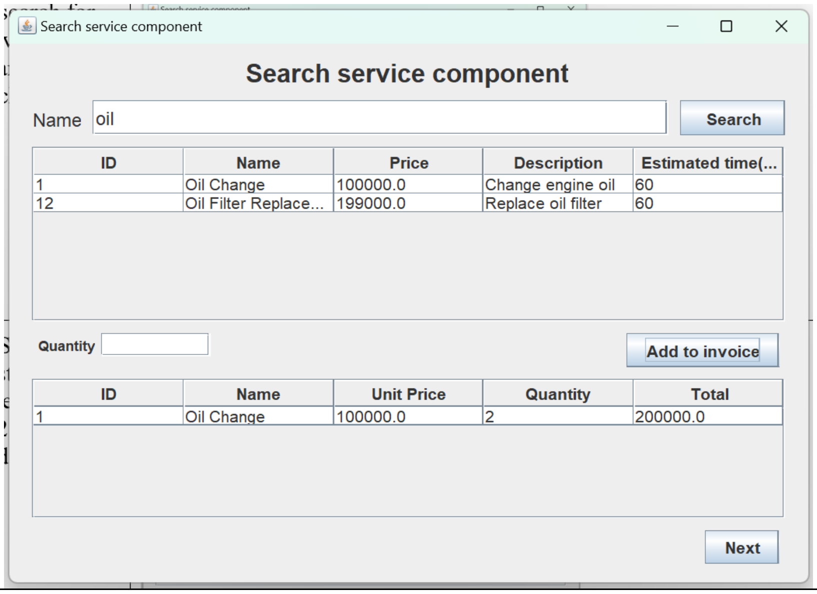
<table>
<tr><th>ID</th><th>Plate number</th><th>Name</th><th>Brand</th><th>Type</th></tr>
</table>
</td>
</tr>
<tr>
<td>5. Enter plate number = 30G-123.12</td>
<td>- Table result with car’s plate number = ‘30g-123.12’ appears</td>
</tr>
</table>

<table>
<tr><td></td><td>
Search car
Search car
Plate number 30g-123.12 Search
<table>
<tr><th>ID</th><th>Plate number</th><th>Name</th><th>Brand</th><th>Type</th></tr>
<tr><td>1</td><td>30g-123.12</td><td>Mazda x6</td><td>mazda</td><td>4 seats</td></tr>
<tr><td>2</td><td>30g-123.124</td><td>BMW SE 2023</td><td>BMW</td><td>4 seats</td></tr>
</table>
</td></tr>
<tr><td>6. Select row with plate number = 30G-123.12</td><td>Search and add service/component UI appear

Search service component
Search service component
Name
<table>
<tr><th>ID</th><th>Name</th><th>Price</th><th>Description</th><th>Estimated time(...</th></tr>
</table>
Quantity
<table>
<tr><th>ID</th><th>Name</th><th>Unit Price</th><th>Quantity</th><th>Total</th></tr>
</table>
</td></tr>
<tr><td>7. Enter keyword = oil to search for service Oil change and click search</td><td>Search service/component UI for keyword=oil</td></tr>
</table>

<table>
<tr>
<td></td>
<td>
Search service component
Search service component
Name oil

<table>
<tr><th>ID</th><th>Name</th><th>Price</th><th>Description</th><th>Estimated time(...</th></tr>
<tr><td>1</td><td>Oil Change</td><td>100000.0</td><td>Change engine oil</td><td>60</td></tr>
<tr><td>12</td><td>Oil Filter Replace...</td><td>199000.0</td><td>Replace oil filter</td><td>60</td></tr>
</table>
Quantity

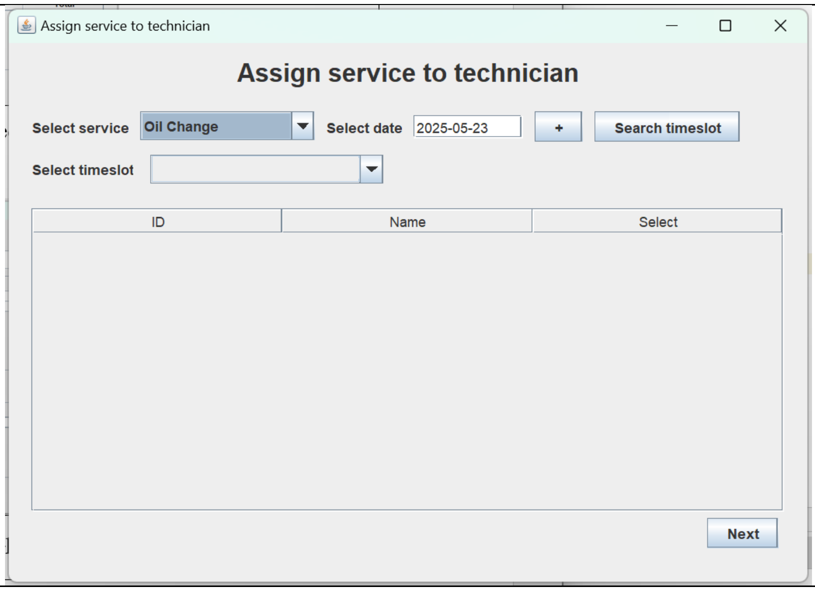
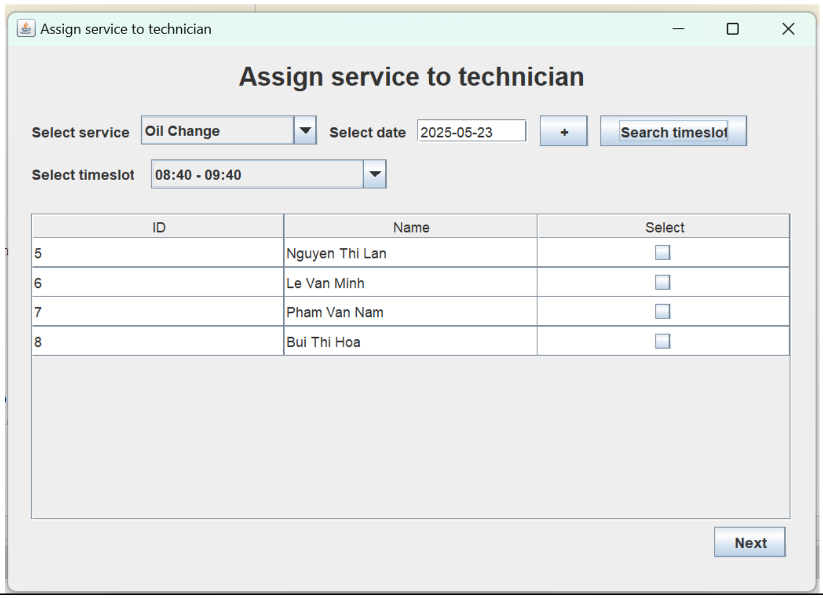
<table>
<tr><th>ID</th><th>Name</th><th>Unit Price</th><th>Quantity</th><th>Total</th></tr>
</table>

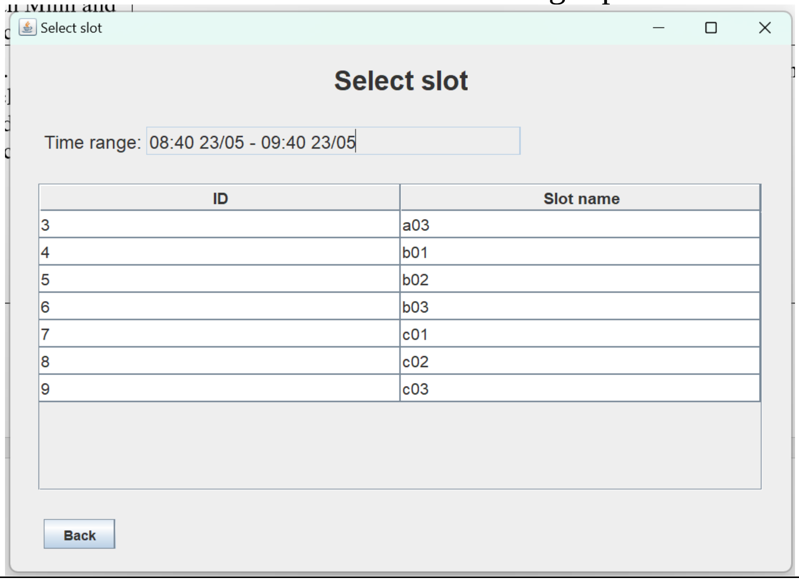
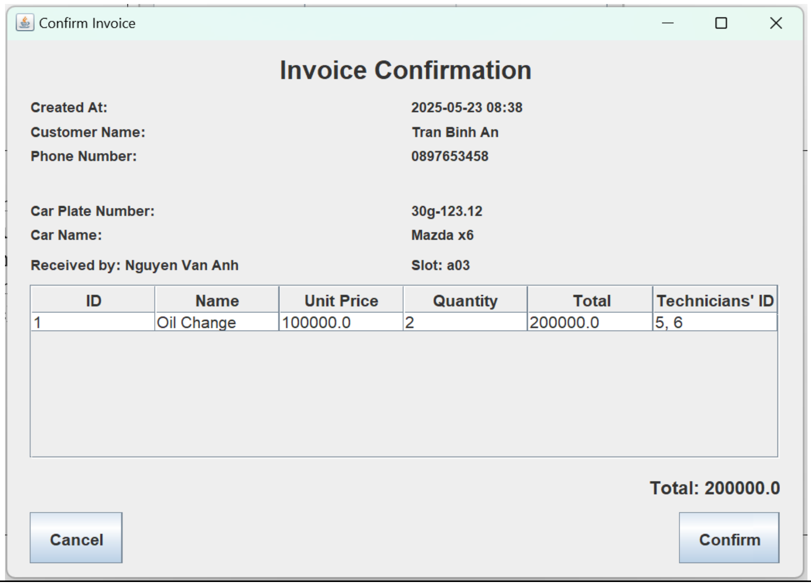
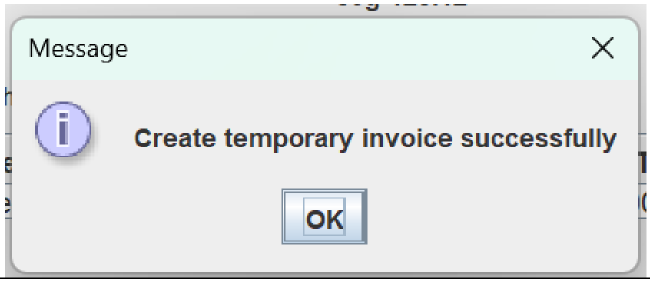
</td>
</tr>
<tr>
<td>8. Select the first row, insert quantity is 2 and click add to invoice</td>
<td>- New service/component is added to table list of added services/components

Search service component
Search service component
Name oil

<table>
<tr><th>ID</th><th>Name</th><th>Price</th><th>Description</th><th>Estimated time(...</th></tr>
<tr><td>1</td><td>Oil Change</td><td>100000.0</td><td>Change engine oil</td><td>60</td></tr>
<tr><td>12</td><td>Oil Filter Replace...</td><td>199000.0</td><td>Replace oil filter</td><td>60</td></tr>
</table>
Quantity

<table>
<tr><th>ID</th><th>Name</th><th>Unit Price</th><th>Quantity</th><th>Total</th></tr>
<tr><td>1</td><td>Oil Change</td><td>100000.0</td><td>2</td><td>200000.0</td></tr>
</table>

</td>
</tr>
<tr>
<td>9. Click next</td>
<td>Assign service to technician UI appears:</td>
</tr>
</table>

	 Assign service to technician Assign service to technician Select service: Oil Change Select date: 2025-05-23 Select timeslot: ID | Name | Select
10. Click select timeslot to search for free timeslots and technicians for service Oil change	- Free timeslots starting from current time and have duration equals to estimated time of service Oil change appears with free technicians in each timeslot  Assign service to technician Assign service to technician Select service: Oil Change Select date: 2025-05-23 Select timeslot: 08:40 - 09:40 ID | Name | Select 5 | Nguyen Thi Lan | ☐ 6 | Le Van Minh | ☐ 7 | Pham Van Nam | ☐ 8 | Bui Thi Hoa | ☐

12. Select the technicians Nguyen Thi Lan and Le Van Minh and click next	The list of free slots from the time being repaired and done appears 
13. Select the slot A03 in the first row	Confirm temporary invoice UI appears 
14. Click confirm button	A successful message appears 

Select slot

Select slot

Time range: 08:40 23/05 - 09:40 23/05

ID	Slot name
3	a03
4	b01
5	b02
6	b03
7	c01
8	c02
9	c03

Confirm Invoice

Invoice Confirmation

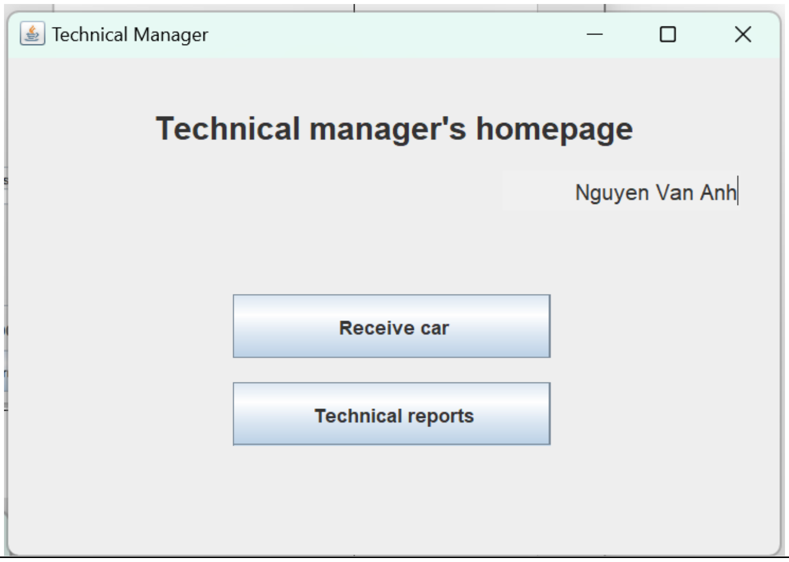
Created At:	2025-05-23 08:38
Customer Name:	Tran Binh An
Phone Number:	0897653458
Car Plate Number:	30g-123.12
Car Name:	Mazda x6
Received by: Nguyen Van Anh	Slot: a03

ID	Name	Unit Price	Quantity	Total	Technicians' ID
1	Oil Change	100000.0	2	200000.0	5, 6

Total: 200000.0

Message

Create temporary invoice successfully

14. Click OK	

Technical Manager

Technical manager's homepage

Nguyen Van Anh

Database after testing

- tblTechService

ID	timeStart	timeEnd	addedServiceID	technicianID
1	2025-05-23 08:00	2025-05-23 09:30	1	1
2	2025-05-23 08:00	2025-05-23 09:30	1	2
3	2025-05-23 08:35	2025-05-23 09:35	2	3
4	2025-05-23 08:35	2025-05-23 10:05	3	4
5	2025-05-23 08:40	2025-05-23 09:40	4	5
6	2025-05-23 08:40	2025-05-23 09:40	4	6

- tblCarInvoice

ID	carID	invoiceID	slotID
1	3	1	1
2	2	2	2
3	1	3	3

- tblPrevInvoice

ID	createdAt	userID	customerID
1	2025-05-23 07:58	1	3

2	2025-05-23 08:30	1	2
3	2025-05-23 08:38	1	1

- tblAddedSerCom

ID	quantity	carInvoiceID	sercomID
1	1	1	2
2	2	2	1
3	1	2	10
4	2	3	1

- Others tables do not change