MONASH University

FIT5042 Design Report (assignment)

smallwood

Smallwood Carsales

Peng Zhao: 25661043

Contents **Page**

1. Overview

1.1 Introduction

Smallwood Carsales Company is a medium-sized car selling company located in Melbourne City. Started in 2010, it has been successfully running its business for over five years. In this period, all the business transactions have been completely handled based on paper work. With the company keeping extending its business, a requirement for a professional, enterprise-level car sale system emerges and leads to the initialization of this project.

1.2 Goal

The goal of this project (referred as Smallwood Carsales afterward) is to develop an enterprise-level car sale system which covers basic business process involved in online car trading, including customer/salesperson management, car catalog management and online trading management. Through this application, customers are able to search, view and purchase cars while salespersons can manage car catalog and maintain sale records. Finally, Smallwood Carsales is committed to provide a convenient yet secure online car sale experience to both car sellers and consumers.

The deliverables from Smallwood Carsales project will be a comprehensive system prototype meeting above goals, comprised of database design, enterprise server implementation and clients (both desk-based application and website) as well as corresponding development documentations.

1.3 Scopes

The major functions covered in this project includes user management, car catalog management and sales management (see details in Section 2 Functional diagram).

To be noticed, sale payment would NOT be covered in this prototype. It would be the responsibility of salespersons in Smallwood Company to handle all sale related payment. The system would only keep a record about the payment status of each sale (see details in section 2.3 Sales Management).

2. Functional diagram

There are three primary function areas in this system: User Management, Car Catalog Management and Sales Management (See Figure 1). Meanwhile, there are two types of user in the system: Salesperson and Customer, each coming with specified functionalities.

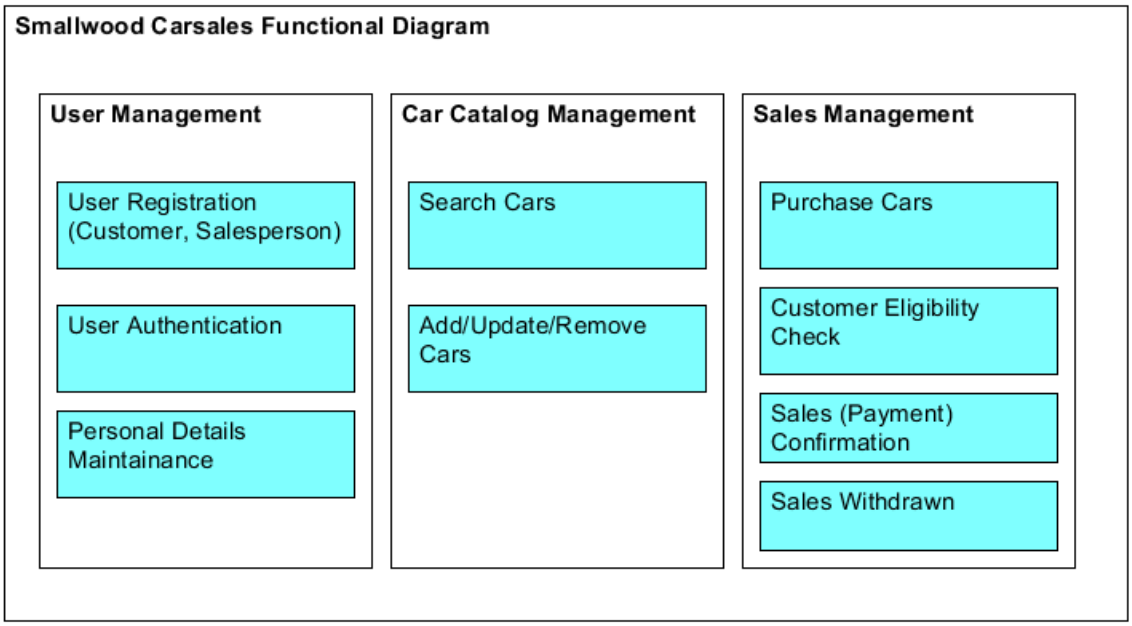

Figure 1: Smallwood Carsales Functional Diagram

2.1 User Management

This module is responsible for user related functionalities.

1) User Registration

There are three types of users: admin, salesperson and customer. Admin is pre-defined user coming with the system. Admin can register salespersons and maintain their accounts. Similarly, salespersons are responsible for customer registration and maintenance (see more in section 6 User Stories).

Customers and salespersons can also be removed from the system by salesperson and admin respectively, but only when they are not involved in any sale records. This is necessary to ensure data integrity.

2) User Authentication

Registered users are authenticated to access corresponding services. It ensures our services are only available for specified users.

3) Personal Details Maintenance

Users are able to view their personal details stored in our system. Admin user is able to update salesperson personal details while salesperson can update customers. The up-to-date personal information is crucial for the system to provide competent services.

2.2 Car Catalog Management

Car catalog management keeps records of cars in storage and available in the system. It also provide matched car information as users query.

1) Search Cars

Customers and salespersons are able to search cars from our storage based on partial properties of cars, like type, make or model. The system would return information for all cars matching user search criteria, like VIN, make, mode, price etc.

Customers can only find cars which are available for them to buy. In other words, only cars in storage, without being sold out, are returned to customers.

Differently, salespersons can view all car details in the system, no matter if they have been sold. This is convenient for them to view previous sales details.

2) Add/Update/Remove Cars

Salesperson are responsible to maintain car information stored in the system. The system would provide interface for them to add or remove cars as well as update some details. However, they can only modify or remove cars which are currently in storage. Otherwise, removing sold cars would lead to database corruption and previous sales lost.

2.3 Sales Management

Sales management is the core function in the project. The system would enable customers to make purchase through interface and meanwhile support salesperson's work to handle sale processes.

1) Purchase Cars

Customers are able to purchase cars based on their selection. The system would accordingly update sale records in the system to matching these behaviors.

2) Customer Eligibility Check

The system should enforce necessary eligibility checks before forwarding users purchase request in accords to business rules. In Smallwood Carsales, only can customers make new purchases before they have paid off all previous ones.

3) Sales (Payment) Confirmation

Although the system itself does not handle payment process for sales, it would still keep record about payment status for each sale. It is necessary to enable customer eligibility check.

4) Sales Withdrawn

Salespersons are able to withdrawn sales at any stage in case customers cancel the purchases. These sale records would by simply removed from database. Related cars are hereby available for customers again.

2. 4 Use Cases

Based on previous analysis of Smallwood Carsales' business, the following use cases are populated.

1) Abstract User

An abstract user is used to represent common functions shared by both customer and salesperson.

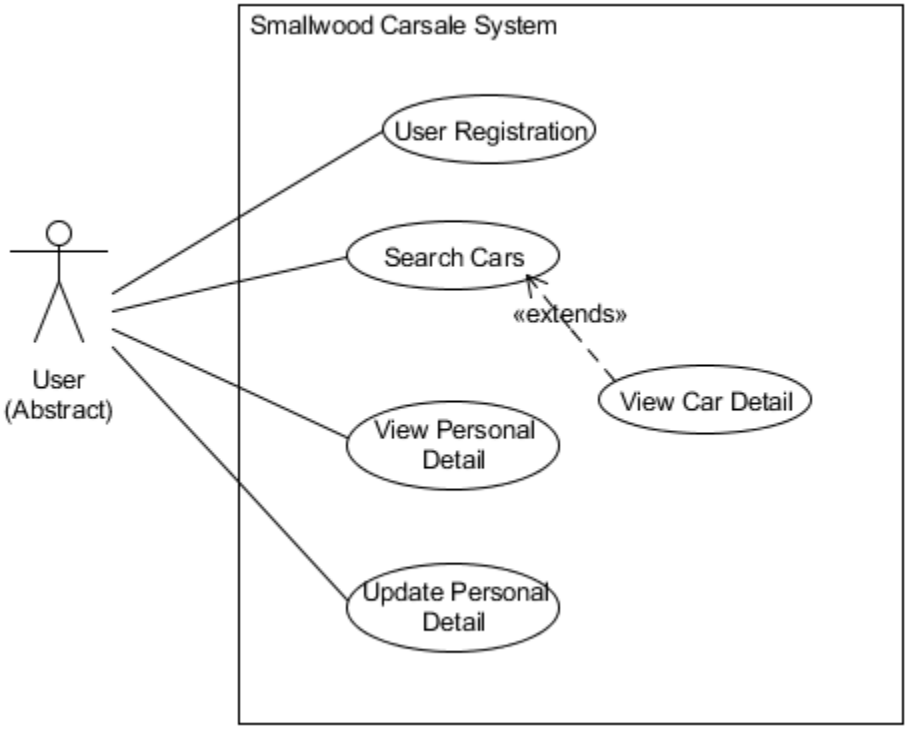

Figure 2: Abstract User Use Case

2) Customer

Apart from the common functions defined above, customers are also able to buy cars and view their previous purchase records. Before each purchase can be made, the system is responsible to check car storage and previous payment information.

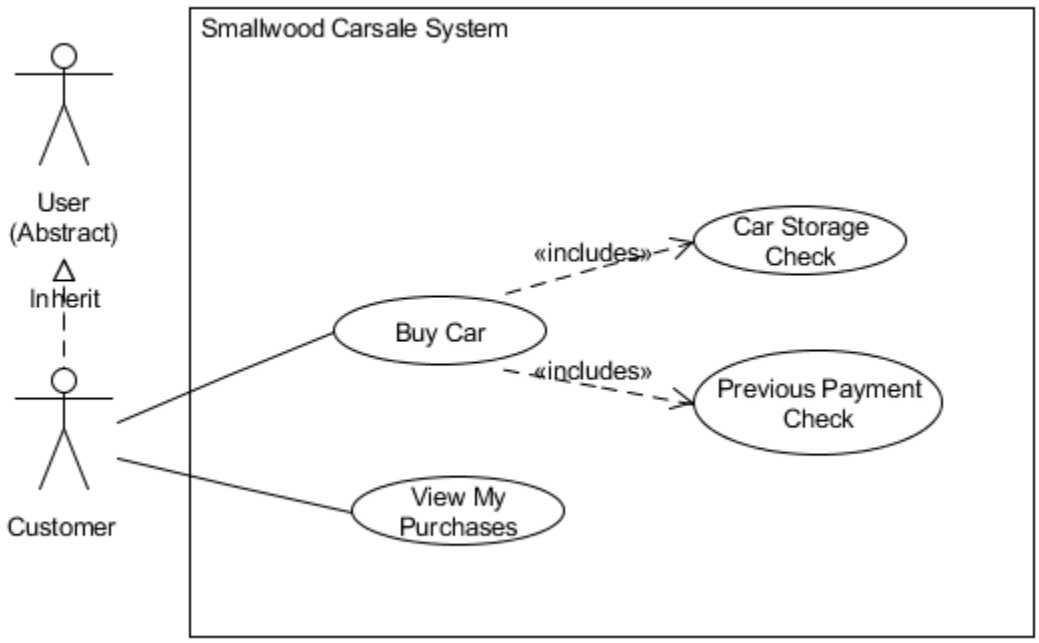

Figure 3: Customer Use Case

3) Salesperson

Salesperson in Smallwood Carsales Company are dedicated and professional staff. They are not allowed to buy cars from the system. Their responsibilities will focus on car catalog maintenance and sales management. They also are able to search and view customer details (including previous and current purchases) in order to make judgement on sale handlings.

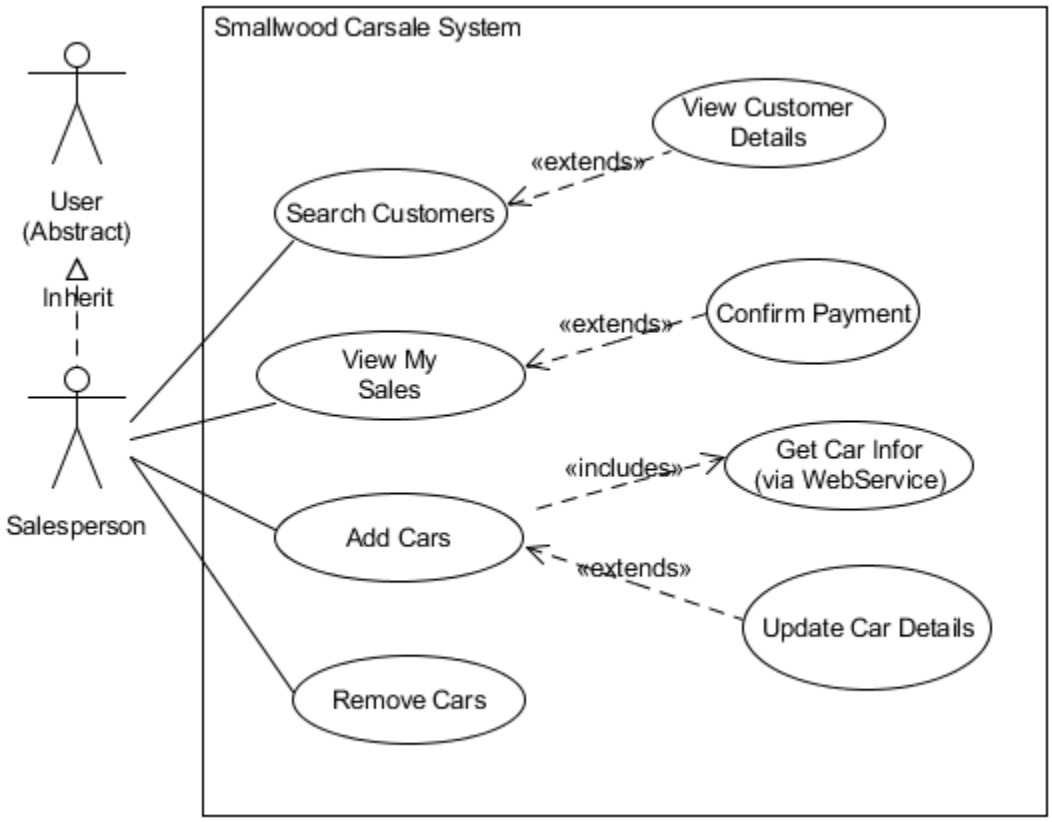

Figure 4: Salesperson Use Case

3. Core program functionality

3.1 System Architecture

There are four modules in Smallwood Carsales system: Database, EJB, Website Client and App Client (See Figure 5). EJB module also consumes external web services but they are not included in the project scope.

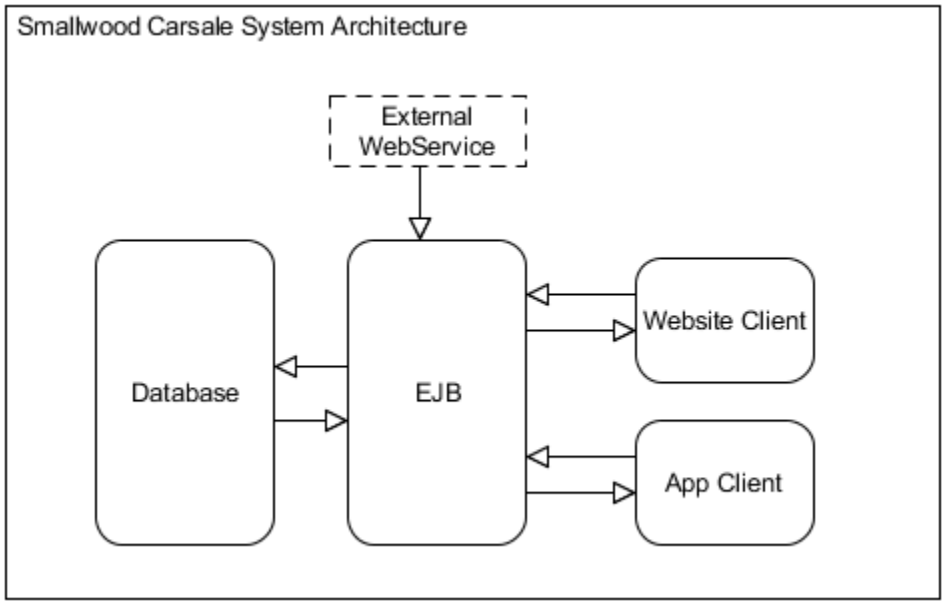

Figure 5: System Architecture

3.2 Entities (Database)

There are mainly five entities used in the system.

1) User

User is an abstract entity hosting common properties of customer and salesperson.

2) Customer

Customer entity inherits from User and also keeps a collection of its corresponding sale records.

3) Salesperson

Salesperson entity inherits from User and also keeps a collection of sales that are made with him.

4) Car

Car entity represents specific car objects in reality.

5) Sale

Sale entity holds information of trades made between customer and salesperson about specific cars. It also record the payment status of that trade.

The data transformation and communication between entities and database are completed via JPA. Both JPQL and Criteria API are used to retrieve data from database. The inheritance between user and customer/salesperson are implemented using Single-Table strategy because there is no additional properties which are specific for child entities at this stage. It is also more efficient. Details entity definition are listed in Section 7 Data dictionary).

3.3 EJB

EJB module implements all business logic and operations in the system. These operations are categorized into four session beans. All these beans are stateless in order to reduce server burden and memory consumption. To support application client, remote interface are enabled.

1) UserManagerBean

UserManagerBean implements all user management related operations.

Table 1: UserManagerBean Functions

Function Name	Description	Input	Output
findUserById	Get personal details of user	**Long id**: user id	User entity
findUserByEmail	Get user details by email	**String email**: user email	User entity
Register	Register new user to the system	**User user**: new user entity	Void (exception is handled outside)
UpdateUser	Update user personal detail	**User user:** user entity	void
findUsers	Find users by criteria	**Id, first name, last**	User list

		name, email	
findAllSalesperson	Get available salespersons	**None**	Salesperson list
RemoveUser	Remove user from system	**Long id**	none

2) CarManagerBean

CarManagerBean hosts all car and car catalog related operations.

Table 2: CarManagerBean Functions

Function Name	Description	Input	Output
findCarsByCriteria (All cars, even sold cars)	Search for a car in the catalogue by the following combination of criteria	**String modelNo, String modelName, String make, Car.CarType Type**	List<Car>
findAvailableCarsByCriteria (Only cars in storage)	Search for a car in the catalogue by the following combination of criteria	**String modelNo, String modelName, String make, Car.CarType Type**	List<Car>
findCarByVIN	Find car by VIN	**VIN**	Car
checkStorage	A customer can only buy car(s) that have at least one in stock	**String VIN**	Boolean: if it is available
addCar	Add an item to the catalogue	**Car car**	Void (exception is handled outside)
populateCarViaService	When adding a car to the system, the information of the car can be obtained via web service instead of being entered manually	**String VIN**	Car
updateCar	update the details of an item in the	**Car car**	void

	catalogue		
removeCar	Delete an item from the catalogue;	**String VIN**	void

4) SaleManagerBean

SaleManagerBean includes all sale related functions.

Table 3: SaleManagerBean Functions

Function Name	Description	Input	Output
checkCustomerPayment	A customer cannot buy another car if he/she has not paid off all purchases on their account	**Long id:** customer id	Boolean: if the customer has paid all previous purchases
buy	Customers can buy car(s) using the system	**Sale sale**: include all necessary information for one sale	void
removeSale	Withdrawn sale records	**Id**	none
confirmPayment	Salesperson is able to confirm the payment of a sale has been made	**Sale sale**: one sale record	void
findSaleById	Find sale by id	Id	sale
retrieveSalesRecord	Find user, either customer or salesperson related sale records	User id	Sale list

3.4 Website Client

Website client is development to support online service to users. It is implemented using Java Server Faces framework. EJB functions are consumed in JSF Managed Beans. Web pages and navigation route are displayed in Figure 6: Website Client Design.

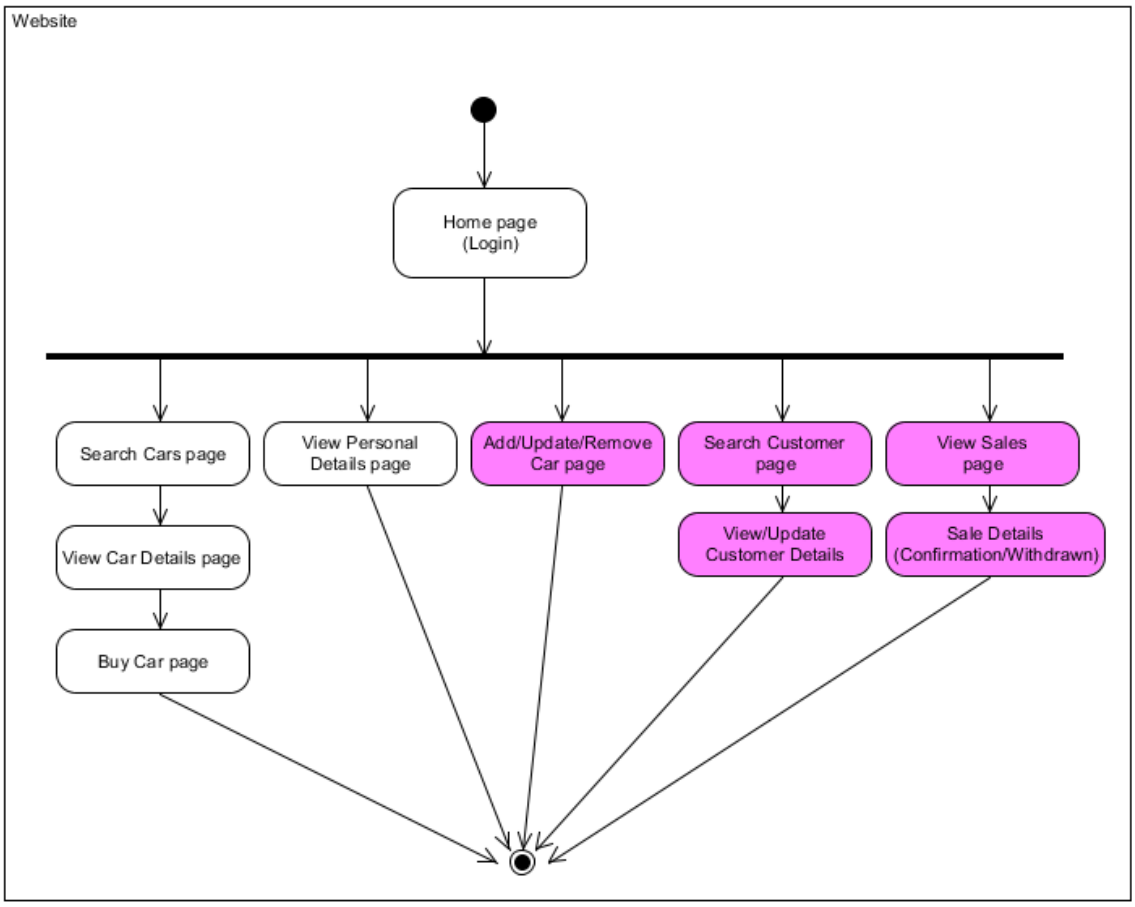

Figure 6: Website Client Design

To present a consistence interface to users, a template page is used to define the general layout of all website pages (see Figure 7: Template Layout).

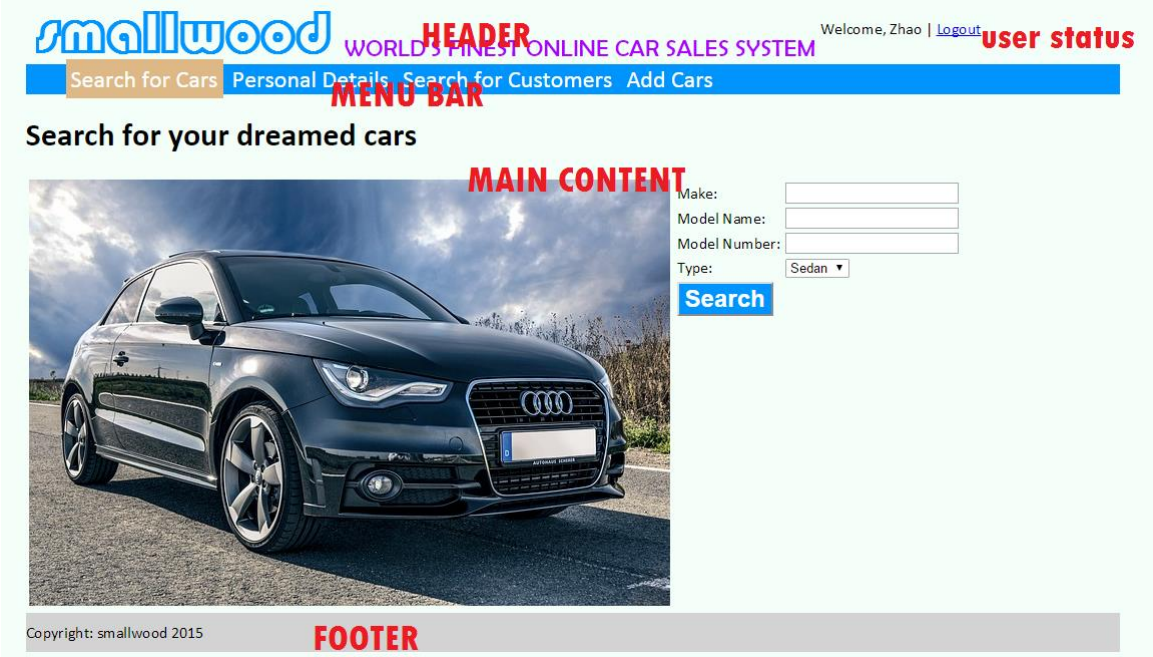

Figure 7: Template Layout

There are six JSF Managed Beans to support website interaction.

Name	Description	Scope	Utilization
LoginBean	Holding current logged user and consuming user account related EJB methods	SessionScoped	Login; register; view personal details; referenced by other beans
UserListBean	Holding user list data and user search criteria, like email, name etc.	SessionScoped (it is convenient to restore search results when return from other views)	Search customer, salesperson, create/update/remove users
CarListBean	Holding car search criteria and result returned from EJB.	SessionScoped	Search cars, buy car, create/update/remove cars
SaleListBean	Holding sale records of certain user	View	Retrieve customer purchases and salesperson sales
SaleBean	Holding details of a specific sale record	View	Sale confirm, buy car, withdrawn sale

3.5 App client design

App client will be developed using JAVA SWING and only implement search / view car functions.

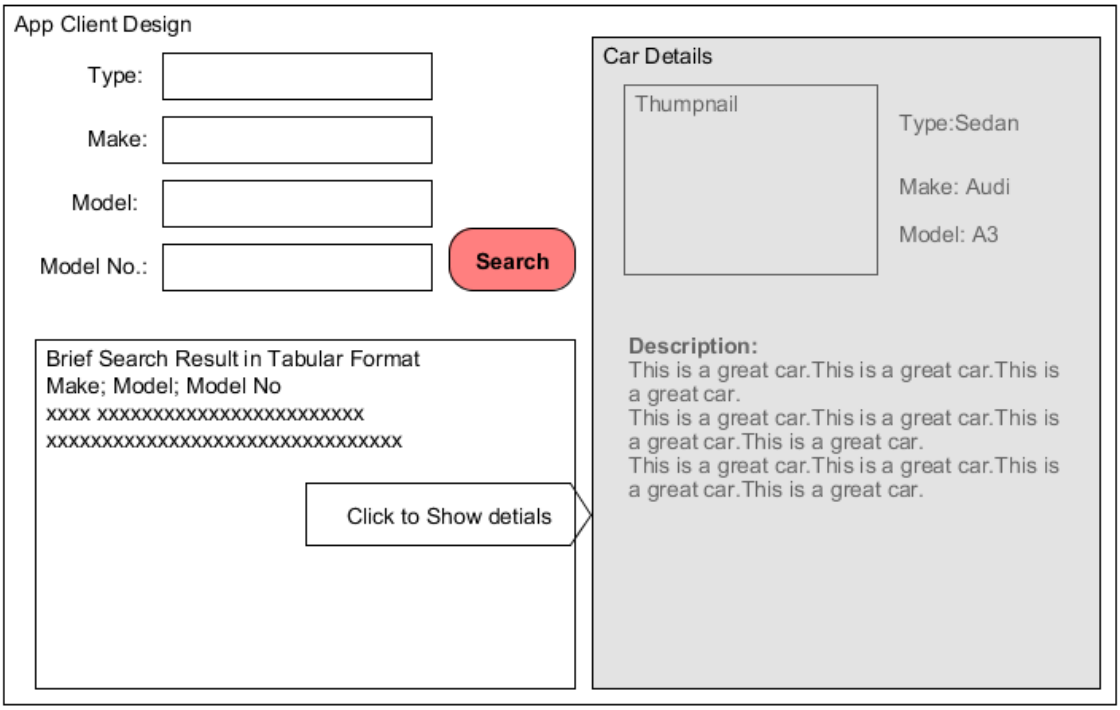

Figure 8: App Client Design

4. Usability Design Review

Since to the users, the client interface would be their playground to interact with the system, my usability design review will focus on the website design part.

1) Navigation

A menu bar is designed to navigate users over the whole site. It stays at the top of every page right below the banner. The order of each menu item is carefully considered so the more frequently used pages will display before other less used ones. In this case, search car would definitely be the most used page, so it always the first. The logo part also provide a shortcut link to the search car page.

Meanwhile, I tried to put relevant information in one single page to reduce the demands to jump over different pages. So users do not need to go to many pages in their operation. Combining the features of efficient menu design and less jumps, the website provides a competent navigation to users.

2) Familiarity

The design and layout of the website is similar with current mainstream website design. No quirky designs are used. The response behavior of page element is also consistent with standard design. Everyone would be an expert to use it.

3) Consistency

The following development principle are maintained to ensure consistency:

(a) One, and only one, master page is used to deliver consistent appearance through different pages. The layout at top, left and bottom of the pages are fixed. Content pages occurs at a fixed position as well.

(b) One color scheme is adopted over all pages: blue as the website main theme; red to show errors.

(c) One font-family is chosen and used in all pages.

4) Error Prevention

We fully understand that users may encounter some errors in the registration page. To prevent these errors happening,

(a) Various kinds of validation controls are used to check user input.

(b) Required input fields are marked using blue border to inform users. Thus help user avoid errors far before they hit the submit button.

(c) Tooltips are comprehensively used to offer guidance to users

(d) Only available operations are open to users after rigorous tests

5) Feedback

The application will react with each user operation and give feedback to users.

(a) All clickable links will change color when mouse hovers.

(b) When users enter incorrect information, error message will show up.

(c) All operable actions will bring appropriate response, either executing user instruction or showing error messages

6) Visual Clarity

Following approaches are used to ensure visual clarity:

(a) High contrast color is used in the website design. The background and fore content can be clearly identified by users.

(b) Important information is highlight using bold font or in bright color.

7) Flexibility & Efficiency

(a) To achieve reasonable efficiency, the website only includes necessary content without some fancy, however heavy effects.

(b) EJB session beans are all designed to be stateless to reduce server memory usage and burden.

(c) The scopes of JSF Managed Beans are designed to be as short as possible in order to reduce data retained in web server.

(d) Use AJAX to reduce page post back thus improve user experience

5. Checklist of site functionality

	TICK
1. Pass Functionality	**if complete**
Search for Car by	✓
Make	✓
Model,	✓
Model No	✓
Type	✓
Results with tabular format with heading.	✓
Option to view the full details	✓
Additional Pass Requirement	✓
Learning Summary Report	✓
2. Credit Functionality	✓
Login using a username and password	✓
Sales people can: View	✓
Add	✓
Update	✓
Delete cars	✓
3. Low to Mid Distinction Functionality	✓
Customers can buy car(s)	✓
Sales people search Customers by a **combination** of ID, last name, first name, type and email.	✓
When adding a car to the system, the information of the car can be obtained via web service	✓
4. Mid to Mid Distinction Requirements	✓
This Design Document	✓
5. High Distinction Requirements	✓
Research Report	✓
6. Technical Requirements	✓
Pass	✓
JSF web clients	✓
GUI Swing application clients	✓
Persistence API	✓
Application managed entity manager or container managed entity manager.	✓
Credit	✓
ONLY web client is required	✓
Interaction between clients and database handled by EJBs	✓
BOTH Criteria API and JPQL	✓
Distinction	✓
Ability of mapping inheritance to database must be demonstrated.	✓
Bean validations used to validate data.	✓
Consumption of web services conducted in EJBs.	✓

Application secured using JAAS API.	✓
Audit	✓
No breaking of copyright	✓

6. User stories

6.1 Can salesperson buy cars in Smallwood Carsales?

Smallwood Carsales is operated based on a company, rather than a third party trading platform. People who sells cars in the system (salesperson) is company staff instead of private sellers. As a result, the salespersons are not allowed to buy cars by themselves (or it is out of project scope). This makes the user role design simpler: salespersons maintain car catalog and confirm sale payment while customers search and bug cars.

6.2 What cars are Smallwood Carsales selling? New or secondhand?

Business in Smallwood Carsales is based on specific cars no matter they are new or secondhand. When its salespersons get sources of available cars, they enter the details into system and then they are available for customers. As a result, VIN is a crucial identifier for cars in Smallwood Carsales. I understand in some car sales, especially for new cars, users only need to search car make or models. They are actually searching a kind of cars, no matter what VIN it is. But it is not the case in Smallwood Carsales. In our system, the car matching a search will be exactly the particular car. All trading is on VIN.

6.3 How do customers know which salesperson to choose in making a purchase?

Actually, no users will decide to buy a car just by viewing the information online. They will go to the company to check the car personally. And over there will be a salesperson to serve them. So when a user makes a purchase, he/she definitely knows which salesperson it is.

6.4 User account management

In this application, salesperson account are maintained by admin user while customers registered by salesperson. In other words, user cannot register their accounts by themselves, but only when they pay visit to our company and ask our salesperson to register account for them.

7. Data dictionary

Entity: User			
Name	**Datatype**	**Description**	**Example**
ID	Integer	Unique identifier	1001
LNAME	String	Last name; must not contain numeric value	Smith
FNAME	String	First name; must not contain numeric value	Will
EMAIL	String	Email address, also used as user name when logging in.	Will.smith@gmail.com
PASSWORD	String	Password	You guess
TYPE	String	SalesPerson, Customer, Admin	Customer
ADDRESS	String	Physical address	756 Dandenong Road, Caulfield, 3162
PHONE	String	Phone number; either a landline or mobile number	9847 4562

Entity: Car			
Name	**Datatype**	**Description**	**Example**
VIN	String	Vehicle Identification Number (A unique identifier)	1M8GDM9AXKP042788
MODELNUM	String	Model number; a unique identifier of the car	(Corolla) 150 series
MODELNAME	String	Model Name	Corolla
MAKE	String	Make /Manufacturer	Toyota
TYPE	String	Categories: Sedan, 4 wheel drive and Truck	Sedan
PRICE	Int	Sale price	$250,000
THUMBNAIL	Image	Photo	Loading…

DESCRIPTION	String	Plain text description of a car	Oh what a fancy car.
URL	String	A URL where more information of a car can be obtained	http://www.google.com

Entity: Sale			
Name	**Datatype**	**Description**	**Example**
ID	Integer	Unique identifier	20004
CUSTID	Integer	ID number of the customer who bought the car	1001
BUYAT	Date	Date when the sale is made	09/17/2015
CARVIN	String	VIN of the car sold	1M8GDM9AXKP042788
SALEPID	Integer	ID number of the salesperson who sold the car	1008
SALESTATUS	String	Current status of sales, like in-progress, confirmed	In-progress
CONFIRMAT	Date	Date when the sale payment is confirmed	10/22/2015